Computational Methods towards Personalized Cancer Vaccines and their Application through a Web-based Platform

Dissertation

der Mathematisch-Naturwissenschaftlichen Fakultät
der Eberhard Karls Universität Tübingen
zur Erlangung des Grades eines
Doktors der Naturwissenschaften
(Dr. rer. nat.)

vorgelegt von

M.Sc. Christopher Mohr

aus Friedrichshafen

Tübingen
2020

Gedruckt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät der Eberhard Karls Universität Tübingen.

Tag der mündlichen Qualifikation:	24.06.2021
Dekan:	Prof. Dr. Thilo Stehle
1. Berichterstatter:	Prof. Dr. Oliver Kohlbacher
2. Berichterstatter:	Prof. Dr. Nico Pfeifer

“Nothing in life is to be feared, it is only to be understood.
Now is the time to understand more, so that we may fear less.”

- Marie Curie (1867–1934)

Erklärung

Ich erkläre hiermit, dass ich die zur Promotion eingereichte Arbeit mit dem Titel:

Computational Methods towards Personalized Cancer Vaccines and their Application through a Web-based Platform

selbständig verfasst, nur die angegebenen Quellen und Hilfsmittel benutzt und wörtlich oder inhaltlich übernommene Stellen als solche gekennzeichnet habe. Ich erkläre, dass die Richtlinien zur Sicherung guter wissenschaftlicher Praxis der Universität Tübingen (Beschluss des Senats vom 25.5.2000) beachtet wurden. Ich versichere an Eides statt, dass diese Angaben wahr sind und dass ich nichts verschwiegen habe. Mir ist bekannt, dass die falsche Abgabe einer Versicherung an Eides statt mit Freiheitsstrafe bis zu drei Jahren oder mit Geldstrafe bestraft wird.

Abstract

Cancer immunotherapy is a treatment option that involves or uses components of a patient's immune system. Today, it is heading towards becoming an integral part of treatment plans together with chemotherapy, surgery, and radiotherapy. Personalized *epitope-based vaccines* (EVs) serve as one strategy that is truly personalized. Each patient possesses a distinct immune system, and each tumor is unique, rendering the design of a potent vaccine challenging and dependent on the patient and the tumor. The potency of a vaccine is reliant on the ability of its constituent epitopes – short, immunogenic antigen fragments – to trigger an immune response. To assess this ability, one has to take into account the individuality of the immune system, among others conditioned by the variability of the human leukocyte antigen (HLA) gene cluster. Determining the HLA genotype with traditional experimental techniques can be time- and cost-intensive. We proposed a novel HLA genotyping algorithm based on integer linear programming that is independent of dedicated data generation for the sole purpose of HLA typing. On publicly available next-generation sequencing (NGS) data, our method outperformed previously published approaches. HLA binding is a prerequisite for T-cell recognition, and precise prediction algorithms exist. However, this information is not sufficient to assess the immunogenic potential of a peptide. To induce an immune response, reactive T-cell clones with receptors specific for a peptide-HLA complex have to be present. We suggested a method for the prediction of immunogenicity that includes peripheral tolerance models, based on gut microbiome data, in addition to central tolerance, previously shown to increase performance. The comparison to a previously published method suggests that the incorporation of gut microbiome data and HLA-binding stability estimates do not enhance prediction performance.

High-throughput sequencing provides the basis for the design of personalized EVs. Through genome and transcriptome sequencing of tumor and matched non-malignant tissue samples, cancer-specific mutations can be identified, which can be further validated using other technologies such as mass spectrometry (MS). Multi-omics approaches can result in the acquisition of several hundreds of gigabytes of data. Handling and analysis of such data usually require data management solutions and high-performance computing (HPC) infrastructures. We developed the web-based platform *qPortal* for data-driven biomedical research that allows users to manage and analyze quantitative biological data intuitively. To emphasize the advantages

of our data-driven approach with an integrated workflow system, we conducted a comparison to Galaxy. Building on qPortal, we implemented the web-based platform *iVacPortal* for the design of personalized EVs to facilitate data management and data analysis in such projects. Further, we applied the implemented methods through iVacPortal in two studies of two distinct cancer entities, indicating the added value of our platform for the assessment of personalized EV candidates and alternative targets for cancer immunotherapy.

Zusammenfassung

Immuntherapie gegen Krebs, eine Therapieform, die Bestandteile des Immunsystems eines Patienten einbezieht oder verwendet, ist auf dem Weg, ein integraler Bestandteil von Behandlungsplänen zusammen mit Chemotherapie, Chirurgie und Strahlentherapie zu werden. Personalisierte epitopbasierte Impfstoffe stellen dabei eine Strategie dar, die wahrlich personalisiert ist. Die Tatsache, dass jeder Patient über ein individuelles Immunsystem verfügt und darüberhinaus jeder Tumor einzigartig ist, machen den Entwurf wirksamer Impfstoffe anspruchsvoll und sowohl abhängig von dem Patienten als auch von dem vorhandenen Tumor. Die Wirksamkeit eines Impfstoffes ist dabei bedingt durch die Fähigkeit der enthaltenen Epitope, kurzer immunogener Antigen-Fragmente, eine Immunantwort auszulösen. Um diese Fähigkeit abschätzen zu können, ist es notwendig, die Individualität des Immunsystem zu berücksichtigen, die unter anderem durch die Variabilität des humanen Leukozyten-Antigen-System (HLA) bedingt ist. Die Bestimmung des HLA-Genotypes durch herkömmliche experimentelle Methoden kann zeit- und kostenintensiv sein. Um dieses Problem zu lösen, stellen wir einen neuen Algorithmus vor, basierend auf ganzzahliger linearer Optimierung, der nicht abhängig von einer speziell für die HLA-Typisierung vorgesehenen Datengenerierung ist. Wir zeigen basierend auf öffentlich zugänglichen Next-Generation-Sequencing (NGS) Daten, dass unsere Methode bereits veröffentlichte Ansätze übertrifft. Die Bindung eines Peptids an ein HLA-Molekül ist eine Voraussetzung für die Erkennung durch T-Zellen. Hierfür existieren genaue Vorhersagealgorithmen. Diese Information ist allerdings nicht ausreichend, um die Immunogenität eines Peptides abschätzen zu können, da für das Hervorrufen einer Immunantwort reaktive T-Zellen mit einem spezifischen Rezeptor für den Peptid-HLA-Komplex vorhanden sein müssen. In dieser Arbeit stellen wir daher eine Methode zur Vorhersage der Immunogenität vor, welche die Modellierung der peripheren Toleranz, basierend auf Darmmikrobiom-Daten in Ergänzung zu der zentralen Toleranz beinhaltet, für die bereits eine Steigerung der Vorhersagequalität gezeigt wurde. Der Vergleich mit einer bereits publizierten Methode lässt den Schluss zu, dass die Verwendung von Darmmikrobiom-Daten und HLA-Bindungsstabilitätsdaten zu keiner Verbesserung der Vorhersagegenauigkeit führt.

Hochdurchsatzsequenzierung ist die Grundlage für die Entwicklung von personalisierten epitop-basierten Impfstoffen. Durch die Sequenzierung von Genomen und Transkriptomen von

Tumorproben und nicht-malignen Gewebeproben können tumorspezifische Varianten identifiziert und durch weitere Technologien wie Massenspektrometrie (MS) validiert werden. Multi-Omik Ansätze können dabei zur Generierung von hunderten Gigabyte an Daten führen. Die Handhabung und Analyse dieser Daten ist häufig nur mit Datenmanagement-Lösungen und High-Performance-Computing (HPC)-Infrastrukturen möglich. Wir präsentieren die Webplattform *qPortal* für datenfokussierte biomedizinische Forschung; diese stellt den Benutzern leicht nutzbare Möglichkeiten für das Management und die Analyse von Daten aus dem Bereich der quantitativen Biologie zur Verfügung. Die Vorteile unseres datenfokussierten Ansatzes in Verbindung mit einem integrierten Workflowsystem werden dabei in einem Vergleich zu Galaxy deutlich. Um Datenmanagement und Analysen in Projekten zur Entwicklung von personalisierten epitop-basierten Impfstoffen zu erleichtern, haben wir basierend auf qPortal die webbasierte Plattform *iVacPortal* entwickelt. Die dabei implementierten Methoden haben wir des Weiteren durch iVacPortal in zwei Studien zu zwei verschiedenen Tumorentitäten angewendet und zeigen dabei den Mehrwert unserer Plattform im Hinblick auf die Beurteilung von Impfstoffkandidaten und alternativen Therapiezielen für die Krebsimmuntherapie.

Acknowledgments

Throughout my Ph.D. and the writing process of this dissertation, I received a great deal of support. First of all, I would like to thank my supervisor Prof. Oliver Kohlbacher for giving me the opportunity to work on this Ph.D. and for providing me with excellent support and guidance throughout the last years.

I would like to thank my supervisor Prof. Hans-Georg Rammensee, especially for introducing me to the field of immunology and awakening my interest in this exciting field already during my studies. Moreover, I would like to thank him for the opportunity to attend many instructive meetings and be part of exciting discussions.

I would like to thank Prof. Nico Pfeifer for agreeing to review this thesis.

During my Ph.D. I had the opportunity to meet a lot of great people in the Applied Bioinformatics group. To the members and former members of the Immuno SIG, including Leon Bichmann, Magdalena Feldhahn, Benjamin Schubert, András Szolek, Matthias Walzer, thank you for the productive joint work on projects, your support, and the excellent discussions. Thanks to all colleagues and former colleagues of the Applied Bioinformatics group. Special thanks to Timo Sachsenberg and Luis de la Garza for the shared teaching experiences, talks over a cup of coffee, and help when needed.

I thank my colleagues and former colleagues at QBiC. In particular, I would like to thank Sven Nahnsen, Andreas Friedrich, David Wojnar, Erhan Kenar, Aydin Can Polatkan, and Stefan Czemmel for the joint work on qPortal and other exciting projects.

I would like to thank Markus Löffler for the excellent collaborative work on an exciting multi-omics project.

Special thanks to Lena Freudenmann, Marie Gauder, Julia Günzler, Alex Peltzer, Timo Sachsenberg, and Julian Späth for proofreading parts of this thesis.

Last but not least, I would like to thank my family, in particular my parents, my grandparents, Christa, Johanna, and Stefan, for their love and support. Finally, I would like to thank Jule for her patience and support. Thank you for being there.

- In accordance with the standard scientific protocol, I will use the personal pronoun ***we*** to indicate the reader and the writer, or my scientific collaborators and myself.
- Unless stated otherwise, we refer to a mutated neoepitope when we use the term neoepitope.

Contents

Chapter 1

Introduction

Cancer is one of the leading causes of death worldwide. In 2018, an estimated 18.1 million new cases of cancer and 9.6 million cancer deaths were reported[1]. The complexity with respect to development and progression is one of the major challenges in cancer research. Loss of growth control is only one characteristic of malignant tumors. In 2000, Hanahan & Weinberg proposed six common traits, defined as *hallmarks*, that drive the transformation from normal to cancer cells[2]. Those alterations in cell physiology are supposed to be shared by all cancer types and responsible for the development of the corresponding genotype. However, as pointed out by Lazebnik in 2010, these alterations are not unique for malignant tumors but might also be present in benign tumors. The unique trait is the invasion of surrounding healthy tissue and metastasis, which is one of the distinct characteristic features of cancer[3]. Processes that are beneficial for disease progression are mainly driven by the tendency of tumors to acquire thousands of somatic mutations, as well as clonal heterogeneity. Unfortunately, these fundamental properties of cancer cause natural progression and therapeutic resistance due to clonal evolution and outgrowth of subclones[4,5]. Despite the advances in cancer diagnostics, there is still a lack of in-depth knowledge of most of the underlying mechanisms attributed to the complexity of the disease. In spite of the incomplete understanding, since the last decades of the 20th century, new treatments were established. While the commencements of oncology were solely based on surgery, radiotherapy and chemotherapy opened up new frontiers in the treatment of cancer, and combination therapies are used efficiently. Modern oncology includes three main sub-areas, which are surgical oncology, radiation oncology, and medical oncology. Besides chemotherapy, medical oncology makes use of therapeutic approaches, such as hormonal therapy and immunotherapy. Notably, cancer immunotherapy has been recently recognized as one of the most promising approaches in cancer treatment. The beginnings of cancer immunotherapy go back to the 1990s. The development of therapeutic monoclonal antibodies (mAbs) such as trastuzumab and rituximab and the approval of the latter by the Food and Drug Administration (FDA) in 1997 was one of the milestones in the beginnings

of immune-based approaches. The specificity of these drugs with respect to the targeted cancer entities is indicative of a shift in cancer treatment approaches. Trastuzumab has been developed for the treatment of HER2-positive breast cancer, while rituximab was developed for chronic lymphocytic leukaemia (CLL) and non-Hodgkin lymphoma, targeting the protein CD20. Until the end of the 20th century, most of the anti-cancer drugs merely killed cancer cells. Such cells were most of the time detected as rapidly dividing cells. Still, most of these chemotherapeutic drugs kill not only cancer but also healthy cells. Such systemic therapies have remained the foundation of modern cancer treatments resulting in severe side effects. However, the overall conception in the field has changed from non-specific cytotoxic agents to highly selective drugs targeting specific molecules and mechanisms. These targeted approaches show lower toxicities and tend to be less prone to the development of resistance. Targeted therapies act on molecular structures and pathways with essential importance for the development and retention of tumors and therefore exploit vulnerabilities of the malignancy. Therapies employ drugs as angiogenesis inhibitors, growth signal inhibitors, and apoptosis-inducing agents. In contrast, immunotherapies intend to induce or enhance immune system modulated mechanisms against the tumor. Today, immunotherapy complements conventional therapies, such as surgery, chemotherapy, and radiotherapy.

Currently, cancer immunotherapy comprises several different treatment approaches. Non-specific immunotherapies encompass agents, such as interferons and interleukins, which are utilized to globally enhance immune effector functions. Another approach is the use of mAbs, designed to specifically bind to a cancer-specific target. mAbs directly guided to operate on cancer cells can be either unmodified ('naked'), linked to chemotherapeutic drugs, or conjugated with radioactive particles. Immune checkpoint inhibitors constitute another class of cancer immunotherapy drugs. The immune system regulates T cell-mediated immune responses through specific proteins, so-called immune checkpoint molecules. Checkpoint inhibitors, mainly mAbs, inhibit proteins used by cancer cells to escape immune-mediated destruction. Deactivating signals on T cells are switched off, facilitating the killing of cancer cells. Concerning cancer vaccines – another cancer immunotherapy approach – one has to distinguish between prophylactic and therapeutic vaccines. The latter is so far primarily of scientific interest in the targeted treatment of cancer. Vaccines in a traditional sense, prophylactic vaccines are, in contrast, administered to healthy individuals to induce immunity against certain bacteria and viruses to protect them from infections.

In the last decades, the success of prophylactic vaccines has become even more critical since infections by certain viruses were identified as a potential cause for the development of cancer[6]. Several studies suggest that carcinogenic infections cause around 15 to 20% of the global cancer burden[7–9]. Hepatitis B Virus (HBV) and Human Papillomavirus (HPV) were identified to cause human cancer. Today, there are vaccines available against HPV[10] and HBV[11] which are highly effective in inducing protective immunity. Further, different strategies exist

for the treatment of cancer employing therapeutic vaccines. One approach uses *ex vivo* matured T cells or dendritic cells (DCs) which are primed using tumor antigens and then administered back to patients. Antigens can arise from different sources. Besides cellular therapies[12,13], proteins or peptides can be used as agents to induce or enhance anti-tumor immunity. Usually, these vaccines are administered together with an adjuvant or an immune modulator and include either intact protein subunits or corresponding DNA or RNA encoding *tumor-associated antigens* (TAAs). However, especially the identification of TAAs capable of inducing a potent immune response has been proven difficult. Besides the fact, that antigens are highly variable in their ability to leverage immunogenicity and immune modulatory mechanisms in tumor environments, these agents are not universal. Possible antigens do not only differ between cancer entities but also vary widely across patients even with the same tumor type[14]. Due to the heterogeneity of the somatic mutational landscape, tumor-associated agents should be identified on a patient-specific level. Personalized *epitope-based vaccines* (EVs) are one example of actively personalized immunotherapies. Here, one tries to identify patient-specific epitopes, short, immunogenic antigen fragments. Epitopes are presented on *major histocompatibility complex* (MHC) proteins and are capable of inducing T cell-mediated immunity. In humans, MHC proteins are encoded by the highly polymorphic genes of the polygenic *human leukocyte antigen* (HLA) region, located on chromosome six. In the context of cancer, epitopes can originate from altered protein expression or non-synonymous mutations resulting in new protein sequences and thus new epitopes, also referred to as neoepitopes[15]. As T cell-mediated immunogenicity is dependent on MHC binding, responses to epitope-based vaccines critically depend on the number of peptides presented by MHC molecules. Additionally, patient survival has been shown to strongly correlate with the number of peptides triggering an immune response[16,17]. Since the number of peptides that can be included in a vaccine is limited, the selection of epitopes with the highest likelihood of success is critical. In the context of cancer immunotherapy, success can be defined as a potent, broad and sustained peptide-specific immune response with a minimal risk of autoimmunity. Therefore, epitopes derived from neoantigens, presumably not present in non-malignant tissues, were recognized as an optimal choice in this respect. Further, the number of neoepitopes are strong correlators of clinical response to immune checkpoint inhibition[18]. Due to the importance of identifying suitable vaccine targets, recent developments in instrumentation, sample processing, and bioinformatics have been used to leverage target identification and characterization[19]. In general, high-throughput technologies play an ever-increasing role in cancer genomics, mainly driven by new technologies reducing costs and turnaround times. State-of-the-art experimental techniques such as *next-generation sequencing* (NGS) and proteomics are used to identify tumor markers, to assess the genetic background of a patient or to determine somatic, tumor-specific variants of a tumor and therefore ultimately lay the groundwork for the identification of patient-specific neoepitopes[19]. The latter usually consists of three main steps: Sequencing and identification of genetic alterations, HLA

genotyping, and identification of potential neoepitopes employing HLA binding predictions of mutation-carrying peptides followed by prioritization and selection of identified targets. The first step typically includes sequencing of matched tumor and non-malignant tissue from patients using exome or whole genome sequencing. Detection of somatic mutations is done using somatic variant calling algorithms. At this time, various publicly available software packages exist[20–23]. However, the concordance of the results among different approaches is rather low[24] due to events such as the admixture of tumor cells in the non-malignant sample and vice versa, subclonal variants, copy number variations, or ploidy changes. Still, these methods are essential for the identification of neoepitopes. Variants are usually annotated with gene and transcript associations and incorporated into the respective transcript sequences. Those sequences are then translated into the corresponding protein sequences. Immunoinformatic framework solutions exist to solve these tasks efficiently[25,26]. As the binding of epitopes to HLA molecules is the most specific step in the antigen presenting pathway of the human immune system, the design of personalized therapeutic vaccines highly depends on the availability of a patient's HLA genotype. In general, high-resolution HLA typing is essential for various clinical applications. Today, most of the experimental protocols used for HLA genotyping are sequence-based typing (SBT) protocols, including Sanger SBT, and more recently based on NGS[27–31]. Hence, *in silico* HLA genotyping methods based on NGS data have been noticed as an economical and efficient alternative and obtained acceptance in recent years. *In silico* HLA typing methods are therefore an active area of research. Several methods[32–36] differing in used data types (exome, RNA, and whole-genome) and the resolution of predicted genotypes exist. However, many approaches are based on data solely produced for HLA genotyping, or do not provide sufficient accuracy for clinical application.

Therefore, we developed *OptiType*, a novel HLA genotyping algorithm based on integer linear programming. OptiType does not depend on data specifically enriched for the HLA gene cluster and can be applied to *whole-genome sequencing* (WGS), *whole-exome sequencing* (WES), and *whole-transcriptome sequencing* (WTS) data. We show that OptiType generates high-precision results at clinically relevant resolution (four digits), outperforming existing computational approaches for HLA class I genes.

As outlined, immunogenicity is one of the primary factors to be considered for the selection of neoepitope vaccine candidates. Traditionally, predicted HLA binding affinities are used as immunogenicity estimates since immunogenicity prediction methods do not yet provide acceptable precision for clinical application. Suggested methods incorporate metrics along the antigen processing pathway in addition to HLA binding[37–40] or use physicochemical properties of target peptides[41,42]. We developed a machine learning approach that incorporates HLA binding estimates in addition to a model for immunological tolerance as suggested earlier[43]. We extended this approach by integrating gut microbiome data to model peripheral tolerance, shown to be affected by commensal microorganisms[44,45].

Although publicly available methods exist for the identification of suitable candidates for personalized cancer vaccines, various hurdles impede their application within biomedical projects. One problem is the vast amount of data generated by high-throughput experiments. NGS data produced by state-of-the-art instruments are often in the range of terabytes per patient. This trend is continuing with even more affordable and faster sequencing technologies. Additionally, more and more biomedical projects make use of multi-omics approaches to assess hypotheses on multiple biological layers. Therefore, it is increasingly challenging to manage the amounts of various data types, numerous patients, extracted tissues, and at the same time record experimental variables and meta data. Despite the problems concerning project and data management, the installation and application of computational methods typically require at least some degree of expert knowledge. Besides, high-performance computing (HPC) infrastructures are usually needed to analyze typical projects efficiently. Existing approaches are trying to solve the mentioned problems by providing compute power, storage resources, and easy-to-use graphical interfaces and therefore hide the complexity from the user. Platforms, such as Galaxy[46] and GenePattern[47], offer users access to computational pipelines and HPC resources through web interfaces. However, existing approaches often neglect the project and data management aspects. To overcome the described hurdles within biomedical projects, we developed *qPortal*, a web-based platform for biomedical applications. qPortal provides resources for project management and data management and empowers users to conduct their experiments through our web-based platform. In comparison to existing solutions, our platform implements a data-driven approach that comes with an added value. Using the implemented functionality, we developed a workbench for the design of personalized (also referred to as individualized) vaccines (*iVacPortal*). The portal offers computational methods required to generate mandatory information, such as the HLA genotype or neoepitope candidates. We utilized iVacPortal in two projects, comprising multiple patients of two different cancer entities, and assessed neoepitope candidates for personalized cancer vaccines on multiple omics layers. While we identified potential vaccine candidates in almost all patients, we could not identify neoepitopes in the patient-specific ligandomes. However, we provide potential explanations for this, such as the influence of the sensitivity of employed devices and *in silico* identification pipelines. Additionally, we validated our approach by reconfirming the identification of neoepitopes using our pipeline in a previously published dataset on melanoma.

This thesis is structured in seven chapters. Following this introduction, the biological background introduces the immune system (Section 2.1), in particular, the adaptive immune response (Section 2.2), cancer (Section 2.3), and epitope-based vaccines (Section 2.4). The experimental background of NGS and bioinformatic applications are outlined in Section 2.5. We discuss state-of-the-art computational methods for computational immunomics as well as workflow management systems in Section 2.6. The following chapters entail the results of this

thesis. In Chapter 3, details on OptiType, our suggested method for HLA genotyping from NGS data, are given. We present our efforts on the modeling of immunological tolerance to improve the performance of predicting T-cell reactivity (Chapter 4) and evaluate the performance in comparison to a previously published approach on an experimentally tested data set. Design and implementation details of qPortal and iVacPortal are given in Chapter 5. Chapter 6 focuses on the utilization of our developed methods and infrastructure. We used presented solutions within two studies on the development of personalized cancer vaccines. A conclusion and outlook are given in Chapter 7.

Chapter 2

Background

This chapter outlines the biological background of this thesis. The first part (Section 2.1) provides a general overview of the immune system, followed by a more detailed description of the adaptive immune response (Section 2.2). Section 2.3 provides a general view on cancer and treatment options, whereas Section 2.4 addresses epitope-based vaccines.

2.1 The Immune System

This section introduces the immunological background and is mostly based on Janeway et al.[48] to which we refer to for a more comprehensive introduction.

The human body is continuously exposed to microorganisms and viruses, that are often pathogenic. However, this exposure will lead to diseases only rarely. Cells, tissues, and molecules of the immune system, interacting in a dynamic network, combat most pathogens before a disease develops. Through the different players, the immune system recognizes pathogens and defends the organism against these infectious agents and their toxins. The immune system has to detect pathogenic agents and aberrant host cells and distinguish them from (healthy) cells of the body. It can be classified into the adaptive and innate immune system (Figure 2.1). Both of the two interacting and entangled subsystems depend on the activity of leukocytes (white blood cells) which reside within peripheral tissues, the bloodstream, and the lymphatic system. The first line of defense is provided by innate immunity which involves cells as well as anatomic and physiological barriers. Adaptive immunity provides a pathogen-specific response that develops during the lifetime of an individual. Principles of innate immunity are outlined in the following section, whereas a detailed description of adaptive immunity is given in Section 2.2.

Organisms have to protect themselves against a variety of pathogens such as viruses, bacteria, fungi, protozoa, and helminths. These pathogens vary widely in their size, their mechanisms of pathogenesis, and their ability to harm the host organism. The innate immunity

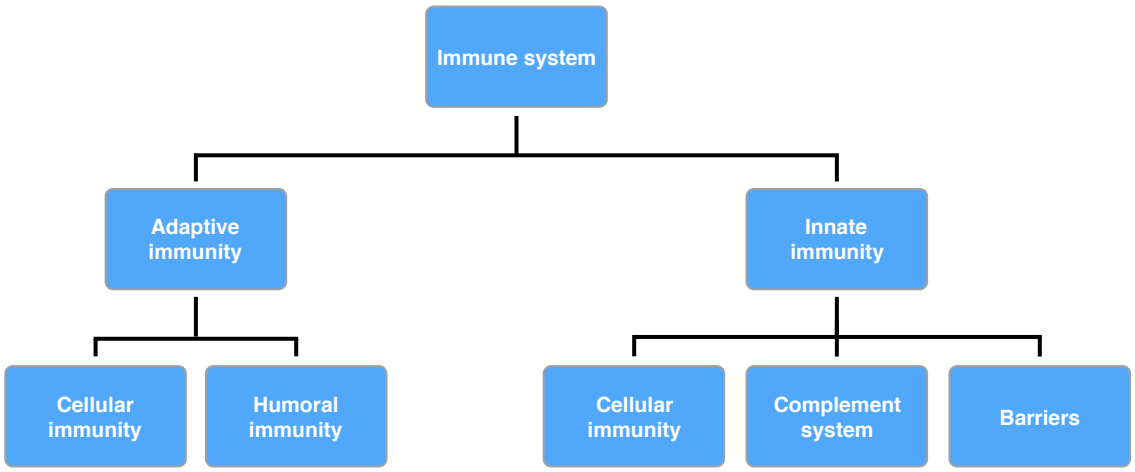

Figure 2.1: Overview of the immune system. The components of the immune system can be classified into innate and adaptive immunity. Innate immunity includes physical, chemical, and biological barriers. Additionally, it comprises the complement system and specific leukocytes, including natural killer cells, macrophages, and dendritic cells, as cellular components. The adaptive immune system involves the cell-mediated immunity (T lymphocytes) and the humoral immunity (B lymphocytes, antibodies).

possesses multiple levels of defense, whereby physical and chemical barriers constitute the first one. Epithelia impose a physical barrier and defend body surfaces against pathogens. The skin, gut, lungs, and facial cavities possess different mechanical, chemical, and microbiological prevention mechanisms to prevent pathogens from reaching internal tissues. Mechanical barriers include the longitudinal flow of air and fluid, movement of mucus, and tears. The arsenal of chemical barriers comprises fatty acids, α- and β-defensins, acidity, cathelicidin, and lysozymes. In addition to that, epithelial surfaces are associated with nonpathogenic bacteria, known as microbiota. The microbiota serves as the third barrier to infection by the production of antimicrobial substances or stimulation of epithelial cells. Additionally, a variety of antimicrobial enzymes and peptides are present in extracellular fluids, the blood, and epithelial secretions. Another component of innate immunity is the complement, which comes into play if the first barriers are crossed by pathogens. Complement is a group of more than 30 soluble plasma proteins. These proteins circulate in an inactive form through blood and other body fluids. Upon the presence of pathogens, complement pathways are triggered by pattern-recognizing receptors.

The innate immune cells, macrophages, granulocytes, mast cells, dendritic cells (DCs), and innate lymphoid cells (ILCs), such as natural killer (NK), ensure the cellular defense of innate immunity. Macrophages and neutrophils are the main cell types seen in the initial phase of innate immune responses. These cell types are capable of recognizing, ingesting, and destroying many pathogens without an adaptive immune response. The group of ILCs includes NK cells which are known to kill individual tumor cells or infected cells by releasing their

cytotoxic granules which contain granzymes and the pore-forming protein perforin. Due to the expression of various activating and inhibitory receptors, NK cells can distinguish between healthy and infected or abnormal cells. Thereby, the balance of signals determines if a cell will be killed or not. One example of surface proteins which are recognized by inhibitory receptors is MHC class I proteins. If MHC class I molecules are downregulated due to infection with viruses or other intracellular pathogens, NK cells are triggered by the signals from activating receptors and the absence of a signal from inhibitory receptors. Additionally, NK cells express Fc receptors. Upon binding of antibodies to these receptors, NK cells get activated and release their cytotoxic granules. This process is called antibody-dependent cellular cytotoxicity (ADCC).

DCs, macrophages, and neutrophils express pattern recognition receptors (PRRs) which recognize pathogen-associated molecular patterns (PAMPs). PAMPs are not part of the organism's host cells and include oligosaccharides, lipopolysaccharides, or unmethylated Cytosine-phosphatidyl-Guanine (CpG) DNA. The class of PRRs includes transmembrane proteins such as Toll-like receptors (TLRs) and cytoplasmic proteins, including nucleotide-binding oligomerization domain-like (NOD-like) receptors (NLRs). TLRs detect PAMPs of extracellular bacteria or phagocytized bacteria and activate different host defense signaling pathways like the nuclear factor kappa-light-chain-enhancer of activated B cells (NFκB) and the interferon-regulatory factor (IRF) pathway. PRRs activate pathways which induce pro-inflammatory cytokines such as tumor necrosis factor-α (TNF-α), interleukin-1β (IL-1β), and type I interferons (IFNs). Cytokines can act on the cell that releases the cytokines (autocrine), on adjacent cells (paracrine) or distant cells (endocrine). During inflammation, cells which reside in the blood migrate to the site of infection by cytokines and chemokines. Cytokines induce local effects, such as activation of lymphocytes, activation of vascular endothelium, increased antibody production, and activation of NK cells. Chemokines are small proteins belonging to the class of chemoattractant cytokines, inducing directed chemotaxis in adjacent cells.

Due to the combination of multiple layers of defense and different types of cells, the innate immune system is capable of detecting and destroying invaders within minutes to hours. The immediate (innate) immune response does not rely on the recruitment of antigen-specific lymphocytes. If pathogens cannot be eliminated by innate immunity, different effector mechanisms keep them in check until an adaptive immune response can be established.

2.2 The Adaptive Immune Response

The adaptive immune system provides effective responses against a wide range of pathogens through antigen-specific lymphocytes, namely *B lymphocytes* (B cells) and *T lymphocytes* (T cells). In the case of the adaptive immune response, high degrees of sensitivity and specificity are achieved by highly variable antigen receptors on the surface of cells and an extensive repertoire of highly variable binding sites of these receptors. Additionally, the adaptive immune

system provides immunological memory. If the body has been exposed to a pathogen once, there will be an immediate response to this pathogen at the time of next contact. There are two types of immune responses involving adaptive immunity. Humoral immunity is mediated by antibodies present in blood plasma and extracellular fluids. Antibodies, also called *immunoglobulins* (Igs), are highly specialized antigen-recognition glycoproteins. B cells possess a membrane-bound form of Ig, termed B-cell receptor (BCR). Upon antigen binding to the BCR, the B cell gets activated and differentiates into plasma cells, which secrete antibodies. Antibodies have two functions: they bind and neutralize pathogens and their toxins, mark them for destruction by phagocytes, and recruit other cells and molecules. There are five distinct isotypes of immunoglobulins, namely IgM, IgD, IgG, IgA, and IgE, which differ in structure and functionality. The second pillar of adaptive immunity is the cell-mediated immune response through T cells. Cellular immunity provides defense against infection or transformation of cells. T cells possess antigen-recognition molecules. *T-cell receptors* (TCRs) resemble the structure of BCRs; however, there is no soluble equivalent. TCRs induce intracellular signaling cascades resulting in the activation of the corresponding T cell. Another difference between BCR and TCR concerns the recognition of antigens, as TCRs do not recognize antigens itself. The recognition of an antigen by TCRs is only possible if peptides (antigen fragments) are bound to MHC molecules. The MHC is a large gene cluster encoding for these molecules. MHC molecules are cell surface glycoproteins that are highly polymorphic. As TCRs are peptide-specific, the formation of this complex adds another dimension to the complexity of antigen recognition and the specificity of the adaptive immune response.

2.2.1 The Major Histocompatibility Complex

The MHC is located on chromosome 6 in humans and on chromosome 17 in mice. In humans, the MHC genes are called *human leukocyte antigen* (HLA) genes, whereas in mice they are referred to as H-2 genes. The human gene cluster comprises more than 200 genes and spans more than 4 million base pairs (bp) (Figure 2.2).

The gene cluster is divided into three regions, namely class I, class II, and class III. There are three classical class I genes (HLA-A, HLA-B, and HLA-C) that encode the α chain of the respective HLA class I proteins. HLA-E, F, and G referred to as 'non-classical' HLA genes, encode HLA class Ib molecules. The class II region contains genes encoding the α and β chains of HLA class II molecules. These genes make up different pairs of class II α and β chain genes, called HLA-DR, HLA-DP, and HLA-DQ. There are four pairs of genes since the HLA-DR cluster contains two β chain genes and therefore makes up for two gene products paired with the DRα chain. Due to the polygeny of HLA, resulting in different HLA class I and class II genes, every individual carries a combination of at least three different HLA class I and three HLA class II molecules. The number of different HLA molecules is further increased since the HLA is highly

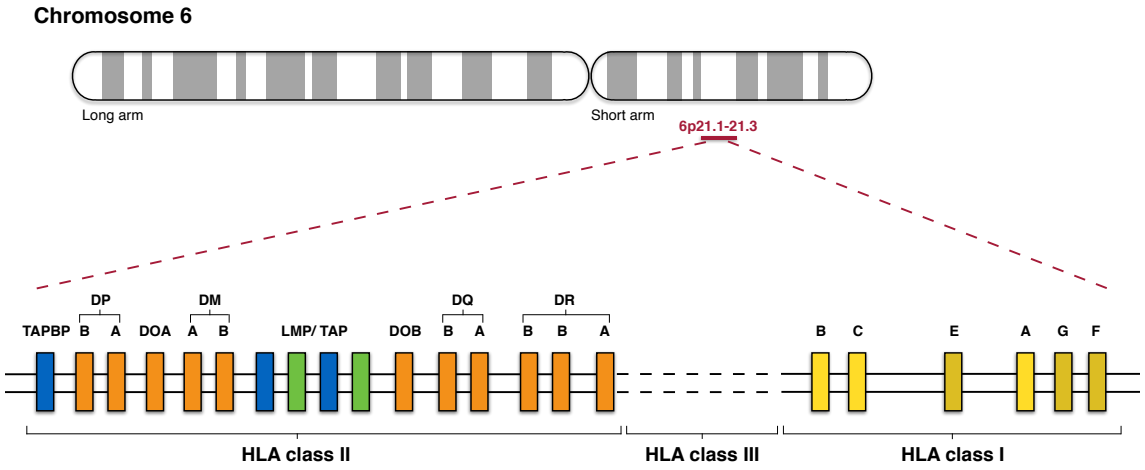

Figure 2.2: Simplified illustration of the genetic organization of the *major histocompatibility complex* (MHC). In humans, MHC is referred to as *human leukocyte antigen* (HLA) and located on chromosome 6. HLA class I genes (yellow), HLA class II genes (orange), and class III genes (not shown) are organized in separate clusters. The HLA class I cluster includes the classical class I genes (HLA-A, -B, and -C) and minor genes (HLA-E, -F, and -G). The genes encoding for HLA class II molecules (HLA-DR, -DP, and -DQ), as well as genes for the TAP1:TAP2 transporter (blue), and LMP genes (green) encoding for proteasome subunits, which are components of the antigen processing pathway make up the HLA class II cluster. Figure based on Janeway et al.[48].

polymorphic. Different HLA alleles encode for proteins that differ by up to 20 amino acids (AA). The substantial variation in the genes of HLA occurs due to point mutations and gene conversions. Consequently, there is a considerable variation in corresponding proteins as well. In fact, there are more than 20,000 alleles in humans for several HLA class I and II genes. In most cases, individuals are heterozygous for the MHC class I and II genes, meaning that the gene loci on both homologous chromosomes carry different alleles. The HLA haplotype describes the combination of HLA alleles on a single chromosome. HLA alleles are named according to a specific nomenclature[49]. The four-digit HLA type includes the gene name, followed by an asterisk, the allele group number, and the specific HLA protein number. Since the expression of HLA alleles is codominant, the products of both alleles at a locus are expressed similarly in a cell. Both HLA gene products, i.e., HLA class I and class II molecules, are closely related in structure but do differ in their subunit composition (Figure 2.3). HLA class I molecules are heterodimers built from four domains, whereas three of these domains are formed by the α chain. The other domain is formed by β_2-microglobulin which is encoded on chromosome 15. The membrane-spanning α chain is non-covalently associated with β2-microglobulin. The peptide-binding cleft of HLA class I molecules is formed by the folded α_1 and α_2 domains. The closed cleft consists of two α-helices and eight β-sheets. Significant differences across HLA class I molecules are located here, resulting in different specificities concerning peptide binding.

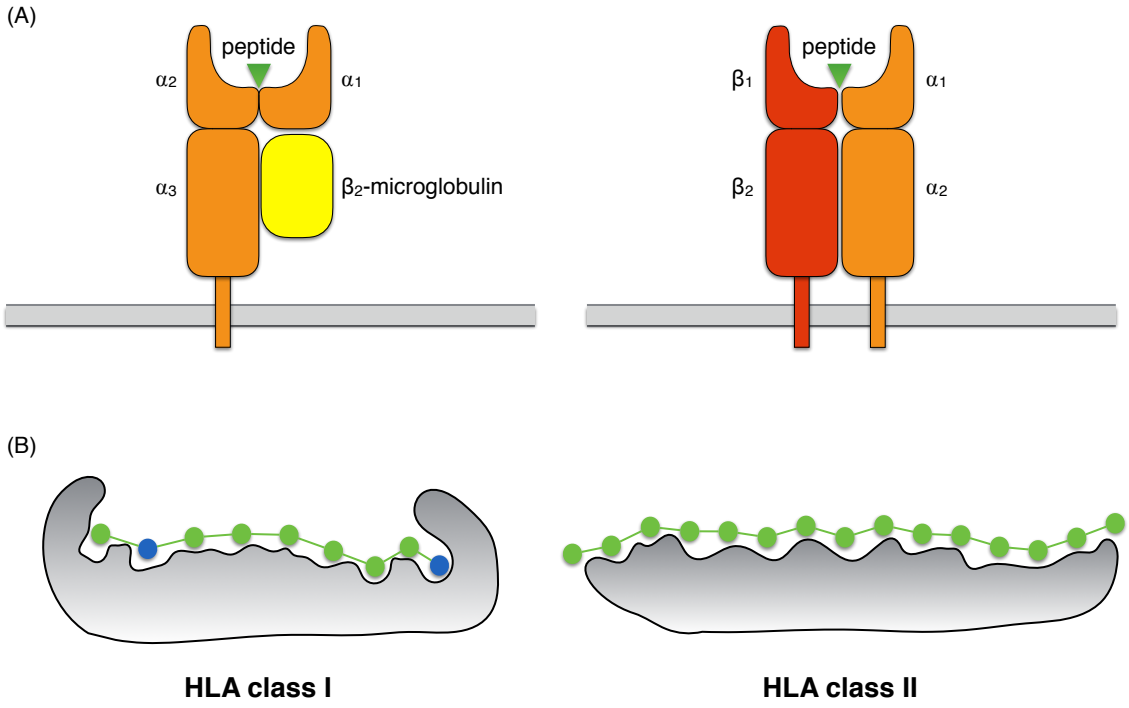

Figure 2.3: HLA class I and II molecules. (A) Simplified schematic structure of HLA class I (left) and class II (right) molecules. Both molecules consist of four domains, whereas class I molecules have three domains formed by α chains and one by β_2-microglobulin. HLA class II molecules are composed of α_1, α_2, β_1, and β_2 domains. Based on Janeway et al.[48]. (B) Differences in peptide-binding by HLA class I and class II molecules. Due to the structural differences, HLA class II molecules have an open peptide-binding cleft and allow peptides to protrude on either side. The closed structure of the cleft in HLA class I molecules restricts the length of peptides and forces all residues, including the anchor residues (blue), to fit into the pocket. Adapted from Toussaint et al.[50].

The same applies to HLA class II molecules which are composed of the two transmembrane glycoprotein chains α and β. Each of the two chains forms two domains resulting in a four-domain structure as with HLA class I molecules. Both chains of the noncovalent complex span the membrane and contribute to the formation of the peptide-binding cleft. Therefore, in contrast to HLA class I molecules, the peptide-binding cleft is formed by two domains of two different non-covalently bound chains. Due to the structure of the peptide-binding cleft, both ends of the cleft are open, allowing peptides to bind without tightly bound ends at either end of the cleft. Both classes of HLA molecules get stabilized upon peptide binding. HLA class I and class II molecules possess different specificities regarding peptide length resulting from differences in the structure of the binding cleft. HLA class I molecules bind peptides of eight to eleven AA in length. Longer peptides bind through kinking in the backbone. The backbone of peptides interacts with the HLA class I molecule through hydrogen bonds. Thereby, one cluster of tyrosine residues forms hydrogen bonds with the amino terminus, whereas a second

cluster forms hydrogen bonds with the C-terminal peptide backbone and itself. HLA class I molecules favor specific amino acid side chains, called anchor residues, at specific positions in the bound peptides. A sequence motif describes the set of anchor residues which are subject to peptide binding for a specific HLA allele. Anchor residues of peptides that bind to HLA class II molecules are located at various distances to the peptide end. In general, peptides binding to HLA class II molecules are at least 13 AA long, in most cases between 13–17 AA. Since the peptide-binding cleft of the molecules is not closed, a peptide can be longer and vary considerably in length. HLA class I and class II molecules do not only differ in the mode of peptide binding but also in their expression patterns. HLA class I molecules are expressed on all nucleated cells and especially highly expressed in hematopoietic cells. By contrast, HLA class II molecules are only present on professional *antigen-presenting cells* (APCs) including B cells, macrophages, and DCs, and epithelial cells of the thymus.

2.2.2 T Cell-Mediated Immunity

B and T cells both express antigen-specific receptors. In contrast to BCRs, TCRs only have one antigen-binding site and are strictly membrane-bound. TCRs are heterodimers, consisting of an α and a β polypeptide chain, which are linked by disulfide bonds and span the T-cell membrane. Each of the two chains comprises a variable amino-terminus and a constant region, whereas the antigen-binding site is formed by the variable regions. Binding specificity is given by the variable region of TCRs, which are clonally expressed. Since the genome cannot directly encode for a sufficient number of genes to generate the needed diversity of antigen receptors, different parts of the variable regions are encoded by gene segments. During the development of lymphocytes, a rearrangement of these gene segments occurs through a process called somatic DNA recombination. As a result, unique coding sequences are generated. In the case of T cells, the locus encoding the α chain contains V and J segments, while the TCRβ locus contains additional D gene segments. These gene segments rearrange to variable domain exons during the development of T cells. The diversity is further increased by the addition of nucleotides between V and J gene segments of rearranged TCRα chains. The loci encoding α and β chain additionally contain one and two gene segments for the constant region respectively. Through the combination of the different α and β gene segments and junctional diversity, an estimated total diversity of 10^8 is reached. Most of the diversity is located in the CDR3 loops, encoded by D and J segments, which form the center of TCR binding sites. This region is mainly in contact with bound unique peptide fragments of peptide-HLA (pHLA) complexes. In contrast to Igs, T cells only recognize antigens if presented on the hosts' cells HLA molecules. Antigens can be *inter alia* of pathogenic origin or originate from tumors. There are two major classes of naïve T cells which differ in the expression of cell surface molecules involved in the recognition of pHLA complexes. Cytotoxic T cells express the cell surface protein cluster

of differentiation 8 (CD8), while helper T cells express the protein CD4. Both co-receptors are in direct contact with HLA molecules and contribute to the overall effectiveness of the response. CD4 and CD8 are both associated with the TCR during antigen recognition and bind to the invariant sites of the pHLA complex simultaneously. CD8 recognizes HLA class I molecules, whereas it is HLA class II for CD4. The co-receptors have a distinct structure. CD4 is a single-chain protein of four Ig-like domains, whereas CD8 consists of two distinct Ig-like domains linked by a disulfide bond. $CD4^+$ T cells can be further categorized into subgroups according to their differentiation. There are five main subtypes of $CD4^+$ effector T cells: Th_1 cells activate macrophages through IFN-γ, Th_2 cells produce cytokines to activate eosinophils and mast cells, Th_{17} cells secrete cytokines to activate epithelial and stromal cells, T_{fh} cells activate B cells, and T_{reg} cells suppress T cell and innate immune cell activity. Some of the T cells (and B cells) become memory cells to provide long-lasting immunity after disease or vaccination. Memory cells can differentiate into effector cells after a subsequent encounter of the corresponding antigens. The differentiation of naïve $CD4^+$ T cells into distinct subclasses is a difference to $CD8^+$ T cells which all differentiate into CD8 cytotoxic T lymphocytes (CTLs). Both types of naïve T cells differentiate upon recognition of peptides presented on HLA class I or class II molecules respectively. HLA class I presents peptides from intracellular antigens, which can be recognized by $CD8^+$ T cells. Upon recognition of foreign peptides, CTLs can kill the corresponding infected or transformed cell. The killing mechanism involves the release of the cytotoxic proteins granzymes, perforin, and granulysin. Granzymes induce apoptosis in cells, perforin forms pores to deliver granzymes into the target cell, and granulysin has antimicrobial activity. Besides, the membrane-bound Fas ligand can induce apoptosis by binding to Fas receptor. Additionally, CTLs release the cytokines IFN-γ, TNF-α, and LT-α to inhibit viral replication, increase HLA expression, and activate macrophages. Peptides of extracellular antigens are presented on HLA class II molecules, recognized by $CD4^+$ T cells. The generation of peptides from native proteins (antigen processing) and the presentation of peptides on the cell surface on HLA molecules (antigen presentation) differs for the two HLA classes (Figure 2.4). Peptides presented to T cells can either originate from antigens derived from the cytosol or vesicular compartments. Peptides originating from the cytosol are transported into the endoplasmatic reticulum (ER) and ultimately loaded on HLA class I molecules. Proteins in cells which are tagged by the ubiquitin-proteasome system (UPS) get continually degraded by the proteasome. The proteasome is a protease complex consisting of a 20S catalytic core and two 19S regulatory caps at each end. The 19S component of the proteasome recognizes ubiquitin molecules which are attached to proteins by the UPS through a process called ubiquitination. Further, trimming of peptides might occur by the enzyme endoplasmic reticulum aminopeptidase associated with antigen processing (ERAAP). The proteasome can exist as constitutive proteasome and as immunoproteasome. The constitutive proteasome is present in all cells, whereas the immunoproteasome is present in cells which have been stimulated with

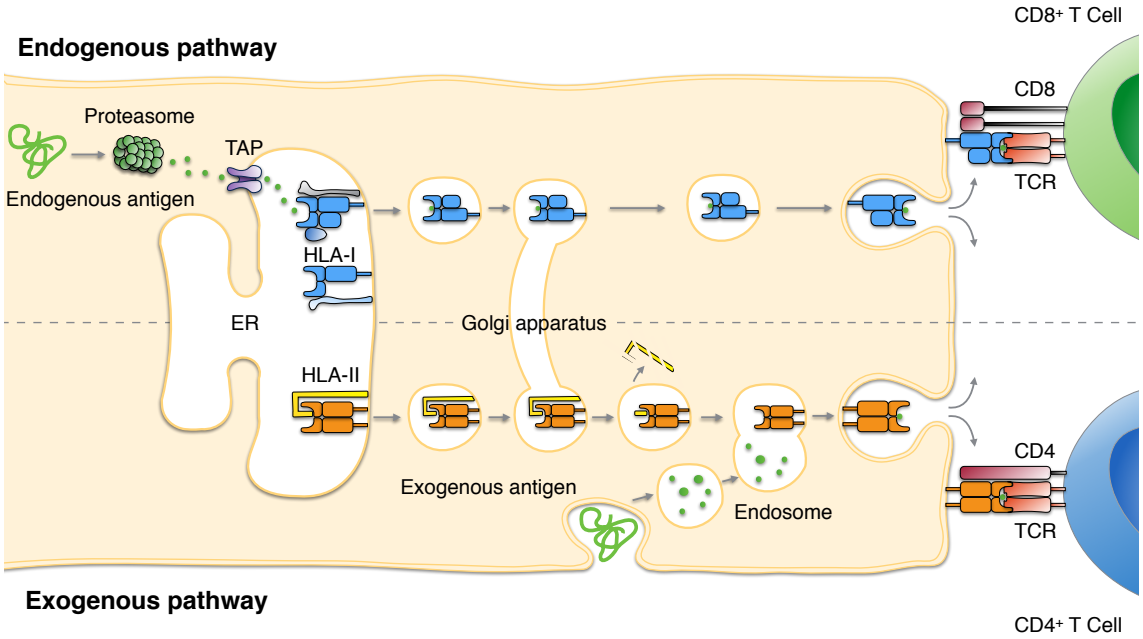

Figure 2.4: Endogenous pathway (top): Endogenous antigens get degraded by the proteasome and transported via TAP into the rough endoplasmic reticulum. First, HLA class I molecules bind to calnexin. After dissociation of calnexin, $\beta 2$-microglobulin, calreticulin, and tapasin bind. Peptide-loaded HLA class I molecules are transported to the cell membrane through the Golgi complex. Exogenous pathway (bottom): HLA class II molecules are bound to invariant chain (li). During the transfer through the Golgi apparatus, li is degraded, whereby the CLIP fragment remains. Exogenous antigens are taken up and degraded in endosomes. Exchange of CLIP for peptides is mediated by HLA-DM. The complex is then transported to the cell membrane. HLA class I- and II-peptide complexes are presented to $CD8^+$ and $CD4^+$ T cells, respectively. Figure based on Janeway et al.[48].

interferons. Interferons trigger the replacement of the three proteolytic subunits $\beta 1$, $\beta 2$, and $\beta 5$ in the catalytic chamber. As a consequence, the enzymatic specificity of the immunoproteasome changes. The cleavage rate after hydrophobic residues increases, while the cleavage rate after acidic residues decreases. The presence of a free terminal α-carboxyl group is a preferred anchor residue for binding to HLA class I molecules and beneficial for the transport of peptides from the cytosol to the lumen of the ER. Two ATP-binding cassette (ABC) proteins, namely transporter associated with antigen processing-1 and -2 (TAP1 and TAP2) form a heterodimer and are associated with the ER membrane. The TAP1:TAP2 complex is an ATP-dependent peptide transporter which carries cytosolic peptides into the ER. Only peptides of 8–16 AA in length with hydrophobic or basic residues at the carboxy terminus are transported by the TAP complex. Therefore, the specificity of TAP matches the features of peptides binding to HLA class I molecules. HLA class I molecules reside in a partially folded state in the ER. During this unbound state, the α chain is associated with the chaperone protein calnexin. Upon binding of β_2-microglobulin, calnexin dissociates from the α:$\beta_2 m$ complex. The α:$\beta_2 m$ complex binds

a complex of calreticulin and ERp57 and to TAP through the TAP-associated protein tapasin. Binding of peptides releases the partially folded HLA class I molecule from the peptide-loading complex, consisting of calreticulin, ERp57, and tapasin. The completely folded pHLA complex leaves the ER and is transported to the cell surface through the Golgi apparatus. DCs can present peptides which have not been generated within the own cytosol through this exogenous pathway of HLA class I molecule loading. This process is referred to as cross-presentation. Antigens originating from intravesicular pathogens such as certain bacteria which replicate inside intracellular vesicles or extracellular pathogens and toxins are presented through the HLA class II antigen processing pathway. Thereby, exogenous proteins are taken up by APCs via endocytic vesicles. Peptides of these agents are ultimately bound to HLA class II molecules and presented to $CD4^+$ T cells. Extracellular pathogens and proteins are internalized into endocytic vesicles either through receptor-mediated endocytosis by Igs on B cells, through phagocytosis by macrophages and DCs, or autophagy. The proteins taken up by endocytosis are transported through endosomes with decreasing pH. Additionally, the fusion with lysosomes containing proteases leads to antigen degradation. HLA class II molecules are first translocated into the ER. To prevent premature binding to peptides and misfolded proteins molecules are bound with a type II membrane glycoprotein called MHC class II-associated invariant chain (li, CD74). A subsequence of li, referred to as class II-associated invariant chain peptide (CLIP) is bound along the peptide-binding groove. Through the association with li, HLA class II molecules are targeted for delivery to a low-pH endosomal compartment. When the HLA class II:li complex enters the endosomal pathway, proteases cleave li, leaving CLIP bound to the molecule. To enable endocytosed antigens, degraded to peptides through acidified endosomes in the cytosol, to bind to the HLA class II molecules, the MHC class II-like molecule HLA-DM binds to the HLA-II:CLIP complex. HLA-DM catalyzes the release of CLIP and binding of antigenic peptides. pHLA-II complexes are then presented on the cell surface.

2.2.3 Immunological Tolerance and Autoimmunity

The adaptive immune system possesses a variety of effector mechanisms to provide defense against pathogens. Cells have to be able to distinguish between self and non-self to prevent the immune system from directing these mechanisms against self-antigens, causing tissue damage. The response to self-antigens is referred to as autoimmunity. Loss of self-tolerance can lead to a variety of autoimmune diseases such as rheumatoid arthritis, multiple sclerosis, and Crohn's disease. The induction of tolerance to self can be classified into central and peripheral tolerance. Central tolerance induces self-tolerance in lymphocytes during the development in the thymus (T cells) and bone marrow (B cells). During the development of T cells in the thymus, cells express low TCR, CD4, and CD8 levels. These double-positive cells interact with self-peptides presented on HLA molecules in the thymic stroma during further development.

If double-positive lymphocytes can recognize self-peptide-self-HLA complexes presented by thymic cortical epithelial cells, they are positively selected. Consequently, the cells mature into $CD4^+$ or $CD8^+$ T cells. The repertoire of T cells is further modulated by the deletion of cells which react too strongly to self-antigens. This process, referred to as negative selection, is facilitated by the autoimmune regulator (AIRE) protein. AIRE promotes the thymic expression of tissue-specific antigens from other organs, enabling negative selection. The negative selection of double-positive thymocytes having strong reactivity to pHLA complexes takes place in the thymic cortex. Immature CD4 or CD8 single-positive thymocytes that receive strong TCR signaling through recognition of pHLA complexes are negatively selected in the thymic medulla. The process in the medulla is mediated by medullary epithelial cells, bone marrow-derived macrophages, or DCs and leads to the deletion of strongly autoreactive lymphocytes. A corresponding process exists for immature B cells which takes place in the bone marrow. After positive and negative selection, the repertoire of T cells includes HLA-restricted but self-tolerant cells. The processes of central tolerance are complemented by mechanisms of peripheral tolerance after lymphocytes left the central or primary lymphoid organs. Therefore, peripheral tolerance ensures tolerance against antigens which are not expressed in the thymus or bone marrow. Mature autoreactive lymphocytes that migrate to the periphery are either deleted by activation-induced cell death or set in a permanent state of unresponsiveness (anergy) if they react to self-antigens. The deletion or inactivation occurs due to missing co-stimulatory signals, such as inflammatory cytokines (IL-6 and IL12) and co-stimulatory molecules (B7.1), which are not present without infection or inflammation. Autoreactive lymphocytes in the periphery can also be inhibited by regulatory T cells (T_{reg}). T_{reg} cells can either be programmed in the thymus (nT_{reg}) or in the periphery (iT_{reg}). Both types express the transcription factor *FoxP3* in response to self-antigen recognition. In the periphery, the induction occurs in the presence of TGF-β and absence of pro-inflammatory cytokines. If these cells get in contact with the same antigen in the periphery, they inhibit other self-reactive T cells, recognizing antigens in the same tissue, to prevent their differentiation. The inhibition is mediated through production of the cytokines IL-10 and TGF-β. Furthermore, T cells can undergo functional deviation. In this case, T_{reg}-cell development is induced instead of effector T-cell development.

2.3 Cancer

This section introduces the biological background of cancer and gives an overview of the history of cancer treatment and state-of-the-art therapeutic options. For a comprehensive introduction, the reader is kindly referred to Robert A. Weinberg[51] and Mendelsohn et al.[52].

2.3.1 Introduction

The term cancer describes a group of diseases that can affect different sites of the body characterized by abnormal cell growth. Cell proliferation is essential for growth and survival of eukaryotic organisms. It is the process which ultimately leads to an increase in the number of cells through cell division. In the growth phases, the processes undergo checkpoints to prevent cells containing damaged DNA from advancing to the synthesis phase. DNA damage might occur because of irradiation or chemical modifications. Further, checkpoints prevent cells showing signs of damaged DNA or erroneous replication from entering mitosis. Typically, cells are responsive to internal and external modulatory signals. Therefore, cells replicate upon specific growth signals and stop replication accordingly and die due to *programmed cell death* (apoptosis). Due to mutations in genes associated with the regulation of the cell cycle and cell proliferation, cells proceed through the cell cycle in an unregulated manner. In addition to inherited germline mutations, DNA damage can be acquired that might drive the transformation of normal cells into tumor cells. Three groups of external agents are associated with the risk of acquiring mutations[52]: physical carcinogens (ionizing radiation), chemical carcinogens (asbestos and tobacco smoke), and biological carcinogens (infections from oncogenic viruses, bacteria, or parasites). The transformation leads to abnormal growth, also referred to as neoplasm, that ultimately forms an abnormal mass of tissue, defined as a tumor. Tumors may be benign or malignant. Malignant or cancerous tumors do not reside at the tissue of origin. They can invade neighboring healthy tissue or even spread to distant sites in the body through blood vessels or lymph systems. The latter process is defined as metastasizing and causes the formation of tumor mass in other organs (metastasis). Metastasis is the primary cause of cancer-associated mortality. In 2015, 8.8 million people died of cancer which makes it the second leading cause of death worldwide after cardiovascular diseases[53]. There are more than 100 distinct types of cancer that are defined by the location of occurrence of the primary tumor. Additionally, subtypes of tumors can be found in specific organs depending on the specific cell type of origin. Lung, liver, colorectal, stomach, and breast cancer are among the most common causes of cancer-associated fatalities. Besides the tremendous health burden, cancer has a significant and increasing economic impact. The estimated total annual financial cost of cancer was estimated at approximately US$ 1.16 trillion in 2010[54].

2.3.2 Genetics of Cancer

The development of cancer presupposes the transformation of normal cells into tumor cells. This multistage process occurs mainly as a result of acquired genetic variations in addition to the genetic predisposition of an individual. Cancer-associated genetic variations affect three classes of genes: *proto-oncogenes*, *tumor suppressor genes*, and genes that are responsible for DNA damage repair.

Proto-oncogenes, such as *RAS*, encode for proteins appropriately driving progression through the cell cycle, which are therefore responsible for the regulation of cell growth and proliferation. Consequently, mutations in these genes might lead to cells proceeding through the cell cycle in an unregulated manner. *Oncogenes* encode constitutively active and overexpressed versions of normal cellular proteins that are involved in cell growth and proliferation. In general, retroviruses or somatic mutations can turn proto-oncogenes into oncogenes. These mutations can be either base substitutions, gene copy number amplifications, or chromosomal translocations. Oncogenes also possess the ability to drive oncogenic transformation due to overexpression.

The second class of affected genes, tumor suppressor genes, are scattered throughout the genome and provide different mechanisms to control proliferation[55]. The encoded proteins prevent unscheduled proliferation, stimulate cell death, and trigger the initiation of permanent cell cycle arrest[56]. They act as negative regulators of proto-oncogenes and ensure regular cell division under normal and stress-induced conditions leveraging cell cycle checkpoints[57]. In tumors, tumor suppressor gene function is frequently diminished. Genes that are maintaining genome stability are affected during tumor progression. Genomic stability is supported through the recognition of acute genomic damage and the subsequent recruitment of enzymatic DNA damage repair complexes[58]. Inactivation of these processes leads to an increased probability of mutations, also affecting oncogenes and tumor suppressor genes[56]. Defects in repair genes, as well as tumor suppressor genes, can occur due to somatic mutations. Besides somatic mutations, epigenetic mechanisms such as DNA methylation can also – and even more frequently – lead to defects in tumor suppressor or DNA damage repair genes[59].

Genetic alterations drive and expedite the progressive transformation of healthy into malignant cells. During this multistep process of cancer cell and tumor development, respective cells acquire a set of biological capabilities. Hanahan and Weinberg defined this set of essential alterations in cell physiology that induce malignancy and drive malignant growth as hallmarks of cancer[2,60]. Multistep tumor pathogenesis involves genomic instability and tumor-promoting inflammation, whereby cancer cells do typically evade growth suppressors, sustain proliferative signaling, avoid cell death, induce angiogenesis, enable replicative immortality, activate invasion and metastasis, avoid immune destruction, and deregulate cellular energetics.

Cancer cells possess the ability to sustain chronic proliferation through the production of growth factor ligands, the expression of cognate receptors, up-regulation of growth-factor receptors, or ligand-independent activation of receptors due to structural alterations. Alternatively, cancer cells can stimulate supply with growth factors through cells in their surrounding[61].

The insensitivity to growth-inhibitory signals does mainly rely on lost functions of tumor suppressor gene proteins such as the retinoblastoma protein (RB) and the tumor protein p53 (encoded by *TP53*)[62]. Another essential alteration is the ability of cancer cells to evade programmed cell death (apoptosis). This control mechanism normally represents a barrier to cancer development since it triggers apoptosis of cells affected by oncogenic mutations[63].

Besides mutations, multiple non-genetic mechanisms such as angiogenesis accelerate the tumor progression. Tumor cells can recruit blood vessels under the involvement of different heterotypic interactions between cancer cells, their mesenchymal microenvironment, and mechanisms like the modulation of vascular endothelial growth factor[64].

Another emerging hallmark of cancer is the ability of tumor cells to evade the immune system[60]. The interactions and underlying mechanisms between the immune system and neoplasms are still not fully understood. However, the immune system accounts for three primary roles in the prevention of tumors[65]. The most obvious mechanism is the protection of the immune system against virus-induced cancers due to its capability to eliminate and suppress viral infections. Immunocompromised individuals as a result of immunosuppression after organ transplantation or immunodeficiency syndromes show increased rates of virus-induced malignancies. In general, by eliminating pathogens, the immune system avoids chronic inflammation and therefore prevents the establishment of an inflammatory environment which is promotive of tumorigenesis[66]. The third primary mechanism is the identification and elimination of tumor cells based on expressed *tumor-specific antigens* (TSAs). Therefore, immune cells are capable of identifying transformed cells that escaped the cell-intrinsic tumor suppressor mechanisms[65]. This direct interplay of immune cells and tumor cells is a dynamic process that is composed of three distinct phases: elimination, equilibrium, and escape[67,68]. In the elimination phase, also referred to as immunosurveillance, both the innate and adaptive cellular part of the immune system contribute significantly to immune monitoring and thus tumor cell elimination[68]. These processes are mainly mediated by CTLs, Th1 helper cells, NK cells, and through the secretion of inflammatory cytokines such as IL-12 and IFN-γ. In general, the outcome of the disease seems to correlate with the immune cell infiltration in tumors[69], what has been shown, *inter alia*, for ovarian and colon carcinoma[70]. During the elimination phase, cancer cells which are highly immunogenic are routinely eliminated. Tumor cells that escape the elimination stage enter an equilibrium phase, where the immune system controls tumor cell growth. Due to an active immunoediting process in the tumor cell population or changes in the host immune system such as immunosuppression, cells can progress into the escape phase and grow in an immunologically unrestricted manner[65]. During this process, various mechanisms are active within the tumor microenvironment to escape immune surveillance and elimination. APCs and T cells are suppressed within the tumor microenvironment due to the production of cytokines such as TGF-β or chemokines, including IL-8 and IFN-γ or the colony-stimulating factor 1[71,72]. Additionally, T_{regs} are recruited, restricting the proliferation and activity of immune cells[73]. Cancer cells also employ modulated expression mechanisms to evade recognition through immune cells. The overexpression of CD47 prevents the recognition by circulating macrophages[74], whereas a downregulated expression of HLA class I molecules precludes the recognition by T cells[75]. At the endpoint of tumor progression, cancer cells ac-

quire the ability to invade and metastasize to distant sites in the body through blood circulation and the lymphatic system[76].

2.3.3 Cancer Treatment

Cancer treatment regimens depend on multiple factors such as the type of cancer, the progression status of the disease, and the genetic constitution of the patient. Traditional strategies of cancer treatment include surgery, radiotherapy, and chemotherapy. Surgery aims at local removal of solid tumors. Depending on the risk of damaging nearby organs, tumors might be resected completely or debulked. Besides the traditional surgery, more recent techniques such as cryosurgery or lasers are applied.

In radiotherapy, high doses of ionizing radiation (X-rays, gamma rays, and charged particles) directly or indirectly (via water molecules) ionize the atoms of DNA molecules, resulting in DNA damage and cell death. Surgery and radiotherapy are often followed up by adjuvant chemotherapy. Since the 1940s, cytotoxic drugs have been used to treat cancer. Most of the early drugs interfered with DNA replication and cell division. One of the first drugs used for chemotherapy was the so-called nitrogen mustard, a DNA-alkylating agent[77]. Another early target for chemotherapeutics was the enzyme dihydrofolate reductase (DHFR), part of the folic acid metabolism and required for DNA synthesis. Targets of conventional chemotherapeutics include enzymes of DNA synthesis, microtubules, and growth factor receptors which have primarily cytotoxic effects that also affect healthy cells of the patients. Due to gained knowledge about oncogenes and tumor suppressor genes, as well as new technologies, there was a shift towards aberrantly functioning gene products as targets. These targets have been identified to contribute to the malignant phenotype of cancer cells. The application of such drugs is also referred to as targeted therapy.

Targeted therapies include *monoclonal antibodies* (mAbs) such as Cetuximab, approved for colorectal cancer[78], and *kinase inhibitors* such as Vemurafenib, approved for the treatment of patients with advanced melanoma expressing a mutated *BRAF* gene (V600E)[79]. However, Vemurafenib is not effective for patients suffering from colon cancers with the same genetic aberrations[80]. In contrast, the mAbs Cetuximab and Panitumumab, targeting the epidermal growth factor receptor, are not effective in colon cancers with a mutation in *K-RAS*[81]. These two examples show that some mutations are beneficial or even necessary for specific therapies to be effective, whereas other mutations prevent therapies from being effective. Therefore, standard treatments already include the screening for genetic and molecular biomarkers. The identification of new genetic aberrations which are cancer drivers and potential targets for new therapeutics is of clinical importance. Cancer driver genes, classified as tumor suppressor genes and oncogenes, are known to be critical for cancer progression due to their function in cell proliferation, differentiation, senescence, and apoptosis[82]. Therefore, targeted therapies

mainly aim at blocking cell proliferation, promotion of cell cycle regulation, and induction of apoptosis. Druggable targets include tyrosine kinases which can be targeted by antibodies against associated cell surface receptors or through intracellularly acting kinase inhibitors of low molecular weight. Imatinib (Gleevec), a prominent tyrosine kinase inhibitor, targets the product of the *BCR-ABL* gene translocation in patients with chronic myelocytic leukemia and gastrointestinal stromal tumors[83,84].

Another group of therapeutic options, immunotherapy, exploits the host's immune system to treat cancer. Immunotherapy is generally based on the fact that the immune system can eliminate neoplasia during the initial growth phase through a process referred to as immune surveillance[85]. Due to the expression of non-self antigens (*neoantigens*), cells of the immune system can detect neoplastic cells. However, cancer cells possess multiple immune escape mechanisms, referred to as *cancer immunoediting*. As the consequence of immune selection pressure, cancer cells may become less immunogenic, insensitive to immune effector mechanisms, and establish an immunosuppressive state within the tumor microenvironment[86].

Still, multiple immunotherapeutic approaches exist to modulate the immune response against cancer cells. An overview of existing cancer immunotherapies is shown in Figure 2.5. Cancer immunotherapy approaches can be broadly classified as passive and active[87]. The classification is dependent on the therapy's ability to activate the host immune system.

Passive therapies include the adoptive transfer of lymphocytes activated *ex vivo* and the administration of mAbs directed against cancer cells. Approaches to reduce cancer-induced immunosuppression and vaccine approaches directed against tumor antigens, inducing specific immune responses are considered active. Vaccine approaches will be discussed in Section 2.4.

The adoptive transfer of T and NK cells is also referred to as adoptive cell therapy (ACT)[89]. It utilizes the ability of lymphocytes to destroy primary and metastatic tumor cells. In one approach, autologous peripheral or tumor-infiltrating lymphocytes are expanded *ex vivo* and reinfused[90]. Approaches using mixtures of $CD8^+$ and $CD4^+$ T cells grown from resected metastasis (TILs) showed good response rates and tumor regression for melanoma patients in clinical trials[91]. Besides TILs, genetically engineered tumor-reactive lymphocytes are a treatment option. These can be equipped with chimeric antigen receptors (CARs) that are built from variable Ig domains and the constant domain of TCRs[92]. Therefore, lymphocytes possess the non-HLA-restricted antigen-recognition property of antibodies[93].

Immunostimulatory mAbs represent another class of agents. These antibodies target different surface molecules and eliminate tumor cells through CDC, ADCC, and the induction of apoptosis. Several antibodies for different kinds of cancer are currently being tested in clinical trials or are already on the market. Rituximab is one antibody directed against CD20 expressed on leukemia and lymphoma cells[94,95].

Another approach referred to as immune checkpoint blockade, targets molecular and cellular mediators of cancer-induced immunosuppression. Targets of checkpoint inhibitors include

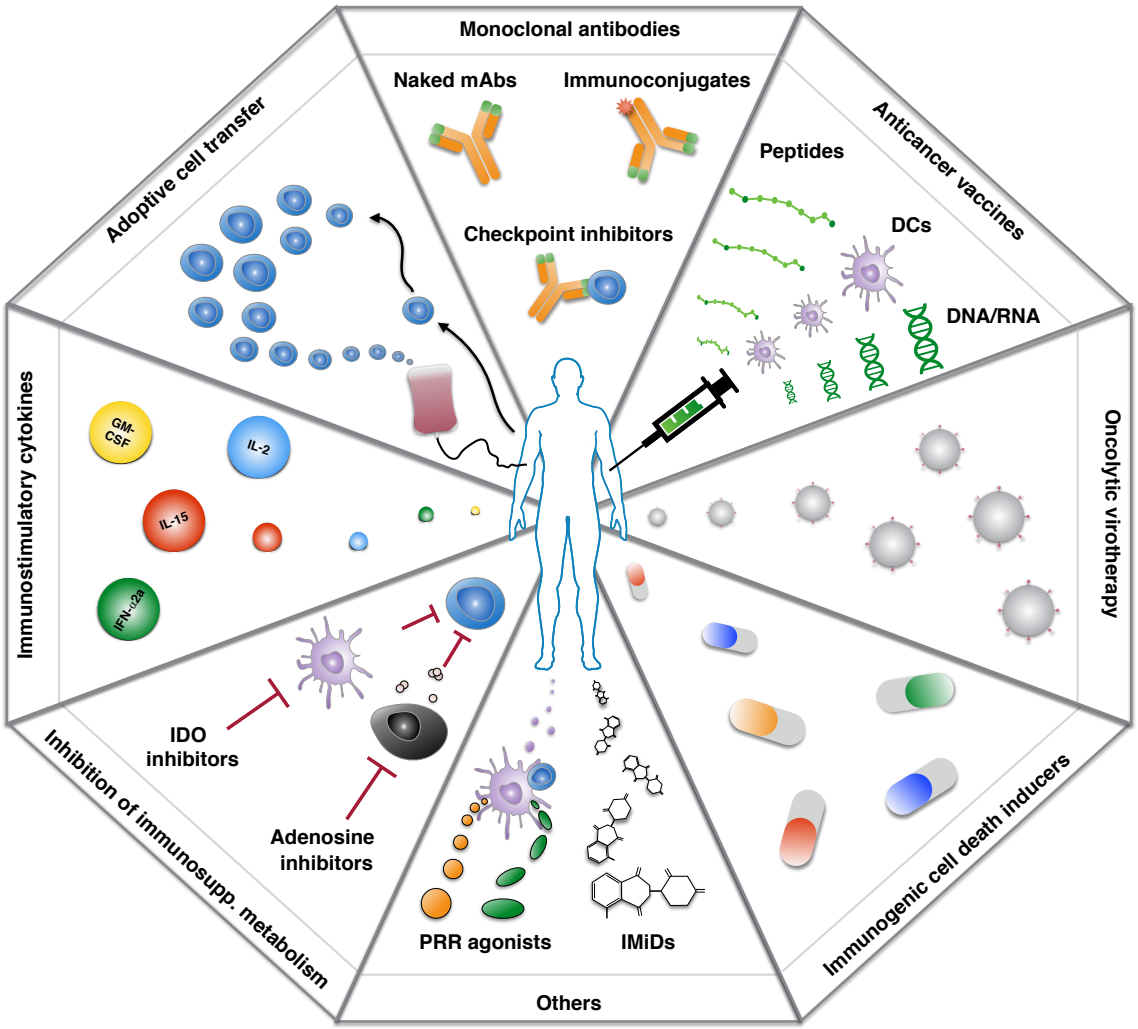

Figure 2.5: Cancer immunotherapy approaches. Therapeutic options include tumor-targeting or immunomodulatory mAbs, indoleamine (IDO) inhibitors, inhibiting the immunosuppressive metabolism, and pattern recognition receptor (PRR) agonists, such as NOD-like receptor agonists or TLR agonists. Anticancer vaccines can be DC-, peptide-, whole-tumor cell, tumor lysate, DNA-/RNA-based. Human body silhouette icon obtained from Reactome Icon Library[88] and adapted. Figure based on Galluzzi et al.[87]

the CTL-associated protein 4 (CTLA-4)[96], an inhibitory receptor downregulating T-cell activation, and the programmed cell death protein 1 (PD-1), which influences T-cell proliferation, cytokine release, and cytotoxicity[97].

In many cases, monotherapeutic approaches have shown to be relatively ineffective. Therefore the combination of different therapies is assessed at high rate[98]. Here, combinations do not solely contain immunotherapeutic approaches but combinations with conventional treatments such as chemotherapy. The combination can enhance antitumor effects of immunotherapy, further improved through tumor cell death and release of tumor (neo)antigens[99]. In combi-

nation with targeted therapies, the same beneficial effects with less severe side effects can be achieved [100].

2.4 Epitope-based Vaccines

This section includes a brief history of vaccination and introduces the more recent development of epitope-based vaccines. Further, applications in cancer therapy and personalized medicine are discussed.

2.4.1 Introduction

Vaccination is used to generate long-lasting and protective immunity. The effect of prior infections on the subsequent predisposition to acquire disease has already been perceived 2000 years ago. Back then, the Greek historian Thucydides noticed during the Peloponnesian War that people who survived an infection during the first outbreak of the plague were not affected by the second outbreak. In the 18^{th} century, it was common practice in China and the Middle East to use small amounts of smallpox pustules to protect people from infections. The procedure caused mild infections but long-lasting protection against the disease. This procedure is today referred to as variolation. In 1798, Edward Jenner discovered that the infection with cowpox provided protection against smallpox and gave experimental proof by successfully testing his hypothesis on several subjects. The immunization procedure was referred to as vaccination due to the Latin translation of cow (*vacca*) and cowpox (*vaccinia*). Later, Louis Pasteur described the concept that vaccination could be applied to achieve immunization against any pathogen. He extended Jenner's approach to other infectious agents like chicken cholera, rabies, and anthrax.

Vaccines are one of the most important contributions of public health in the past 100 years. Due to the early findings by Edward Jenner and Louis Pasteur, deaths from infectious disease could be decreased, and others, including smallpox, could even be eradicated. Many different approaches to the development of vaccines exist. Whole-organism vaccines either consist of killed, inactivated, or (living) attenuated organisms. Other vaccines make use of purified antigens like inactivated exotoxins or capsular polysaccharides (subunit vaccines), bacterial toxins (toxoid vaccines), microbial DNA introduced by viruses or bacteria (recombinant vector vaccines), mRNA-encoded antigens (RNA vaccines), and DNA-encoded antigens (DNA vaccines). Effector mechanisms include the activation of $CD4^+$ and $CD8^+$ T cells, which establishes direct protection by cytotoxic T cells and shaped antibody responses through $CD4^+$ T cells. Protective immunity caused by vaccines is mostly acquired through the induction of antibodies. Therefore, subsequent infections of cells can be prevented through neutralization. Protection against some pathogens requires additional cell-mediated responses through $CD8^+$ T cells or the presence

of preexisting antibodies, as in the case of the poliovirus. Different vaccine approaches have in common that ultimately an epitope is being presented to the immune system. However, the mechanisms leading to the epitope generation before presentation differ due to the difference in administered vaccine components.

A rather new rational design and optimization strategy of vaccines, which does not require the whole organisms in a killed or attenuated form uses T-cell epitopes directly[101–103]. *Epitope-based vaccines* can be either used as prophylactic vaccines in a traditional sense to protect individuals from an infective agent or as therapeutic vaccines to reduce or arrest disease progression. Both approaches aim at the stimulation of T cells and the generation of immunological memory. Due to the absence of infectious material, there is, in contrast to attenuated vaccines, no risk of reversion and thus no risk of pathogenicity.

However, epitope-based or synthetic antigen-based vaccines are often less immunogenic than whole organism vaccines. To maximize the potency of epitope-based vaccines, different adjuvants and modes of delivery are employed[104] to boost or modify the immune response. One approach is the covalent or non-covalent attachment of peptides to biological macromolecules. These include proteins, such as heat-shock proteins which interact with the innate immune system and thus are capable of modulating the immune response[105] and proteins which are ligands of receptors on APCs such as TLRs[106]. Self-adjuvanting vaccines make use of lipopeptides which interact with APCs expressing TLRs and induce DC proliferation[107]. Another approach utilizes recombinant cytokines, such as granulocyte-macrophage-colony-stimulating factor (GM-CSF)[108] and IL-12[109] as adjuvants. The class of oil-emulsion type adjuvants is mostly used for *therapeutic cancer vaccines*. Montanide is one example of oil-based adjuvants which have shown to have beneficial effects on immunogenicity and has been approved for therapeutic vaccines[110]. One of the most widely used and accepted adjuvants in humans is Alum (aluminum hydroxide)[111]. Alum showed to induce potent antibody responses. However, it has proved to be ineffective for some antigens and to have limited capacity to augment cell-mediated immune responses[112]. In other approaches, antigen delivery systems with additional immunostimulating activity are used. Particulate delivery systems[113] include immunostimulatory complexes (ISCOMs)[114], hollow spherical constructs of phospholipid bilayers (liposomes)[115], membrane vesicles secreted from epithelial and hematopoietic cells (exosomes)[116], and spherical unilamellar lipid membrane vesicles embedded with viral membrane proteins (virosomes)[117]. Particulate delivery systems have shown to increase effective uptake by APCs in comparison to antigens in solution[118]. Cell-based vaccine delivery systems are mainly based on DCs, that are *ex vivo* generated, activated, and loaded with antigens or peptides on their surface[119].

In order to maximize diversity and potential immunogenicity, peptide-based vaccines are designed to contain multiple epitopes in the case of prophylactic vaccines[120]. Therapeutic cancer vaccines usually contain multiple epitopes of different tumor antigens to increase the

probability that one of the components is present on the tumor and prevent escape mechanisms. The peptides can thereby be administered as a mixture or concatenated into a polypeptide [121]. Polypeptide vaccine constructs are referred to as string-of-beads vaccines which can contain spacer sequences between each pair of epitopes to maximize the desired processing and thus recovery of T-cell epitopes.

2.4.2 Cancer Vaccines

In the last decades, the development and application of cancer immunotherapies have been mainly driven by the characterization of genes encoding tumor-associated antigens [122–124]. Immunotherapy options include cancer vaccines which can be based on DCs, recombinant viruses, RNA/DNA, whole-tumor cells or lysate, and peptides. The general idea of using *therapeutic cancer vaccines* is based on the discovery of existing $CD8^+$ and $CD4^+$ T cells in patients, able to recognize TAAs [125]. Peptide-based vaccines aim to trigger T cell-associated immune responses [126]. Studies have shown, that the presence of $CD4^+$ or $CD8^+$ cytotoxic T cells in tumors and an IFN-γ gene signature has a strong association with prolonged patient survival [127,128]. Additionally, this type of vaccine has several advantages, including feasible synthesis, chemical stability, the absence of oncogenic potential, easy administration, and low frequency of side effects [129,130]. Peptide-based cancer vaccines are either designed from TAAs or TSAs. TAAs can be grouped into four categories: differentiation antigens, cancer/testis (CT) antigens, overexpressed antigens, and universal tumor antigens [130]. Differentiation antigens are specifically expressed by a type of tissue and the associated tumor. Most of the genes encoding for these antigens are known in the context of melanoma, including Melan-A/MART-1 and gp100/pMel17 [131,132]. CTAs are expressed in human germ cells within the testis and trophoblasts but not in other normal tissues. Due to their expression in different types of human cancers and the absence of HLA class I expression in testis cells, these antigens serve as tumor-specific T cell targets. Characterized genes encoding for this type of antigens include the melanoma antigen-encoding (MAGE) gene family [133,134]. The class of overexpressed antigens corresponds to antigens which show low expression profiles in normal tissues and overexpression in different tumor types. The HER-2 protein is one example which has shown induced T-cell immunity upon peptide vaccination [135]. Many reported TAAs have restricted expression concerning different tumor types. Antigens, such as survivin [136] and telomerase [137], defined as universal tumor antigens show expression across a broad variety of cancers. A variety of peptide-based vaccines derived from antigenic TAA epitopes have been investigated in clinical trials. The results indicate that these vaccines were able to induce antigen-specific T-cell responses but showed limited evidence of clinical effectiveness [138]. Most of these vaccines were based on CT antigens, differentiation antigens, or overexpressed antigens [139]. As suggested by Rosenberg et al., one explanation for the lack of clinical effectiveness might be the elimination of respective T cells due to central

tolerance of the immune system[129]. Early on, the "Tübingen approach" presented the strategy to identify, select, and validate HLA-presented peptides derived from TAAs, or TSAs if available, for the development of multi-peptide cancer vaccines[140]. Today, therapeutic cancer vaccines are more and more based on TSAs, including epitopes derived from viral gene products or neoepitopes. Neoepitopes originate from non-synonymous tumor-specific (somatic) mutations. Therefore, the selection of these antigens as immunotherapeutic targets should be more effective and safe. Such mutations are highly distinct across different cancer entities and individuals. Thus, the design of cancer vaccines based on such peptides has to be personalized. Earlier studies in mice indicated that vaccination against neoepitopes is effective for small tumor burdens[141–143]. Neoepitope encoding mutations have been discovered through WES on matched tumor and non-malignant samples. The selection of validated mutations was guided by RNA expression and the presence of HLA-presented peptides. Further studies gave evidence for the presence of tumor-specific neoantigen-reactive T cells in patients who received TILs as part of an adoptive immunotherapy[144–146]. Initially, studies on exploiting cancer exome data to analyze therapy-induced T-cell reactivity against personal neoantigens were mainly based on animal models[141,147]. However, more recent case reports show increasing evidence for the efficiency of cancer exome-guided analysis and its application for T-cell reactive neoepitopes in humans. Van Rooij et al. showed the presence of T-cell reactivity against two neoantigens for a melanoma patient based on cancer exome data[148]. Two recent studies demonstrated the safe and effective exploitation of individual (cancer) mutations for personalized vaccination in human melanoma patients[149,150]. Both studies were based on a vaccine design pipeline comprising the identification of individual mutations, *in silico* prediction of neoepitopes, as well as the design and synthesis of personalized peptide-encoding vaccines. Sahin et al. vaccinated 13 patients with RNA vaccines encoding neoepitopes and reported that all patients showed T-cell responses against multiple neoepitopes[149]. Additionally, the vaccine-induced tumor infiltration by neoepitope-specific T cells and killing of autologous tumor cells could be shown for two patients[149]. In 2017, Ott et al. published results on six patients who received a vaccination, and showed the induction of *de novo* T-cell clones which were reactive against multiple neoantigens[150].

2.5 Next-Generation Sequencing

The majority of data, including patient-specific derived genomic sequences and variations, used in this thesis has been generated using NGS. In the following subsections, the history of DNA sequencing in the last decades is outlined, and major technologies, including their applications, are described. Further, we specify bioinformatics approaches for the processing of NGS data in various applications.

2.5.1 The History of DNA Sequencing

The beginnings of DNA sequencing go back to the 1970s with the simultaneous developments of Sanger on the determination of DNA sequences by primed synthesis with DNA polymerase [151] and efforts on chemical sequencing methods by Maxam and Gilbert [152]. Sanger's efforts resulted in the also referred to as first-generation of sequencing methods using the dideoxy method that applies the concept of chain-terminating nucleotide analogs [153]. Due to its lower technical complexity and easier scalability in comparison to Maxam-Gilbert sequencing, Sanger sequencing was the preferred method. In 1995, the first eukaryotic genome of *Saccharomyces cerevisiae* [154] and the first multicellular eukaryote *Caenorhabditis elegans* [155] were successfully determined. In 2001, the most significant milestone in sequencing history was achieved with the first completed draft of the human genome sequence [156,157], followed by the first release of the finished-grade human genome three years later [158]. The human genome sequence has been determined using a whole-genome shotgun approach coupled with automated Sanger sequencing. Due to the small amounts of DNA processed per unit time of sequencers at that time, in addition to high costs, it took ten years and three billion dollars to sequence the first human genome. Despite the establishment of automated Sanger sequencing and many technical improvements resulting in modern capillary electrophoresis (CE)-based Sanger sequencers [159], limitations of the technique were obvious and called for the development of new approaches for DNA sequencing. The shift away from automated Sanger sequencing since the last decade has also been catalyzed by the DNA sequencing technology initiative of the National Human Genome Research Institute (NHGRI). The targeted objective was to reduce the costs through new technologies by four orders of magnitude to about $1000 per human genome in ten years [160]. Subsequent developments resulted in a variety of NGS technologies, also referred to as high-throughput sequencing. The rapid evolution of next-generation DNA sequencing technologies and their application led to a significant drop in cost per genome since 2008. In general, NGS requires less DNA than Sanger sequencing and takes less time due to the possible combination of chemical reaction and signal detection as well as parallelization of read generation. In 2017, the cost per genome was on the brink of breaching the $1,000 boundary (July 2017: $1,121) whereas, the cost of determining one megabase (Mb; a million bases) dropped to $0.012 [161] (Figure 2.6).

2.5.2 Sequencing Technologies

Different sequencing technologies are defined by the unique combination of methods which can be grouped as template preparation, sequencing, imaging, and data analysis [162]. Template preparation includes either the clonal amplification of single DNA molecules or single DNA molecule templates. Usually, clonally amplified templates are created by emulsion PCR (emPCR) [163] or solid phase amplification [164]. Subsequently, fragment templates or mate pair

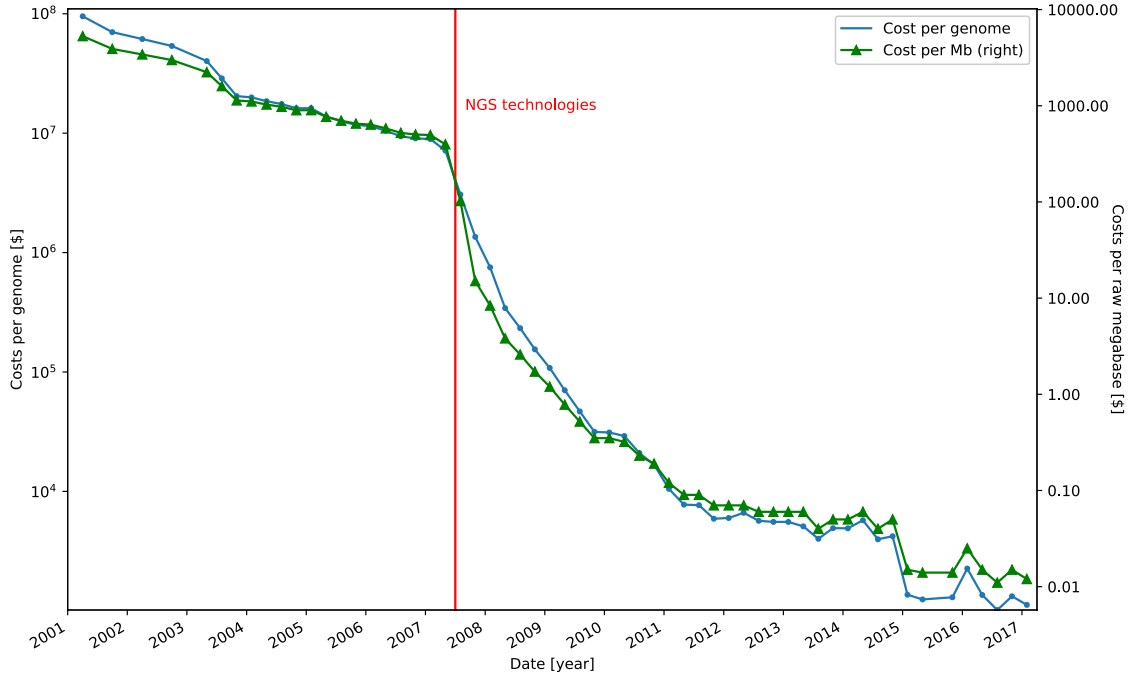

Figure 2.6: Development of the cost per genome and raw megabase of DNA sequence from September 2001 to July 2017. The cost per genome (blue) and cost per raw Mb (green) decreased significantly, especially after the introduction of NGS technologies in 2008 (red line). Data taken from Wetterstrand [161].

templates, created from genomic DNA by randomly breaking it into smaller sizes, are immobilized or attached to a solid surface or support. Forms of sequencing approaches involve cycle reversible termination (CRT), single-nucleotide addition (SNA), real-time sequencing, and sequencing by ligation (SBL) where DNA polymerase is replaced by DNA ligase [162]. In the process of read generation, individual bases in each fragment are identified through imaging methods based on the measurement of bioluminescent signals or four-color imaging of single-molecular events [162]. Data analysis methods vary depending on the used sequencing platform and applications. Established (commercial) NGS platforms include Illumina (Solexa) sequencing by synthesis (SBS) [165], Roche 454 pyrophosphate sequencing (discontinued in 2013) [166], AB Sequencing by Oligo Ligation Detection (SOLiD) [167], and Ion Torrent Personal Genome Machine (PGM) sequencing [168]. More recent methods, also considered as third-generation sequencing, include Oxford Nanopore Technologies sequencing [169] and Pacific Biosciences single-molecule real-time (SMRT) sequencing [170]. These new real-time sequencing technologies enable single-molecule sequencing. Moreover, even higher throughput, faster turnaround time and especially longer reads can be accomplished in comparison to second-generation sequencing technologies. Technical details and a comparison of established NGS platforms are covered by several reviews [162,171] and will not be discussed here as it is beyond the scope of this thesis.

Since the majority of data analyzed in this thesis has been generated using Illumina HiSeq systems, this technology will be discussed in more detail. Illumina HiSeq systems employ sequencing by synthesis chemistry. During sequential cycles of DNA synthesis, fluorescently labeled deoxyribonucleotide triphosphates (dNTPs) are incorporated into a DNA template strand. This process is catalyzed by DNA polymerase. Upon incorporation, due to the fluorophore excitation nucleotides can be identified. In the library preparation step, DNA or complementary DNA (cDNA) samples are randomly fragmented and ligated with 5′ and 3′ adapters. The resulting adapter-ligated fragments are then amplified by PCR and gel purified. Afterwards, clusters are generated by loading the library into a flow cell. The fragments with complementary library adapters are hybridized to the flow cell surface with surface-bound oligos and amplified into distinct, clonal clusters using bridge amplification. Subsequently, sequencing by the Illumina SBS technology involves a proprietary reversible terminator-based method[165]. During each sequencing cycle, all four reversible terminator-bound dNTPs are present. Single bases are detected as they are included into DNA template strands. Bases are identified by emission wavelength and intensity captured by flow cell imaging and emission recording from each cluster. The number of sequencing cycles determines the length of reads. Produced read lengths of the HiSeq 2500 system range from 36 to 250 bp depending on the run mode. Higher throughput compared to CE-based sequencing instruments is achieved due to the massively parallel sequencing of millions of fragments instead of single DNA fragments. The HiSeq 2500 platform has a throughput of ten gigabases (Gb) per day to one terabase (Tb) per run, rendering the processing of eight human genomes at 30x coverage or 150 human exomes per run possible. Multisample sequencing studies can be performed in a short amount of time through the simultaneous pooling and sequencing of libraries (multiplexing) in a single run. Further improvements were achieved by the development of paired-end sequencing, the sequencing of both ends of DNA fragments in a library and the alignment of forward and reverse reads as read pairs. This enables the production of twice the amount of reads at the same time, as well as more accurate read alignment.

2.5.3 Applications

The expanding availability of NGS technologies in labs, the increasing throughput and accuracy, as well as the reduction of costs facilitate the ever-expanding set of NGS applications, especially in the area of biomedical applications. The study of genomes is not limited by their size anymore, nor bound to the characterization of single genes associated with genetic disorders including cancer and rare diseases[171]. Instead, high-throughput sequencing of organisms makes large amounts of genetic data available, enabling a variety of studies in genomics, metagenomics, epigenomics, and transcriptomics. Available standard protocols involve whole-genome sequencing (WGS), RNA sequencing (RNA-Seq), targeted sequencing, including exome and

16S sequencing, as well as chromatin immunoprecipitation sequencing (ChIP-seq), methylation sequencing (Methyl-seq). Initially, NGS technologies have been successfully applied for *de novo* sequencing of genomes for different species. Further developments led to the application of these methods for resequencing of human genomes and exomes. Thereby, the identification of *single nucleotide variants* (SNVs) could be accomplished by comparison of the sequenced reads and available reference genomes. Furthermore, sequencing can be applied for the global characterization of structural variation (SV), characterized as large (>1 kb) segments which have been duplicated, deleted, or rearranged. However, due to short read lengths of NGS platforms, this task is more challenging in comparison to the determination of SNVs [172] but might get more feasible using more recent technologies including SMRT [173]. International research projects, such as 1000 Genomes Project [174,175], try to catalog the human genetic variation for many individuals in diverse populations. Through the application of RNA-Seq, in-depth characterization of the transcriptome of various cells, tissues, and organisms became practicable [176]. Specific applications include the quantitation of transcript abundance, the characterization of different present gene isoforms, the identification of actively translated messenger RNA (mRNA) transcripts ratios, RNA-editing, as well as cellular roles of RNA and allele-specific expression estimation. Applications are not solely based on the analysis of single organisms but are also applied for species classification and gene discovery of metagenomic samples. Metagenomic studies of microbiomes include diverse sources as the ocean, soil, and human body. Shotgun sequencing approaches are usually used for the detailed characterization of species and gene composition, whereas 16S ribosomal RNA (rRNA) gene sequencing is applied for the characterization of phylogenetic relationships. Recently, efforts have been made to characterize the human microbiome and its role in human health and disease [177]. Concerning medical applications, WGS, WES, and WTS help to acquire a deeper understanding of the genetics of diseases, especially for rare Mendelian disorders and cancer. Large consortia for cancer genome sequencing such as The Cancer Genome Atlas (TCGA) as part of the National Cancer Institute's (NCI) Genomic Data Commons (GDC) [178] and the International Cancer Genome Consortium (ICGC) [179] characterize and collect tumor and matched normal samples. The comparison of the cancer genome to the matched unaffected genome enables the comprehensive characterization of somatic genome alterations and therefore the detection of somatic variants.

2.5.4 Bioinformatics for Next-Generation Sequencing

The choice of bioinformatics tools for the analysis of NGS generated data highly depends on the application. As the focus of this thesis is the analysis of personal human genomes and transcriptomes, as well as (somatic) variant detection, involved processes are explained in detail. The analysis pipeline for genetic variant analysis typically includes: quality-control, preprocessing, read alignment, post-alignment processing, and variant calling and annotation [180].

Large-scale transcriptome analysis usually involves different read alignment strategies, as well as the quantification of expression in RNA-Seq data.

Quality Control of Raw NGS Data

The initial step in most of the NGS-based applications is quality control (QC) of raw reads to assess quality issues including low base quality, contamination with adapter sequences, and biases in the base composition. Usually, depending on the used NGS technology, raw reads are given in FASTQ format[181]. The file format is an extended version of the FASTA file format, which includes, in addition to the sequence identifiers and text-based representation of sequences, a Phred-scaled base quality score. The Phred quality score (Q score) has been originally developed for Phred base-calling, an algorithm for the identification of bases from fluorescence signals[182,183]. The quality score, given in ASCII space, is related to the probability P of a base to be incorrectly called by the sequencer and is defined as follows: $Q = -10\log_{10} P$. Tools, such as FastQC[184] can be used for the assessment of raw read quality to determine the necessity of preprocessing steps such as base trimming, read filtering, and adapter clipping. Quality measurements include the distribution of (PHRED) quality scores across bases of reads, the guanine-cytosine (GC)-content distribution, read length distribution, and sequence duplication level.

Preprocessing

Preprocessing of NGS data usually involves trimming of low-quality bases, adapter removal in reads, and the removal of redundant reads or undesired sequences involving contamination from primers or other species[180]. Existing solutions employ different approaches such as semi-global sequence alignments (Cutadapt[185]), hash-based search followed by a simple score-based search (Trimmomatic[186]), bit-masked k-difference matching using a dynamic programming algorithm (Skewer[186]), and probabilistic approaches to detect the overlap between forward and reverse reads (SeqPurge[187]).

Read Alignment

An essential step in the resequencing of DNA is to gain insights in the genetic difference to an available reference genome. Therefore, preprocessed reads are usually mapped against a reference genome assembly. In the case of unknown genomic sequences, genome assemblies can be determined de novo using assembly algorithms. Concerning resequencing of the human genome, the two primary sources of reference genomes are the University of Santa Cruz (UCSC) and the Genome Reference Consortium (GRC). Both sources released (sub-)versions of the human genome namely the UCSC versions *hg18*, *hg19*, and *hg38*, as well as the GRC versions *GRCh37* and *GRCh38*. The goal of read alignment is to determine the location within

a reference genome sequence that matches the observed DNA sequencing read and therefore, ultimately identify the original site of the read. A certain amount of mismatches has to be tolerated since sequencing errors in the read sequence as well as actual variations between the reference genome and the sequenced genome exist.

Implementations for read sequence alignment usually carry out a multistep procedure. In the first step, heuristics are used to identify a smaller set of locations within the reference where the read could be placed. Afterwards, more accurate (semi-) global or local alignment algorithms, including derivatives of the Needleman-Wunsch algorithm[188] or gapped and ungapped versions of the Smith-Waterman algorithm[189], are used to identify the exact location[190]. Fundamental techniques for the first step of this multistep process include hash-based alignment methods and Burrows-Wheeler transform (BWT)-based methods[191]. Hash-based alignment methods create a hash-table data structure for indexing and scanning the sequence data. Depending on the implementation, hash tables are either build on the set of input reads[192,193] or the reference genome[194,195]. Read mapping implementations based on BWT, including BWA[196,197], Bowtie[198], and Bowtie 2[199], create an index of the reference genome to facilitate rapid search and low-memory consumption. Due to the reordering of the reference genome sequence using BWT, multiple occurring sequences are placed together and therefore allow efficient index creation. The final index creation is based on suffix arrays, such as the FM-index[200], that are created from the derived BWT sequence to facilitate efficient subsequence search. Additionally, existing read alignment implementations handle mismatches, gaps, and paired reads using scoring schemes, which are based on base quality values or the edit distance. Current read alignment solutions can be broadly classified in tools striving to find the best mapping location of a read according to an assigned score (best-mappers[196,199,201]) or tools enumerating a comprehensive set of locations (all-mappers[202,203]). Other approaches, such as Masai[204], combine the enumeration of all read locations with the option to perform best mapping.

In the case of read alignment of RNA-Seq data, two basic strategies exist, given that a reference genome or transcriptome is available. Reads are mapped to the reference genome using gapped mappers or to the reference transcriptome using ungapped mappers. Read alignment implementations have to address the problem of aligning spliced reads across introns, as well as the determination of exon-intron boundaries. The discovery of exon junctions is either guided by initial read alignments or existing gene annotations[205]. The final alignment is based on identified junctions. Available implementations of both strategies include TopHat[206,207], and STAR[208]. Internally, TopHat uses Bowtie or Bowtie 2 to map the reads to the reference genome. The identification of possible exon-exon splice junctions, without reference annotation, is based on the initial mapping step. The generated database of possible splice junctions is then used to map reads against junction candidates to confirm their location. STAR uses a multistep alignment procedure to align non-contiguous sequences directly to the reference

genome. In the seed finding phase, a suffix-array algorithm is used for the sequential search for a Maximal Mappable Prefix (MMP), followed by cluster generation of aligned seeds around a set of anchor seeds. Full alignments are then generated by joining the seeds within defined genomic windows.

Most of the recent read alignment implementations generate alignments in Sequence Alignment Map (SAM) file format or its binary representation (BAM)[209], which include fields for reference sequence name, mapping position, mapping quality, and base quality.

Variant Calling

Variant calling is the process of identifying genomic variations of a sample by the comparison of aligned reads to the reference genome. Variants include SNVs, (short) insertions/deletions (InDels), CNVs, and large SVs. These variants might be inherited and present in germ cells (germline variants) or only present in somatic cells (somatic variants). The process of distinguishing somatic mutations from germline mutations, based on somatic and matched germline samples, is defined as somatic variant calling. In cancer research, one important application of somatic variant calling is to identify somatic mutations which are present in tumor cells and are distinct from germline polymorphisms. Germline variant calling methods employ different approaches for variant detection. Existing algorithms include the Genome Analysis Toolkit (GATK) HaplotypeCaller[210,211], FreeBayes[212], and SAMtools[209]. All of these methods rely on Bayesian approaches. HaplotypeCaller uses a Bayesian model to estimate the likelihood of the genotype based on observed sequence reads covering a specified locus. SNV, InDel, and SV calling accuracy is further improved by local assembly of aligned reads. SAMtools uses a two-step process for variant calling. First, `mpileup` is used to compute possible genotypes and their likelihood from aligned reads along the genome. Then, bcftools calls SNVs and InDels based on the estimated genotype likelihoods. FreeBayes implements a generalized version of the Bayesian statistical method by Marth et al.[213] to allow multiallelic loci and non-uniform copy number across samples[212]. SNVs, InDels, multi-base mismatches, and CNVs can be detected simultaneously.

Somatic variant calling methods can be broadly grouped into two categories. GATK and SAMtools can be used to detect somatic variants, where variant calling is performed subsequently on the tumor and the normal sample. Afterwards, genotype-based subtraction methods are used to distinguish between variants present in all samples (germline) and variants which are only present in the tumor sample (somatic). Other methods perform variant calling simultaneously on both samples using different Bayesian approaches (Strelka[20], SomaticSniper[21,214], MuTect[215]) or Fisher's exact statistics (VarScan[216] 2). The detection of variants is either based on joint diploid genotype likelihoods or shared allele frequency between the samples. Reviews

with a comprehensive assessment of variant detection accuracy and comparisons of available (somatic) variant calling methods are available elsewhere[217,218].

Variant Annotation

Detected variants are usually stored in Variant Call Format (VCF)[219] files, which can be annotated with the affected gene, transcripts, transcript-relative coordinates, and amino acid changes. Further annotations include the type of the variant with respect to the location (exonic, intronic, or intergenic) and its functional role (synonymous, non-synonymous, or frameshift)[220]. These annotations are necessary to understand the functional effects and to perform prioritization analysis of detected variants. Common variant annotation methods include ANNOVAR[220], SnpEff[221], and Ensemble VEP[222]. Additionally, genomic variation data can be linked to existing databases. Thus, variants can be annotated with additional information including minor allele frequency (MAF) using dbSNP[223], evidence in pathogenesis from disease variant databases (ClinVar[224]), and cancer association (COSMIC[225]).

Gene and Transcript Quantification

The most common goal of RNA-Seq analysis is to retrieve estimates of gene and transcript expression. Therefore, RNA-Seq reads can be mapped to a genome or transcriptome reference. Estimates of gene and transcript expression are then calculated based on the number of aggregated raw reads mapping to each transcript[226]. Methods such as HTSeq[227] are used to quantify expression on gene-level by counting the overlap of reads with genes. Gene transfer format (GTF) files provide essential genomic coordinate information of exons and genes. Other approaches as implemented in Kallisto[228] do not rely on mapped RNA-Seq reads, but rather give expression estimates based on a reference transcriptome and raw reads from RNA-Seq experiments. Kallisto constructs pseudoalignments of reads using hashing of read-derived k-mers and a de Bruijn graph representation of the transcriptome, identifying the transcripts from which the reads could have originated[228]. Most common applications need comparisons of expression levels among samples. Hence, different measures are used to normalize counts and account for factors such as feature-length and library-size. One of the proposed measures, *reads per kilobase per million mapped reads* (RPKM)[229] is derived as follows:

$$RPKM = \frac{10^9 \cdot C}{N \cdot L},$$

where C is the number of mapped reads that fell onto the feature, N is the total number of mapped reads in a given sample, and L is the length of the feature in bp. Other measures include *fragments per kilobase per million mapped reads* (FPKM), which is used in paired-end RNA-Seq experiments since two reads can correspond to a single fragment and therefore should not be

counted twice. The measure *transcripts per million* does not account for the total number of mapped reads, but takes the read length into account. It has been proposed as an alternative to RPKM[230]. Still, for the comparison across samples, especially regarding differential expression analysis, some methods work on mapped read counts rather than normalized data. Most common are comparisons across different biological samples to estimate effects on RNA expression levels in the presence of drug treatment or to compare cells in a healthy and diseased state. Differential expression analysis usually involves three steps: normalization, statistical modeling of gene expression and testing for differential expression. To perform multi-sample comparisons, DESeq2[231], and edgeR[232] use normalization approaches (median-of-ratios, trimmed mean of M values (TMM)[233]) based on negative binomial distributions, to allow a higher variance, especially for biological replicates, than the Poisson distribution.

2.6 Towards High-Throughput Computational Immunomics

Developments concerning sequencing technologies lead to ever-increasing numbers of sequenced genomes accompanied by large amounts of data. Combined with clinical and epidemiologic data, NGS data is of high relevance in immunology research but at the same time presents new challenges, including data handling and data processing. Immunomics is defined as the interdisciplinary field of immunology, immunoinformatics, genomics, proteomics, and bioinformatics. Challenges, as mentioned above, gave rise to the field of *computational immunomics* including computational methods and resources to make sense of immunological data, mechanisms of immune function and disease pathogenesis. Moreover, there is an increasing demand for easily accessible, unified access points of integrated data resources with means of standardized data analysis on powerful computing resources. This chapter introduces the central concepts of computational immunomics, web-based portals, and workflow systems.

2.6.1 Computational Immunomics

A key challenge in computational immunomics is the rational design of EVs for the prevention of infectious disease and the treatment of cancer using (personalized) immunotherapies. Since one requirement of the induction of a T cell-mediated immune response is MHC binding, computational approaches aim to predict the outcome of the MHC processing pathway which includes proteasomal cleavage, TAP transport, and MHC binding. Methods for the prediction of proteasomal cleavage sites utilize a variety of machine learning algorithms which can be either trained on *in vitro* or *in vivo* data. NetChop[234,235] uses an algorithm based on neural networks which have been trained on *in vitro* degradation data (NetChop 20S) and MHC ligand data (NetChop C-term) to account for different specificities of the constitutive proteasome and the immunoproteasome. Proteasomal cleavage matrices (PCMs), derived from observed cleavage

probabilities, can be used to predict cleavage using probability-based models as suggested by Donnes et al.[38] TAP transport prediction methods try to estimate TAP affinity and therefore assess the likelihood of peptides being transported to the ER. The method SVMTAP[38] uses support vector regression (SVR) and sparse binary encoding for peptides. The algorithm was trained on peptides with experimentally derived IC50 values as used in other methods[236]. Peters et al.[237] suggested a stabilized matrix-based approach.

HLA ligand binding is necessary, although not sufficient, for the induction of T cell responses. Due to its importance, various approaches have been suggested in the last two decades. Methods for MHC class I and II binding prediction are available. However, MHC class II binding algorithms (SYFPEITHI[238], NetMHCII[239,240] NetMHCIIpan[241]) are less performant due to the different binding mode of pMHCII complexes and unknown positions of the binding core. In MHC class I prediction, allele-specific predictors give estimates for MHC binding affinity for a subset of MHC alleles trained on experimental data. Available approaches use position-specific scoring matrices (PSSMs)[238,242], and modern non-linear machine learning methods like *support vector machines* (SVMs)[243], and artificial neural networks (ANNs)[244,245]. Pan-specific methods such as NetMHCpan[246,247] use training data from quantitative binding essays of related alleles to provide predictions for alleles with insufficient data points. NetMHCpan 4.0, the most recent pan-specific method, has been trained on binding affinity and eluted ligand data[248]. The integration of presented peptides identified by mass spectrometry (MS) in the training data led to increased predictive performance[248]. NetMHCcons[249] is a consensus method for MHC class I prediction which integrates NetMHC, NetMHCpan, and the matrix-based method PickPocket[250] to increase prediction accuracy. Since the availability of immunomic data and the demand for training data sets is increasing, immunomic databases have been established. The Immune Epitope Database (IEDB)[251] provides curated epitope data, as well as analysis tools and prediction algorithms for T-cell and B-cell epitope prediction. Another source for naturally processed MHC ligands and T-cell epitopes is the SYFPEITHI database[238]. Although various prediction methods along the MHC processing pathway exist, the prediction of T-cell reactivity remains challenging. Methods as NetCTLpan try to combine predictions of the different steps of the MHC processing pathway to estimate T-cell reactivity[39]. POPISK[42] uses an SVM with a weighted degree string kernel. Toussaint et al. incorporated immunological tolerance estimates to improve predictions[43]. Even though the presentation on MHC does not guarantee recognition by TCRs and T-cell reactivity, MHC binding predictions are often used as T-cell reactivity estimates due to the lack of high-accuracy prediction methods.

Assuming that T-cell reactivity estimates are given, algorithms for the selection and assembly of epitopes exist. The problem of selecting the best set of epitopes as vaccine components can be solved using heuristics[252] or global optimization[253,254]. To optimally assemble selected epitopes and therefore improve vaccine efficiency by an increased epitope recovery rate, two methods formulated and solved the problem as a derivative of the traveling salesman problem

(TSP) without spacers[255] and optionally with spacers[256]. Since the importance of accessibility and efficient use for a broad user group are increasing, web-based solutions for epitope selection[257] and vaccine design[258,259] have been suggested. The need for new complex data analysis pipelines has been addressed by software packages such as ImmunoNodes[260], providing the graphical development of computational immunology workflows or the Python frameworks FRED[25] and FRED2[26].

2.6.2 Workflow Systems and Web-based Portals

With the growing amount of data, its availability, and the relevant issue of reproducibility[261,262], there is an increasing need for the execution of structured chained analysis steps, especially for large-scale data analysis in bioinformatics. Workflow engines help to define pipelines consisting of a combination of parametrized steps and to automate the execution of these pipelines. *Workflow management systems* (WMSs) usually provide (graphical) means of workflow creation, execution, and monitoring. Galaxy[46,263,264] is a free software system which comes with a workflow engine. It is distributed as a public web service (`http://usegalaxy.org`) and a software package for local instances that can be deployed on Unix systems. The public web service provides a broad range of pre-installed tools which can be executed on the connected infrastructure through a graphical user interface (GUI). The Galaxy software offers integration of computing environments such as clusters and clouds[265]. Through the Galaxy Tool Shed[266], tool configurations and 'recipes' can be obtained and installed from a central location. A similar feature is provided by Taverna[267,268], which enables sharing of workflows via myExperiment[269]. Taverna is a domain-independent WMS that provides an open-source tool suite for the design and execution of scientific workflows. The tool suite includes a graphical workbench to create and execute workflows, as well as a server application for remote execution of workflows. To increase portability and scalability across different workflow engines and hardware environments, the Common Workflow Language (CWL)[270] specification has been suggested. The CWL standard uses task-based workflow definitions with explicit input and output statements. Workflow engines like Galaxy and Taverna will provide implementations of the CWL specification in the future. The Konstanz Information Miner (KNIME) analytics platform is an open-source platform for the development and execution of workflows[271]. It provides a user-friendly GUI that enables users to build workflows from existing KNIME nodes or custom user-build nodes. Execution on distributed *high-performance computing* (HPC) resources or cloud-based execution is possible via the fee-based suites KNIME Server and KNIME Cloud Server. The *grid and cloud User Support Environment* (gUSE) is an open-source WMS that comes with a set of high-level grid services[272]. The customizable service stack enables users to create, execute, and monitor workflows. Through the generic Distributed Computing Infrastructure (DCI) framework, multi-DCI and -cloud workflow execution is possible on services such as

clusters, grids, supercomputers, desktop grids, and clouds. All services can be accessed via the Web Services Parallel Grid Runtime and Developer Environment (WS-PGRADE) component, which serves as a web-based GUI. Besides, given services can be accessed without the graphical WS-PGRADE component via the remote API. Snakemake[273] is a Python-based workflow engine inspired by the build system GNU Make[274]. Workflows are inferred from a set of rules which are used to create a directed acyclic graph (DAG). The graph represents the sequence of rule executions where rules specify input files, output files, and a shell command or Python code. The Snakemake execution environment allows the execution on single-core machines, as well as on compute clusters[273].

To further ensure reproducibility, bioinformatic tools can be containerized. Containers are isolated systems that provide virtualization of an execution environment and share the same operating system (OS) kernel as the host. Therefore, containers are more lightweight and efficient than virtual machines (VM) which need a guest OS and thus create more overhead. Container engines like Docker[275] and Singularity[276] allow users to define recipes to build the container image. The workflow engine Nextflow[277] combines the two concepts of workflows and containerization. It is a reactive workflow framework and a domain-specific language that defines and executes pipelines made of different processes, which can be written in any scripting language. Due to the container integration, processes can be executed in a Docker or Singularity container.

As there is a growing demand for easily accessible GUI-based solutions for reproducible research, highly tailored web-based portals are used to bring resources to scientific communities. Such resources can be a combination of information from diverse sources with unified access, presented in a uniform way as in the BioMart Central Portal[278]. Other portals, such as the ICGC Data Portal[279], GDC Data Portal[178], and cBioPortal[280], additionally provide essential resources for data analysis and visualization. Access to bioinformatics pipelines and computing resources like HPC systems is provided by portal solutions like the public Galaxy Server[46] or GenePattern[47].

Existing web-based portal solutions are based on different technologies such as web application servers, containers, and frameworks. In a Java-based setting, web application servers like Apache Tomcat[281], GlassFish[282], and WildFly (former JBoss)[283] implement different aspects of the Java EE specification including the Servlet specification, the JavaServer Pages specification, and protocol specifications like HTTP. Liferay Portal[284] and GateIn Portal[285] (former JBoss Portal) are open-source Java-based portal solutions which can be deployed on application servers as mentioned above. In general, portals act as a software platform for building websites and web pages which can consist of pluggable software components, called portlets.

Chapter 3

HLA Genotyping from Next-Generation Sequencing Data

The content of this chapter is to the most extent part of the manuscript:

OptiType: precision HLA typing from next-generation sequencing data

Szolek, A.*, Schubert, B.*, Mohr, C.*, Sturm, M., Feldhahn, M., and Kohlbacher, O.

Bioinformatics, 30(23), 3310-3316 (2014)

* Joint first authors

3.1 Introduction

As described in the background on MHC (Section 2.2.1), the HLA cluster is one of the most polymorphic regions in humans which plays a crucial role in adaptive immunity. Therefore, HLA molecules are relevant to many biomedical applications and medical areas, such as vaccinology[286,287], regenerative and transplantation medicine[288,289], and autoimmune diseases[290,291].

In particular, with the recent advances in precision medicine, there is an increasing need for fast and accurate techniques for HLA genotyping. This is even more important for personalized medicine approaches in (cancer) immunotherapy, where the activation of the patient's immune system is used to fight the disease. Consequently, the individual HLA genotype has implications for immune recognition and therefore the effectiveness of treatments due to the HLA-mediated restricted recognition by T cells. Personalized cancer vaccines are already designed from the particular genetics and biology of the patient. Therefore, this development is dependent on

the availability of an individual's HLA alleles. HLA genotype information can be available with different degrees of resolution, whereas one can distinguish between HLA allele families (two-digit) or distinct HLA protein sequences (four-digit). However, the determination of present HLA alleles is challenging for several reasons, such as the vast allelic diversity.

To this day, 12,800 different HLA-I and 4,800 HLA-II alleles have been identified (IPD-IMGT/HLA[292] Release 3.31, April 2018). As shown in Figure 3.1, the number of HLA alleles, and the availability of genomic sequences is steadily growing.

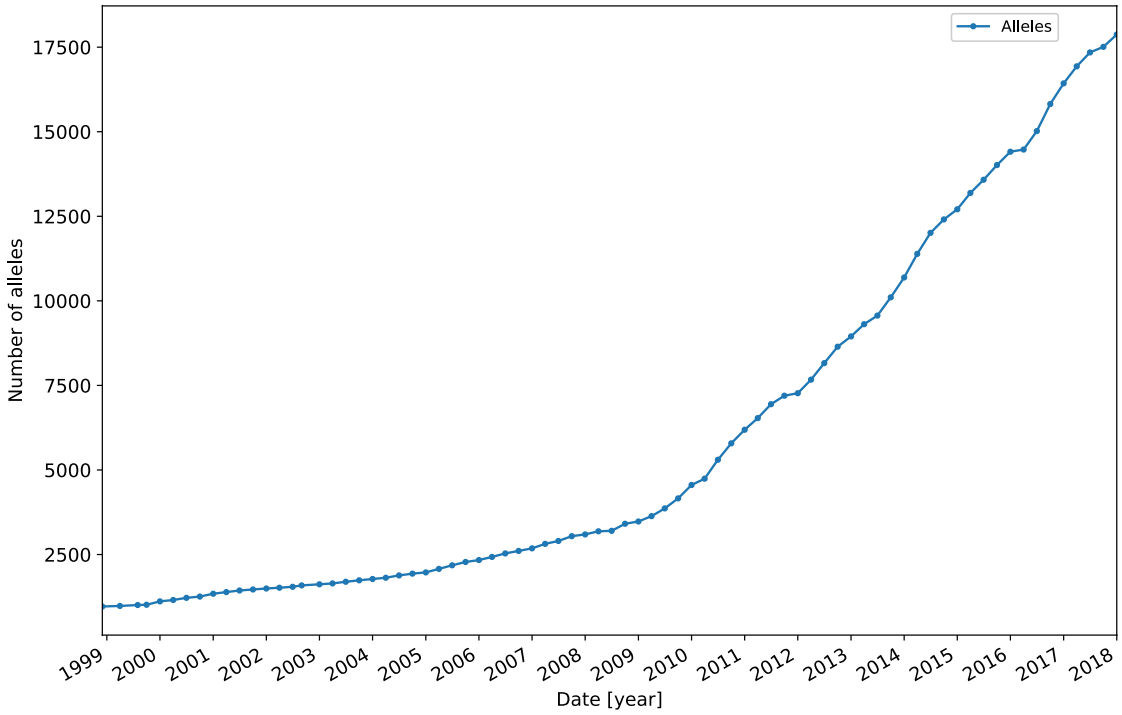

Figure 3.1: Database growth of the IPD-IMGT/HLA database. The number of HLA class I and II alleles has been steadily increasing since the first release in December 1998. Today, in January 2018, the database contains 17,874 HLA sequences (data derived from `http://www.ebi.ac.uk/ipd/imgt/hla/stats.html`).

Additionally, HLA alleles share significant sequence similarity which renders the unique identification of a genotype based on short-read sequencing techniques highly complex. High degrees of sequence similarity is even present across different loci, which often leads to ambiguous genotyping results of HLA typing approaches[293]. Established approaches are based on sequence-specific oligonucleotide probe hybridization, PCR amplification with sequence-specific primers, or serotyping techniques. Such probing techniques are usually labor-intensive and time-consuming, which promoted the development of HLA enrichment and sequencing techniques for the sole purpose of HLA typing. The use of targeted NGS has been demonstrated by Gabriel et al.[294] and Bentley et al.[295]. Other new established protocols[30,296–298], which

still require elaborate preparations, are based on NGS technologies as well. In 2013, Danzer et al. proposed a protocol using GS 454 Junior sequencing[299]. Even though their approach enables high-resolution typing within two days, computational approaches were needed to reduce time and money consumption further. One hybrid approach, where the computation of the posterior probability of HLA allele pairs has been combined with a 454 GS FLX Titanium sequencing pipeline, was established by Erlich et al.[300]. Still, this approach relies on specific sequencing and PCR amplification techniques.

To overcome these limitations and avoid additional costs and turnaround time, routinely generated data could be used. Cheaper personal genome sequencing and routine sequencing of patient exomes or whole genomes, established in many larger clinical centers, open up new possibilities for HLA typing. Still, there is an unmet need for fast and precise computational HLA typing approaches from short-read sequencing data. One reason for that is the above-mentioned high variability of the HLA loci which renders established read mapping and variant calling-based analysis of NGS data not suitable for HLA genotyping.

Related Work

Several methods based on various techniques have been proposed in the last couple of years. In 2012, Warren et al. proposed an algorithm (HLAminer[32]) based on allele-specific scoring for whole genome, exome, and transcriptome sequencing. Based on properties of contigs aligned against an HLA reference database, derived through *de novo* assembly, scores for HLA alleles are calculated. For each locus, the highest-scoring alleles are reported. Other approaches for RNA-Seq data include seq2HLA[33], a greedy algorithm based on read count maximization, and HLAforest[35], a tree-based top-down greedy algorithm. The tree is constructed for each read based on the mapping results against the HLA references.

Liu et al. published an approach (ATHLATES[293]) which determines the most probable HLA allele pairs for each locus based on the minimal Hamming distance to the variable positions of each exon. For each exon of the HLA reference sequences, the best mapping contig is determined. The Hamming distance for each allele is then calculated to all aligned exons, followed by the application of different filtering criteria.

In 2013, Major et al. proposed an approach which selects allele pairs based on the optimal coverage depth and sequence coverage of their alignment[36]. Subsequently, applied filtering steps enforce certain sequence coverage of exons 2 and 3, number of mismatches, and alignment orientation of paired reads.

The aforementioned methods do not provide sufficiently accurate predictions, especially concerning clinical applications. Kim et al. and Warren et al. reported four-digit HLA genotyping accuracies of 85–90% on RNA-Seq data[32,35]. For WGS and short-read RNA-Seq data, the reported accuracy was even lower. Other limitations include the reported HLA genotype

resolution and incomplete typings for specific samples. The proposed approach by Boegel et al. (seq2HLA) is only capable of reporting two-digit HLA genotypes[33]. Major et al. reported an accuracy of 94% on exome sequencing samples[36]. However, due to the applied stringent filtering criteria, their method only yielded full typings for 161 out of 217 samples.

One reason for limitations concerning typing accuracies might be the independent consideration of each HLA locus. As mentioned above, HLA alleles share high sequence similarity across loci. Reads might, therefore, map to alleles of different loci with equally good alignment scores. Common to the above approaches is also the disregard of intron sequences in exome and WGS data. This results from the unavailability of complete sequence data. 94.6% of HLA sequences contained in the IPD-IMGT/HLA database[292] (Release 3.14.0, July 2013) lack parts of their exonic or intronic sequences.

Project Outline

We implemented a new HLA genotyping algorithm (*OptiType*) based on integer linear programming. The above-described issues are tackled by the simultaneous consideration of all major and minor HLA-I loci and the inclusion of intronic information. Our approach is based on the assumption that the actual present genotype explains more reads than any other genotype. An allele explains a read if the corresponding read is aligned to it with fewer mismatches than to any other allele. By maximizing the number of explained reads, we find the optimal combination of HLA alleles and therefore the most probable present HLA genotype. OptiType is capable of producing accurate predictions from NGS data that has not been specifically enriched for the HLA cluster. The pipeline comprises three key steps (Figure 3.2).

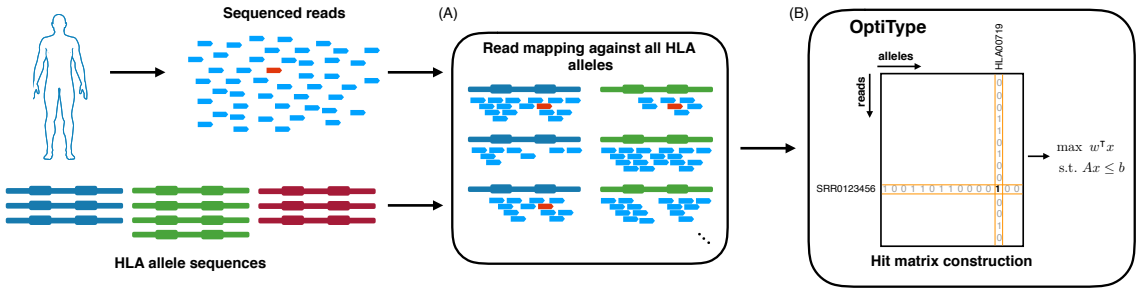

Figure 3.2: Key steps of the HLA typing pipeline of OptiType. (A) First, NGS reads are mapped against a constructed HLA allele reference. The libraries (genomic/ coding DNA sequence (CDS)) are constructed from exon 2 and exon 3 of known HLA-I alleles and flanking intronic regions in the case of genomic sequences. (B) From the mapping results, a binary hit matrix $C^{R \times L}$ is constructed for all reads $r \in R$ mapping to at least one allele $a \in L$. A successful mapping of read r to allele a is denoted by $C_{r,a} = 1$. For each HLA-I locus, up to two alleles are selected by the formulated ILP based on the hit matrix. The solution is optimized for the number of explained reads. Figure adapted from Szolek et al.[301]. Human body silhouette icon obtained from Reactome Icon Library[88] and adapted.

For the initial alignment step, only exon 2 and 3 are taken into account due to the availability of these subsequences for all alleles. Consideration of other sequences would thus favor alleles with complete sequence information. We constructed an HLA allele reference set which further includes flanking intronic regions for WES and WGS data. Since the intronic information is not available for most of the alleles, we impute the missing information from other alleles based on the phylogenetic information. Based on the read alignment results, we generate a binary hit matrix $C^{R \times L}$ for all reads $r \in R$ mapping to at least one allele $a \in L$. To find the optimal set of HLA alleles, we formulated a particular case of the set cover problem[302] as an integer linear program (ILP). The ILP solution is optimized for the number of mapped explained reads, whereas up to two alleles are selected simultaneously for each locus. Additionally, minor HLA loci G, H, and J are considered during optimization to account for ambiguous read alignments, as long subsequences of these minor loci present a high similarity to major loci.

3.2 Materials and Methods

This section describes implementation details of our algorithm for HLA typing based on NGS data, including the formulated ILP. Further, we give detailed information on employed software, data sets, and carried out simulation studies. The pipeline was implemented in Python 2.7 using the module Pandas 0.12 (`http://pandas.pydata.org`) with HDF5 1.8.11 (`https://www.hdfgroup.org/HDF5`) data persistence support. The source code is available at `https://github.com/FRED-2/OptiType` under a three-clause BSD license. Performed comparisons of results to experimental typings are based on the percentage of correctly predicted alleles. In the case of the evaluation of zygosity predictions, the correctness of the predicted zygosity in comparison to experimental results without considering the typed alleles is reported. The statistical analysis was conducted using R 3.0.2. The 95% confidence intervals were calculated by bootstrapping with 100,000 repetitions.

3.2.1 Reference Construction

The initial read mapping step of the OptiType pipeline is based on a constructed reference library for coding DNA sequences (CDS) and genomic nucleotide sequences. Corresponding information for all HLA-I alleles was obtained from the IPD-IMGT/HLA database[303] (Release 3.14.0, July 2013). Concatenated exon 2 and 3 coding sequences build the reference library for RNA-Seq data. For the DNA sequence reference library, we further include the intron sequences flanking exon 2 and 3. OptiType uses reconstructed intron sequences for alleles where intron sequences are not available. In case of missing sequence data, the data is completed by information from the closest phylogenetic neighbor from the set of complete HLA allele sequences based on sequence similarity. As shown by Blasczk et al.[304], the ancestral lineage of the alleles

is reflected by highly systematic mutations and characterize the intronic variability in HLA. The reconstruction step resulted in 10,779 reconstructed sequences for 6,489 partial alleles. The sequence similarity calculation was based on distance matrices computed using Clustal Omega 1.2.0[305]. Distance matrices were computed for the set of all alleles with complete sequence information and alleles with the same exon sequence availability based on concatenated exon sequences. All nearest neighbors for each partial allele were used for the reconstruction of sequences. A leave-one-out validation of the sequence reconstruction quality was performed based on the reconstructed intron sequences in comparison to the original sequences. For each allele with full sequence information, introns 1, 2, and 3 were removed and reconstructed using nearest neighbors that are identified on the basis of exon 2 and 3 sequences. This resulted in a similarity of 99.89% ($\pm$ 0.43%), which corresponds to an edit distance error of 1.2 on average over the combination of three introns. For alleles of the same loci, the sequence similarity between introns amounts to 97.36% ($\pm$ 2.15%).

3.2.2 Read Alignment

Read alignment was performed with RazerS3 3.1, released as part of the open source C++ library project SeqAn[202,306]. RNA-Seq reads were mapped against the constructed nucleotide CDS reference library. Reads of WES and WGS were mapped against the constructed genomic nucleotide reference library. The applied parameter settings (`--percent-identity 97 --distance-range 0`) result in all best alignments for every read and a sequence identity of at least 97%. The maximum number of reported best matches (`--max-hits`) was set to infinity. Further, we performed read alignment with Bowtie 2[199] on one data set.

3.2.3 Hit Matrix Construction

Based on the read alignment results, a binary matrix $C^{R\times L}$ was constructed for all reads $r \in R$ mapping to at least one allele $a \in L$ of the reference. In the case of paired-end data, the matrix was constructed for each read pair individually. A successful alignment of read r to allele a is denoted by $C_{r,a} = 1$, unsuccessful alignment with $C_{r,a} = 0$ respectively. Matrix rows of reads from paired-end data were combined with a point-wise `AND` with their matching pair. Reads without mapping mate reads were discarded. The matrices were filtered for alleles whose four-digit subtype is not available in the allele frequency database[307] or dbMHC[308]. Further, reads with identical rows, mapping to the same alleles, were combined and represented by a row weight vector o_r. Completely covered alleles were removed from the matrix by dropping corresponding columns. An allele b is defined to cover allele a if $(C^T_{:,a}C_{:,b} = C_{:,a}) \wedge (|C_{:,a}| < |C_{:,b|})$ with $a, b \in L$. Allele b therefore covers all reads which map to a but additionally explains other reads. The resulting matrix was used for model construction.

3.2.4 Optimization Problem

The main assumption of our algorithm is that the true HLA genotype explains more mapped reads than any other combination of HLA alleles. Therefore, we want to determine the combination of up to six major and six minor HLA-I alleles which maximizes the number of explained reads. We formulated the underlying optimization problem as an ILP, which guarantees an optimal solution for a linear objective function subject to linear constraints and integrality requirements on the variables[309]. The ILP is defined as:

$$
\begin{aligned}
&(O1) \qquad \max_{S \subseteq L} \sum_{r \in R} o_r \cdot (y_r - \beta \cdot g_r) - \sum_{a \in L^R} \gamma \cdot x_a \\
&\textbf{s.t.} \\
&(C1) \qquad \forall X \in \{A,B,C,G,H,J\} \qquad \sum_{a \in X} \chi_a \leq \tau^{\max} \\
&(C2) \qquad \forall X \in \{A,B,C,G,H,J\} \qquad \sum_{a \in X} \chi_a \geq \tau^{\min} \\
&(C3) \qquad \forall r \in R \qquad \sum_{a \in L} \chi_a \cdot C_{r,a} \geq y_r \\
&(C4) \qquad \forall r \in R \qquad g_r \leq \tau^{loci} \cdot \gamma_r \\
&(C5) \qquad \forall r \in R \qquad g_r \leq \sum_{a \in L} \chi_a - n^{loci} \\
&(C6) \qquad \forall r \in R \qquad g_r \geq (\sum_{a \in L} \chi_a - n^{loci}) - n^{loci} \cdot (1 - \gamma_r)
\end{aligned}
\tag{3.1}
$$

A binary variable χ_a was introduced with $\chi_a = 1$ if a is part of the solution set $S \subseteq L$ for each allele $a \in L$. For each read $r \in R$, a binary variable y_r denotes if read r is explained by one of the selected alleles $a \in S$. The diploid nature of the human genome is reflected by constraints $C1$ and $C2$ which enforce the selection of at least one ($\tau^{\min} = 1$) and at most two alleles ($\tau^{\max} = 2$) per locus. Alleles from the major (HLA-A, -B, -C) and minor loci (HLA-G, -H, -J) are given by A, B, C, G, H, J. Constraint $C3$ is used to enforce $y_r = 1$ if the corresponding read r can be explained by the current solution set based on the binary matrix $C^{R \times L}$. To account for the preference of heterozygous allele combination because of spurious hits, the regularization term g_r for each read $r \in R$ was introduced, defined as follows:

$$
g(r) = \begin{cases} \sum_{a \in L} \chi_a - n^{loci}, & \text{if } y_r = 1 \\ 0, & \text{otherwise} \end{cases}
\tag{3.2}
$$

where n^{loci} describes the number of loci (here $n^{loci} = 6$).

The objective function includes the regularization term and a weighting constant β. This constant represents the portion of reads which have to be explained additionally to choose

a heterozygous solution over a homozygous solution. Depending on y_r, the constraint $C6$ enforces g_r to take on one of the limit values given by $C4$ (g_r is limited to 0 if read r cannot be explained by the solution) and $C5$ (g_r is limited to the number of heterozygous loci). The constant β was set to 0.009 based on the result of an evaluation carried out by performing nested five-fold cross-validation for different values in the range from 0.00 to 0.05 with a step size of 0.001. The evaluation was performed on a data set consisting of 253 runs of the 1000 Genomes Project[174,175] and evaluated in terms of percentage of correctly typed alleles. The data set was stratified for evenly distributed heterozygous and homozygous cases.

The ILP was formulated with the Python module Pyomo, which is part of Coopr[i] 3.3, and solved with ILOG CPLEX[ii] 12.5.

3.2.5 NGS Data Sets used for Evaluation

We conducted a comparison with previously published methods on publicly available NGS data sets with PCR-verified HLA genotyping information. The full list of the used samples and their accession IDs is given in Appendix Table E.1. Two data sets of 2 × 100 to 2 × 102 bp long Illumina HiSeq 2000 reads, previously used by Warren et al. and Kim et al., including 16 samples of a colorectal RNA-Seq study (SRP010181[32]), and 20 low coverage WGS data samples of the HapMap Project[310] were obtained from the NCBI Sequence Read Archive[308].

An RNA-seq data set (ERA002336[311], 37 nt long paired-end reads, Illumina Genome Analyzer II) derived from 50 lymphoblastic cell line samples of CEU HapMap individuals was used for the comparison with Boegel et al. and Kim et al. Data sets were obtained from the European Nucleotide Archive[312].

A comparison with Major et al.[36] was carried out based on two data sets of 41 WGS HapMap samples and 182 WES samples (1000 Genomes Project) respectively. The comparison was only based on samples with fully predicted genotypes by Major et al. (12 HapMap WGS and 161 1000 Genomes Project samples). Additionally, we included 253 Illumina HiSeq 2000 and Genome Analyzer II exome sequencing runs from the 1000 Genomes Project[174,175] to this benchmark set.

Moreover, OptiType was validated on two data sets, which have been used by Major et al. They benchmarked their method on a HapMap WGS data set consisting of 41 runs, partly overlapping with those used by Warren et al., and an exome sequencing data set consisting of 182 runs of 1000 Genomes Project samples. Only samples for which Major et al. predicted full genotypes were considered, resulting in 12 HapMap WGS and 161 (1000 Genomes Project) data sets. We expanded this benchmark set by including additional data from the 1000 Genomes Project[175] consisting of all 253 Illumina HiSeq 2000 and Genome Analyzer II exome sequencing runs.

[i] `https://pypi.org/project/coopr.pyomo`
[ii] `https://www.ibm.com/products/ilog-cplex-optimization-studio`

Further, we compared OptiType with ATHLATES based on their publicly available benchmark data set, including 11 samples from the 1000 Genomes Project[293].

To evaluate the influence of HLA enrichment, two samples derived from the same patient were sequenced on an Illumina HiSeq 2500 with 101 bp long reads. Additionally, one of the samples was enriched with a SureSelectXT Human All Exon V5 kit (Agilent Technologies; Böblingen, Germany), the other with a custom SureSelect HLA kit provided by Michael Wittig (Institute of Clinical Molecular Biology, Christian-Albrechts-University of Kiel, Germany) and Agilent Technologies.

3.3 Results

In this section, we first show the overall performance of our NGS-based HLA typing method in comparison to state-of-the-art tools on RNA-Seq, WES, and WGS data. We then evaluate the influence of intronic reconstruction, HLA enrichment, and coverage depth.

3.3.1 Overall Performance

We evaluated OptiType's performance in comparison to HLAminer, ATHLATES, seq2HLA, HLAforest, previously published computational methods, and the method by Major et al. on publicly available data sets. Overall, OptiType achieved an accuracy of 97.1% (CI_{95}: 96.1–97.80%) on four-digit level and 99.3% (CI_{95}: 98.7–99.7%) on two-digit level on the 361 benchmark samples. This corresponds to 939 of 950 correct heterozygous and 127 of 133 correct homozygous loci predictions.

As depicted in Figure 3.3, OptiType outperforms the above-mentioned methods on all data sets. The increase in accuracy ranges from 4 to 15% which corresponds to a 65 to 83% lower rate of incorrect allele predictions. On a small subset of 11 samples, ATHLATES showed comparable performance. The data set was initially used for benchmarking their algorithm.

OptiType achieved an average accuracy of 97.6% (CI_{95}: 96.7–98.4%) when applied on all 253 paired-end Illumina exome sequencing runs of the 1000 Genomes Project, where 667 of 676 (98.7%) heterozygous and 80 of 83 (96.4%) homozygous loci were typed correctly.

3.3.2 Influence of Intronic Reconstruction

To assess the influence of intron sequence reconstruction, which is used for WES and WGS data, we evaluated the performance on data from the 1000 Genomes Project. The reference database was therefore constructed by taking only exon 2 and 3 sequences into account. In order to avoid the loss of reads located at the exon boundaries, we performed read alignment with Bowtie 2 in local alignment mode and the same mismatch tolerance as in the default mapping procedure using RazerS3.

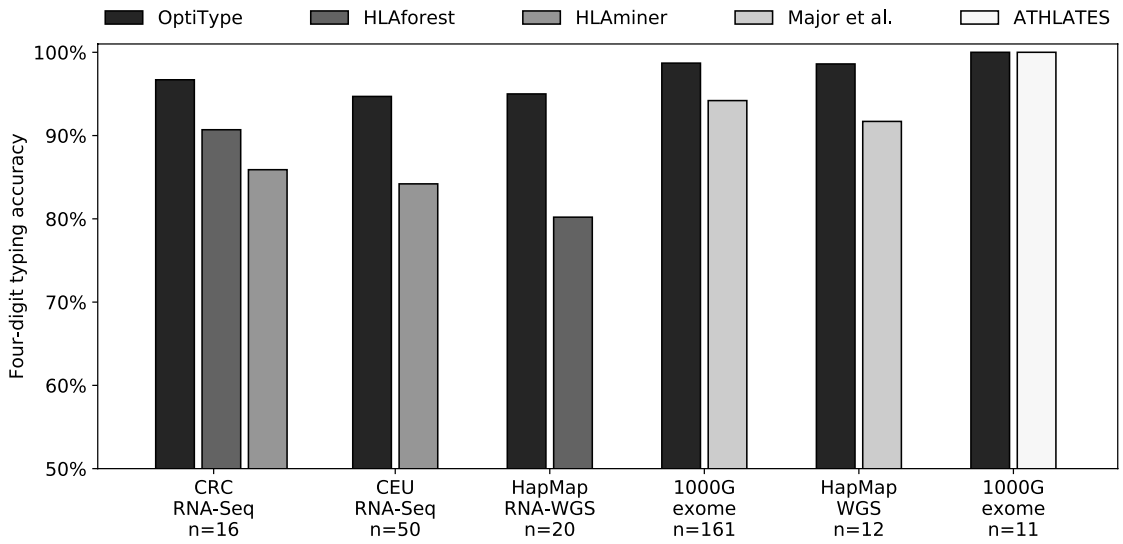

Figure 3.3: Performance in comparison to state-of-the-art HLA typing algorithms. OptiType's performance was evaluated in comparison to four other published algorithms on publicly available data sets including RNA-Seq, WES, and WGS data. The performance is measured in terms of four-digit typing accuracy of the major HLA-I loci. Figure adapted from Szolek et al.[301].

The mapping of the paired-end reads was not successful in a significant amount of cases due to the length of exons 2 and 3 which is ~270 bp, respectively. Thus, the hit matrix generation was done in two distinct ways: (1) considering only mapped reads (2) allowing mapped reads without corresponding mates. For (1) OptiType achieved an accuracy of 93.5% (CI_{95}: 91.8–95.1%) and 90.6% (CI_{95}: 89.0–92.3%) for setting (2). In comparison to the default OptiType pipeline, including intronic information, this corresponds to a 2.7- to 3.9-fold increase in error.

3.3.3 Influence of HLA Enrichment and Coverage Depth

To analyze the effects of HLA enrichment on OptiType's prediction accuracy, we examined a sample with an average coverage depth of ~4,100× on HLA-I loci. Therefore, the HLA genotype was predicted for subsets of reads by randomly extracting a decreasing number of reads from the complete sample as a simulation of scenarios with different coverage depths. With a coverage of ~12×, which corresponds to ~0.3% of the total amount of reads, OptiType still correctly predicted the genotype. This amount of reads corresponds to as little as ~15% of the exome sample of the same subject without specific HLA enrichment.

Further, we conducted a simulation study using all 1000 Genomes Project exome samples. In total, resampling (>4,000×) with a restricted number of reads was applied to 253 individual samples. An average coverage depth of 10x on HLA-I loci showed to be sufficient to achieve an accuracy of 95%. The results of this experiment as well as the investigation of the effects of

shorter read lengths (2 × 37 bp), which showed little impact on the prediction accuracy, are shown in Figure 3.4.

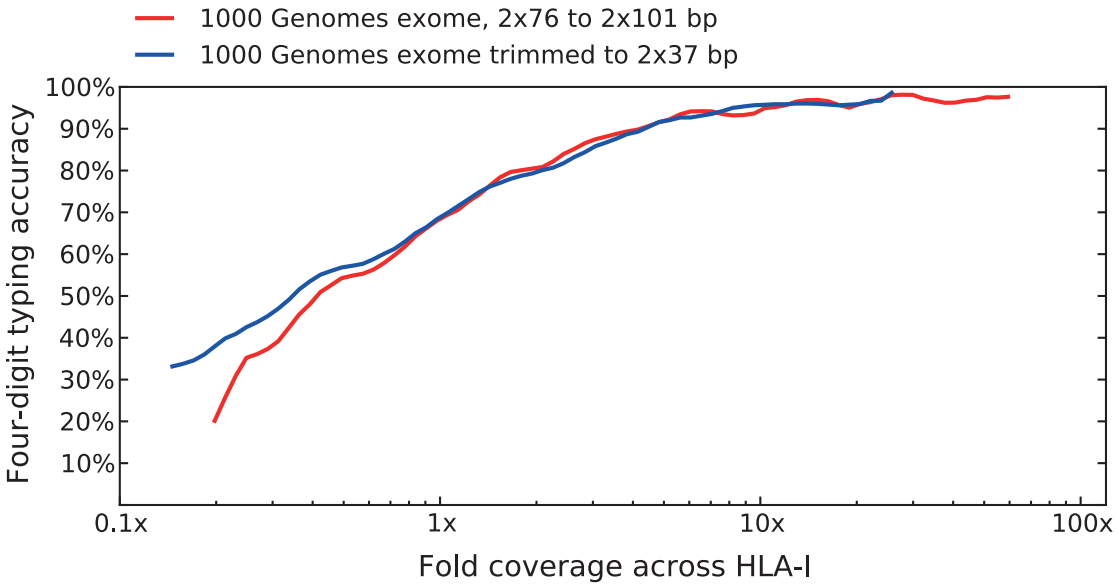

Figure 3.4: Influence of read length and coverage on the prediction accuracy. To simulate different coverage depth conditions, reads of 253 exome sequencing runs (1000 Genomes Project) were subsampled (>4,000×). The same subsampling procedure was applied after trimming the original reads to 37 bp to assess the impact of read length. Results were evaluated with respect to four-digit typing accuracy and showed a minor effect of read length. An average coverage depth greater than 10x over the HLA-I loci yields maximal accuracy[301].

3.4 Discussion

HLA genotyping is of vital importance for medical areas like regenerative and transplantation medicine. With the increasing relevance of new therapeutic approaches like precision medicine and personalized medicine and their clinical applications such as the design of personalized vaccines, the development of HLA genotyping methods with short turnaround times, cost efficiency, and high four-digit level accuracy becomes even more important. State-of-the-art experimental HLA typing methods usually involve labor-intensive techniques which require the generation of data with the sole purpose of HLA typing. However, decreasing costs for NGS and its' broader availability in clinical settings provide the foundation of *in silico* approaches based on NGS data.

Previously published *in silico* HLA typing approaches did not show sufficiently accurate prediction results, especially concerning clinical applications. This was mainly due to assumptions made by these methods, including the isolated inference of HLA genotypes for each locus.

Therefore, we implemented OptiType which does not include these limiting assumptions and includes further information to yield correct HLA typing results on four-digit resolution. Our benchmark on data sets from RNA-Seq, WES, and WGS technologies with read lengths ranging from 2 × 37 bp to 2 × 101 bp showed an accuracy of 99.3% (CI_{95}: 98.7–99.7%) on two-digit-level and of 97.1% (CI_{95}: 96.1–97.80%) on four-digit-level typing. On 361 runs, OptiType correctly predicted the zygosity for 939 of 950 heterozygous and 127 of 133 homozygous loci, corresponding to an accuracy of 98.4% (CI_{95}: 97.5–99.1%). Due to it's applicability to data from different NGS techniques, the improved accuracy with respect to two- and four-digit HLA typing, and short run times, OptiType presents an excellent alternative to previously published *in silico* approaches.

Improved accuracy could be mainly achieved due to the simultaneous consideration of all major and minor HLA-I loci, the adherence of an equal *a priori* chance for every allele to be identified, and the inclusion of intronic information. We considered only exon 2, exon 3, and flanking intron sequences to minimize the disadvantage of alleles with partial sequence information. The unavailability of intronic information for incomplete alleles was thereby tackled by the phylogeny-based reconstruction of intron sequences. This approach of a tailored construction of a reference sequence database could even be extended to other regions with an increasing number of full HLA allele sequences available. However, the prediction will still be restricted to the used reference and, therefore, can only predict known alleles.

As shown by the conducted simulation study, the coverage depth of samples above a certain level, as well as the read length, do not have a strong influence on the prediction accuracy. This is in line with the reported observation that the number of covered bases has a stronger influence than coverage depth[32].

Problematic cases in terms of ascertaining the correct genotype usually result from the absence of any reads for sequence segments, the constitution of a heterozygous locus by alleles with high sequence similarity, and low coverage on distinguishing segments. Further, unresolved ambiguities arise for specific genotypes with reads mapping to minor and major loci. Additionally, one limiting factor concerning achievable accuracy on data sets is the presence of inaccurate experimental typings as previously observed[300]. In general, a confidence measure would be desirable for reported HLA genotypes. However, attempts to calculate a confidence measure on the basis of enumerated solutions and their objective values were not successful.

Since OptiType's publication, new approaches[34,313–318] and reviews[319,320] on *in silico* HLA typing methods have been published. Reported accuracy values suggest that OptiType is still among the most accurate HLA-I genotyping methods (Figure 3.5). For WES and RNA-Seq data, OptiType achieved the highest accuracy values in all direct comparisons. On the WGS benchmark data set[318] used by Xie et al. to evaluate the performance of their method xHLA, OptiType achieved a slightly lower accuracy (97%) than xHLA (99.7%) and HLA*PRG[316] (98.5%) for 488 samples. However, HLA*PRG has the drawback of very high computational demands

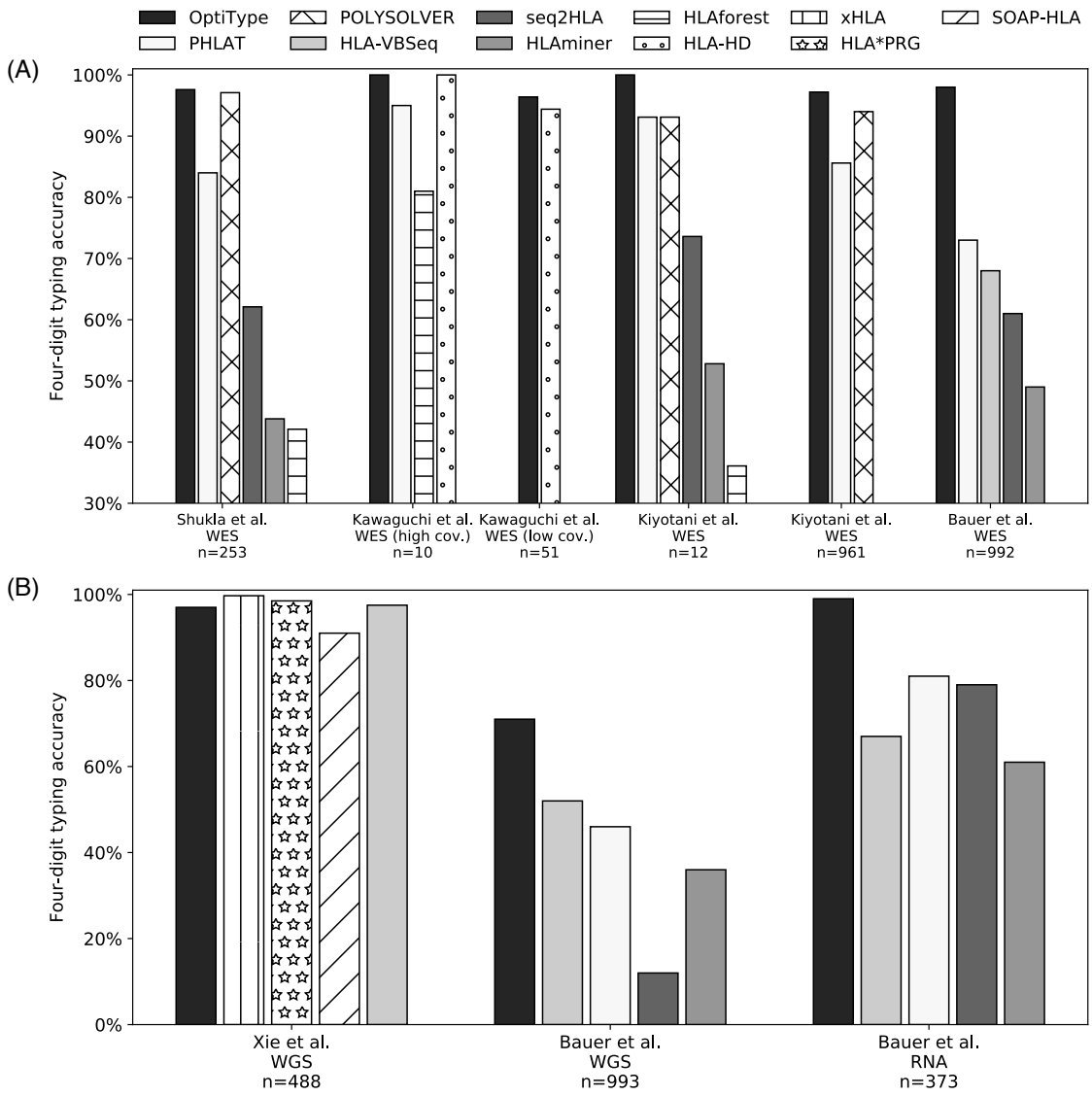

Figure 3.5: Published HLA typing benchmarks including *OptiType*. (A) Performed benchmarks of *in silico* HLA typing methods based on WES data sets. (B) Reported four-digit HLA typing accuracies on WGS and RNA-Seq data sets. The reported accuracies originate from reviews by Kiyotani et al.[319] and Bauer et al.[320] on available *in silico* typing methods and publications on new proposed HLA typing algorithms by Shukla et al.[314], Kawaguchi et al.[317], and Xie et al.[318].

and requires ~30–250 CPU hours per sample as reported[318]. As with OptiType, xHLA uses an ILP-based approach initially based on information from exon 2 and 3 only. Integration of information based on protein-level alignment, multiple sequence alignment-based alignment expansion, and subsequent refinement iterations seem to increase performance. Similar features could be integrated into OptiType as well to improve its performance.

One advantage of xHLA and other approaches are undoubtedly their ability to perform HLA typing for class II. Still, this feature could be included in an updated version of OptiType. Run times, which are typical in the range of minutes to a few hours per sample (including read mapping) depending on the number of reads and read lengths, and memory consumption could still be improved. Possible changes include the replacement of RazerS3 with a different read mapper like Yara[204] or alternative alignment strategies in the initial mapping step.

Chapter 4

T-Cell Immunogenicity: Modeling Immunological Tolerance

4.1 Introduction

Immunogenicity is the ability of an antigen to induce a humoral or cell-mediated immune response. However, only a small portion of peptides encoded by a foreign antigen usually induces strong responses. This phenomenon is referred to as immunodominance[321]. The potential of antigen fragments (*epitopes*) to generate an immune response is thereby dependent on properties of the peptide itself and systemic features, such as central and peripheral tolerance. Some amino acids have shown to be associated with immunogenicity, as well as specific positions in the peptide influence immunogenicity to a more considerable extent than others[322]. The premise for T cell-mediated immunogenicity is the presentation of peptides on HLA class-I and -II molecules. This process is dependent on the antigen processing (Chapter 2.2.2) within APCs and nucleated cells, as well as the binding to HLA molecules. Still, the potential to induce a T-cell response is dependent on the presence of a suitable T-cell clone for the corresponding pHLA complex in the T-cell repertoire. Further, pHLA complexes are formed with self and non-self peptides, demanding T cells to be able to discriminate between them. The T-cell repertoire is initially shaped through negative and positive selection processes in the thymus (*central tolerance*). T cells with moderate affinity to self-pHLA complexes are positively selected. Negative selection ensures self-tolerance by depletion of naïve T cell progenitors with high affinity for self-antigens bound to HLA. Another mechanism ensures differentiation of self-reactive cells into Foxp3$^+$ T_{reg} cells[323]. In addition to the induction of tolerance to peptides conserved in the proteome[324], there is evidence for similar mechanisms for T cells reactive against peptides of commensal microorganisms, such as bacteria in the gut microbiome[44,45]. T cells remain tolerant to non-self peptides which exhibit high sequence similarity to self-peptides[325]. Assessment of immunogenicity is of high importance to guarantee the efficacy of therapies, such as

cancer vaccines[326], or to minimize the risk of adverse reactions for agents like biotherapeutics[327]. Immunogenicity estimates could especially guide the selection of epitopes in the field of neoantigen-based vaccines which is a crucial aspect of their development[328]. Traditionally, predicted pHLA affinities are used as immunogenicity estimates. Many approaches for the prediction of HLA class-I and class-II binding affinities are available and have been reviewed[329]. However, even though binding to HLA molecules is a prerequisite for immunogenicity, pHLA affinity does not correlate with the strength of induced immune responses[330,331]. Therefore, it is not enough to assess HLA binding to ensure HLA-restricted immunogenicity[332].

Related Work

To improve the prediction of immunogenicity estimates, methods[37–40] have been developed that include further metrics of the antigen processing pathway, such as proteasomal cleavage and TAP transport, in addition to HLA binding. However, the performance improvements were rather weak. Calis et al.[322] developed a linear regression model to weight epitope position based on their influence on TCR recognition. Data was derived from previous studies on the analysis of peptide recognition by T cell clones and the structure of pHLA-TCR complexes[333–336]. In 2007, Tung et al. proposed a machine learning approach using an SVM on physicochemical properties encoded peptides[41]. Although this approach was later extended to advanced string kernels[42], it did not exceed an accuracy of 68% for HLA-A2. Toussaint et al. further increased the prediction performance by combining sequence properties with self-tolerance information encoded as the distance of the peptide in question to the 100 closest peptides in the proteome, defined as *distance-to-self*[43]. Recently, Rasmussen et al. developed an artificial neural network approach combining HLA binding prediction methods and pHLA stability estimates[337,338]. The combined approach showed an increased prediction performance of CTL epitopes in comparison to the methods alone[338]. This observation is in line with the conjecture that the stability of pHLA complexes correlates better with immunogenicity than the binding affinity[339–341] which has been confirmed for an HLA-A*02:01-restricted T cell epitope set in vaccinia virus infections[342].

Project Outline

In this work, we propose a method for the prediction of immunogenicity which is based on *distance-to-self*[43] measures. We extended this approach by modeling *peripheral tolerance*. A previously described[43,343] memory-efficient tree-like data structure (*trie*) is used to store large sets of peptides and compute pairwise distances efficiently. Through consideration of gut microbiome data for the distance calculation, we extend the set of considered peptides and account for potential peripheral tolerance selection mechanisms. Every query peptide is encoded by three feature vectors that are used for the employed machine learning approach (Figure 4.1).

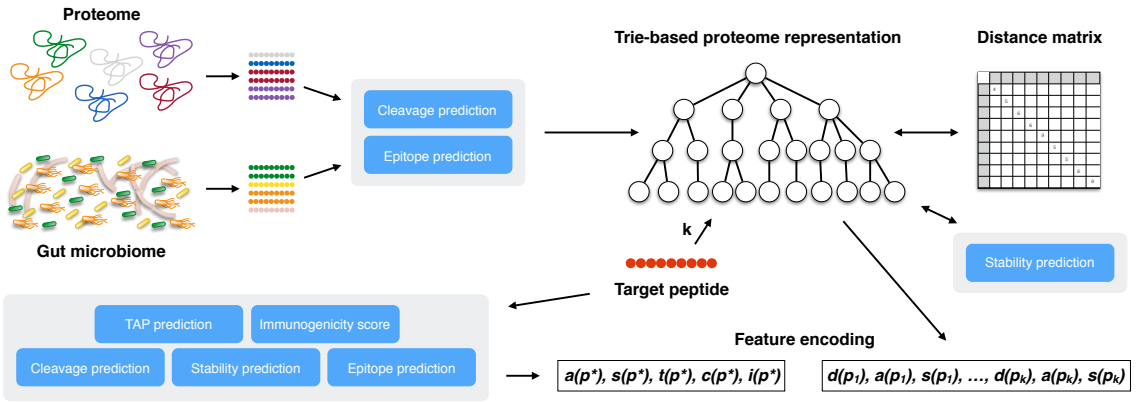

Figure 4.1: Simplified scheme of our immunogenicity prediction approach. We construct memory-efficient tree-like data structures (*tries*) that contain peptides from proteome and gut microbiome data. Peptides are filtered for cleavage and HLA binding. For every target peptide p^*, we compute the distance to the k closest peptides of the self-peptide set using a BLOSUM-derived distance matrix. For the feature vector construction, we add HLA-binding affinities and peptide-HLA (pHLA) complex stability estimates for every of the k closest peptides. Further, we compute properties of the target peptide such as HLA-binding affinities, cleavage scores, and pHLA complex estimates and construct a five-dimensional feature vector. Additionally, we encode the peptide sequence itself. The feature vectors are then used in an implemented multiple-kernel support vector regression approach.

We use a feature encoding based on the BLOSUM50-encoded peptide sequence, **distance-to-self**, binding affinity, and stability of the k closest peptides, and antigen processing measures of the target peptide.

4.2 Materials and Methods

In this section, we give information on the data sets used for modeling central and peripheral tolerance. Further, we describe the calculation of distance-to-self measures and the feature encoding. We evaluated our implemented prediction method on an experimentally derived data set described here. Additionally, we provide a detailed description of the employed machine learning approach.

4.2.1 Modeling Central and Peripheral Tolerance

To account for self-tolerance of T cells, we include a representative set of self-peptides presented to T cells. We consider human proteome data to model central tolerance, as previously published[43]. Additionally, we include data derived from the human gut microbiome to model peripheral tolerance. Not every peptide contributes to tolerance since the antigen has to be processed, and the corresponding peptides have to be presented on HLA. Therefore, we predict cleavage sites using NetChop[235] 3.1 with a threshold of 0.5 (default value) and consider only

the respective resulting peptides. As the binding of peptides to HLA is a prerequisite for the T-cell recognition, we further use NetMHCpan[247] 3.0 to predict HLA binding for the respective allele and only consider binders, i.e., peptides predicted to bind with affinities (predicted IC50) of $\leq 500\,\text{nM}$ (score > 0.425).

Proteome Data

The human proteome data was retrieved from the UniProtKB/TrEMBL database[344] (accessed 2016-07-04).

Thymus Proteome Data

As a presentative thymus proteome data set, we used the maximum thymus proteome (thymus-max)[43,345]. The selection of proteins was made based on whole genome microarray data. The thymus-max data set contains all proteins that are present and marginally expressed in the thymus[345].

Gut Microbiome Data

Human gut microbiome data was derived from the NIH Human Microbiome Project[177,346] database (accessed 2014-10-20) as protein FASTA file. The complete data set contained 2,019,324 sequences in total from 382 distinct organisms. Since the data set comprises organisms that have been isolated in different individuals and samples, we created three data sets of representative gut microbiomes. Arumugam et al. reported the identification of three clusters (referred to as enterotypes) with respect to overrepresented genera in the human gut microbiome[347]. According to this classification we created subsets of the complete gut microbiome

Table 4.1: Representative human gut microbiome data sets. We split the full human gut microbiome reference into three subsets representing the three enterotypes as previously defined[347].

Identifier	Genera	Number of Sequences
gut-et1	Acidaminococcus, Bacteroides, Roseburia, Faecalibacterium, Anaerostipes, Parabacteroides, Clostridiales	413,052
gut-et2	Prevotella, Streptococcus, Enterococcus, Desulfovibrio, Lachnospiraceae	216,760
gut-et3	Akkermansia, Alistipes, Klebsiella, Ruminococcus, Escherichia/Shigella, Dialister, Mitsuokella, Methanobrevibacter, Eggerthella, Ruminococcaceae, Subdoligranulum, Coprococcus, Collinsella, Blautia, Eubacterium, Dorea	225,809

reference (Table 4.1) in the following referred to as *gut-et1*, *gut-et2*, and *gut-et3*. The resulting protein FASTA files contain all protein sequences of organisms of the overrepresented genera.

Distance-to-self Measure

To calculate the distance between two peptides, we employed the distance definition by Toussaint et al.[43]. The distance between a peptide and a set of peptides is defined as the smallest pairwise distance to one peptide of the set. We used a distance measure derived from the BLOSUM45 substitution matrix[348]. The symmetric 20×20-matrix D is generated as previously published[43]: each entry a_{ij} of the substitution matrix A is replaced by $\frac{a_{ij}+a_{ji}}{2}$, resulting in a symmetric matrix A'. The non-negative matrix A'' is generated by shifting all entries by the absolute minimum value of A'. Further, all entries are divided by the maximum entry of A'', resulting in a normalized matrix A'''. The distance matrix D is computed by subtracting each entry of A''' from 1: $m_{ij} = 1 - a'''_{ij}$. Efficient computation of distances, based on the distance matrix, was realized by a memory-efficient trie-based approach[43,343]. Tries, where each peptide is represented by a leaf, were generated from the sets of peptides. All peptides represented by a path from the same node have a common prefix.

4.2.2 Feature Encoding

Each target peptide p^* is encoded by three feature vectors. The first vector $\Phi_s(p*)$ is constructed from the target peptide sequence using a 20-dimensional amino acid encoding derived from the BLOSUM50 substitution matrix. Furthermore, we calculate the HLA binding affinity for the respective allele $a(p*)$ with NetMHCpan[247] 3.0, pHLA-complex stability estimates $s(p^*)$ using NetMHCstabpan[338] 1.0, TAP transport efficiency estimates $t(p^*)$ using SMMTAP[237] 1.0, the cleavage score $c(p^*)$ using NetChop[235] 3.1, and the the immunogenicity propensity score[322] $i(p^*)$. The generated 5-dimensional feature vector is defined as follows:

$$\Phi_i(p^*) = [a(p^*), s(p^*), t(p^*), c(p^*), i(p^*)].$$

For the construction of the tolerance feature vector, we calculate the distance of the target peptide p^* to the k closest peptides of the proteome, the binding affinity of the k closest peptides, and the pHLA-stability estimates, resulting in the following feature vector $\Phi_d(p^*)$:

$$\Phi_d(p^*) = [d(p_1), a(p_1), s(p_1), \ldots, d(p_k), a(p_k), s(p_k))].$$

In the following, we will use $k = 100$, resulting in a 300-dimensional feature vector. We implemented the feature vector generation using FRED2[26].

4.2.3 Immunogenicity Prediction

For the prediction of immunogenicity for a given target peptide, we employed multiple kernel learning with support vector classification and one kernel on each feature vector. We used a *local alignment string* kernel on the encoded peptide sequences and a *Gaussian radial basis function* (RBF) kernel on the feature vectors encoding immunological properties ($\Phi_i(p^*)$) and self-tolerance ($\Phi_d(p^*)$), respectively. The prediction functionality was implemented using the machine learning toolbox Shogun[349] and accessed via the Python interface.

4.2.4 Evaluation Data Set

For the evaluation of our approach, we used the same data set as previously described by Toussaint et al.[43]. Originally, the data set was provided by the Department of Immunology (Tübingen, Germany). The data set comprises nonameric peptides derived from Epstein-Barr virus. Peptides predicted to bind to HLA-B*35:01 using SYFPEITHI[238] and manually selected ones were tested for T-cell reactivity using the enzyme-linked immunosorbent spot (ELISPOT) assay. In total, the data set comprised 151 peptides, including 49 immunogenic and 102 non-immunogenic peptides. The number was reduced to 45 positive and 45 negative data points by Toussaint et al. to adapt the binding affinity distributions for both cases and therefore preclude the learning of HLA binding instead of immunogenicity[43].

4.3 Results

In our benchmark, we assessed if the incorporation of gut microbiome data to model peripheral tolerance improves the prediction performance for T-cell reactivity. Further, we incorporated features with respect to processes of the antigen processing pathway and evaluated the different models on one experimentally validated data set.

4.3.1 Self-tolerance Data

From the initial sets of proteins *thymus-max*, *gut-et1*, *gut-et2*, and *gut-et3*, we generated all 9-mer peptides with a netChop score greater than 0.5. Initially, the *thymus-max* set included 9,584,195, *gut-et1* 33,621,547, *gut-et2* 18,835,603, and *gut-et3* 29,238,321 unique peptides of length nine. On average, we observed 30.0% $\pm$ 4.7% of the initial amount of peptides in the resulting sets after filtering based on the proteasomal cleavage predictions. The number of peptides predicted to bind to HLA-B*35:01 averaged at 510,369. This corresponds on average to 2.3 $\pm$ 0.3% of the whole peptide set and 7.5 $\pm$ 0.4% of the filtered set. Hence, the following number of peptides for the four data sets were considered for trie generation: 218,780 (*thymus-max*), 836,449 (*gut-et1*) , 471,199 (*gut-et2*), and 515,049 (*gut-et3*).

4.3.2 Prediction Performance

We evaluated different models to assess the benefit of incorporating stability estimates and gut microbiome data to model peripheral tolerance. The performance was evaluated based on the mean *area under the Receiver Operating Characteristic* (auROC) using stratified nested five-fold cross-validation. We used the approach suggested by Toussaint et al. as baseline (auROC=0.78) which included a Gaussian RBF kernel on BLOSUM50-encoded peptide sequences and a Gaussian RBF kernel on 201-dimensional features vectors encoding self-tolerance based on *thymus-max*.

All tested models included the Gaussian RBF kernel on BLOSUM50-encoded peptide sequences. We did not observe a performance increase in comparison to Toussaint et al. when incorporating gut microbiome data. The combination with the three gut microbiome data sets *gut-et1* (auROC=0.73), *gut-et2* (auROC=0.73), and *gut-et3* (auROC=0.72) resulted in a decreased performance in terms of the mean auROC. Similar performances were observed when including stability estimates to the *thymus-max* model for the target peptide only (auROC=0.71) and additionally for all closest 100 peptides (auROC=0.74) without gut microbiome data. We observed the best performance when we used *gut-et1* with stability estimates for the target peptide and the closest 100 peptides (auROC=0.79). This was not the case for *gut-et2* (auROC=0.72) and *gut-et3* (auROC=0.73). Incorporation of TAP, cleavage, and immunogenicity propensity scores to the previously mentioned models yielded a mean auROC of 0.78 when using *gut-et1* in addition to *thymus-max*. We observed worse auROC values when using *gut-et2* and *gut-et3* which resulted in a mean auROC of 0.73 in both cases. The performances of selected models in comparison to the baseline model are depicted in Figure 4.2.

4.3.3 Integration in ImmunoNodes

To improve the availability of our approach for calculating the distance of given peptides to a set of peptides, we integrated the functionality in the immunoinformatics toolbox ImmunoNodes[260]. The toolbox is fully integrated into the visual workflow environment KNIME[271] and can be easily used within workflows. We provide one node (`Distance2SelfGeneration`) to generate custom reference *tries* for a given protein FASTA file. The length of the included peptides can be specified by the user. Additionally, the node `Distance2SelfCalculation` empowers users to calculate the BLOSUM-derived distance of the k closest peptides in a pre-calculated or custom build reference *trie* for a list of peptides. Further, ImmunoNodes includes pre-calculated *tries* for the human reference proteome, generated from all peptides of length 8-11 AA.

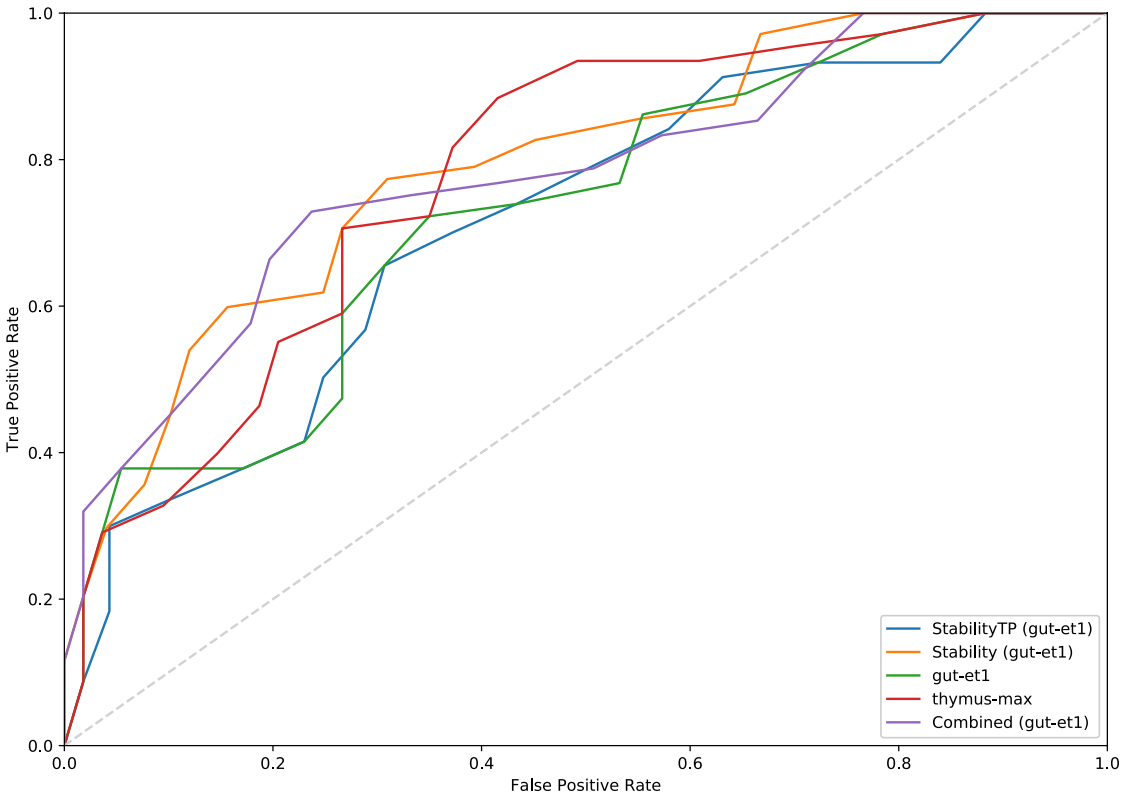

Figure 4.2: Performances of different implemented models for T-cell epitope prediction. Given are the mean ROC curves over stratified nested five-fold cross-validation. The approach by Toussaint et al. (red) includes a Gaussian RBF kernel with BLOSUM50 encoding for the target peptide and a self-tolerance model based on *thymus-max*. The other models also include a Gaussian RBF kernel with BLOSUM50 encoding, a self-tolerance model based on *thymus-max* and gut microbiome data (*gut-et1*), stability estimates for the target peptide (*StabilityTP*), additionally stability estimates for all 100 closest peptides (*Stability*), and further TAP, cleavage, and immunogenicity propensity scores (*Combined*).

4.4 Discussion

The selection of peptide candidates for epitope-based vaccines, such as cancer vaccines, is based on the immunogenic potential of the corresponding peptide to maximize the capability of inducing an immune response. Thus, there is a need for the accurate assessment of T-cell epitopes. To reduce costs and turn around times, *in silico* methods are desirable. However, current *in silico* prediction methods are not reliable enough to be used in biomedical applications. The two major prerequisites for a T-cell response are the presentation of a peptide in a stable pHLA complex and its recognition by a reactive T-cell clone with a suitable TCR. The repertoire of reactive T-cell clones is mainly shaped by thymic selection which ensures tolerance against self-peptides (*central tolerance*). Toussaint et al. suggested a prediction method[43,345], which models central tolerance by incorporating distances of the target peptide to a set of

self-peptides. The approach is based on the assumption that antigenic peptides that are very similar to self-peptides are highly unlikely to induce a T-cell response. Therefore, the method incorporated system-wide properties to the prediction model and thereby distinguished itself from existing purely sequence-based methods. We suggested a method that additionally models *peripheral tolerance* that ensures the protection of the host against self-reactive T cells in the periphery. Since peripheral tolerance includes exposure to antigens of the gut microbiome, we included human gut microbiome data in the generation of representative self-peptide sets. Incorporation of stability estimates for the target peptide and the 100 closest peptides, as well as gut microbiome data (*gut-et1*), led to a slight increase in prediction performance (auROC=0.79) in comparison to the method by Toussaint et al. (auROC=0.78). Other applied models yielded similar or reduced prediction performance. The results suggest that the incorporation of stability estimates and gut microbiome data do not have enhancing effects on the prediction performance. As reported by Toussaint et al.[43], the generation of tries from the whole proteome did not increase the performance. Similarly, it might be necessary to apply even more rigid filters for the gut microbiome based on information of the commensal microbiota. To render this possible, more sequencing data and studies on the gut microbiome are needed. Recently, Carrasco Pro et al. presented indications that microbial sequence similarity is relevant when assessing immunogenicity[350]. They investigated BLOSUM62-derived distances of HLA class II non-epitopes and epitopes to the human microbiome. The microbiome similarity was found to be associated with a decreased or increased likelihood of immunogenicity[350]. As stated, they did not detect significant effects for HLA class I epitopes[350]. Recently, a similar *distance-to-self* measure was employed in a study on neoepitopes. Bjerregaard et al. reported that the distance of neoepitopes to self-peptides is a predictor of immunogenicity under certain conditions[351]. This only seems to be the case if the neopeptide and the corresponding wild-type peptide have comparable HLA-binding affinities. The similarity was calculated by the kernel similarity measure[352] published by Wen-Jun Shen et al., which calculates similarity based on k-mer matching using a BLOSUM similarity measure. Based on this, the tool MuPeXi[353] identifies tumor-specific peptides and assesses their immunogenic potential based on HLA-binding estimates of the mutant and wild-type peptide, expression levels, mutant allele frequency, and a penalization term based on the self-similarity. Still, the reported performance (auROC=0.63) was rather weak for three tumors with available T-cell reactivity data.

We suggested an approach based on the incorporation of gut microbiome data. It has been shown that the immune system is in contact with intestinal microbiota and shaped by commensal bacteria such as bacteria in the gut microbiome[44,45]. Further, it was discovered that the intestinal microbiota has an impact on the effectiveness of cancer immunotherapy[41,354,355]. However, the exact mechanisms are still not known. One important question to ask is if immune cells, especially T cells, are in contact with all bacteria that reside in the gut lumen

or if we should only consider a specific subset of organisms for the modulation of peripheral tolerance. Moreover, it is not just the gut microbiome which contributes to the human microbiota. Potential interactions between microbes of the human skin microbiome and the immune system should be considered as well[356]. In the last decade, the number of *in silico* HLA-I binding prediction methods has grown significantly. Besides that, methods were suggested for the prediction of other mechanisms involved in antigen processing, such as proteasomal cleavage and TAP transport. Still, the performance of T-cell epitope prediction algorithms, which often combine HLA binding prediction and other antigen processing prediction methods suggest, that it requires more than binding motifs. In addition to the polymorphic nature of HLA molecules and their interaction with the peptide, one has to consider the interaction of TCRs with pHLA complexes. Therefore, the study of T-cell repertoires and the diversity among TCRs, especially with respect to complementary determining regions (CDRs) might contribute to the understanding of immunogenicity-determining factors. In 2017, two studies reported the identification of CDR sequence motif-based clusters within epitope-specific T-cell repertoires[357,358]. Glanville et al. developed the algorithm GLIPH[358] (Grouping of Lymphocyte Interactions by Paratope Hotspots) for the identification of TCR clusters, while Dash et al. suggested the similarity measure TCRdist[357] for T-cell receptors. Efforts in decoding the antigen specificity based on T-cell receptor sequences are critical for improving our understanding of the mechanisms of T cell-mediated immune responses. Recent developments in high-throughput sequencing methods, such as immune repertoire sequencing[359] (Rep-Seq), and single-cell RNA-Seq technologies increase the feasibility of obtaining TCR sequences. Data resources, such as the database VDJdb[360], a thorough and curated storage of TCR sequences together with T-cell specificity assays, might help to decipher T-cell specificities and further increase immunogenicity prediction accuracies to an acceptable level for clinical applications.

By integrating our approach for the calculation of the *distance-to-self* measure in the immunoinformatics toolbox ImmunoNodes[260], we facilitated future developments of T-cell epitope prediction methods incorporating self-tolerance models.

Chapter 5

iVacPortal – A Web-based Portal for Personalized Vaccine Design

Parts of this chapter were published in:

---

qPortal: A platform for data-driven biomedical research

Mohr, C.*, Friedrich, A.*, Wojnar, D., Kenar, E., Polatkan, A. C., Codrea, M. C., Czemmel, S., Kohlbacher, O., and Nahnsen, S.

PLoS ONE, 13(1), e0191603 (2018)

* Joint first authors

5.1 Introduction

As outlined in Chapter 2.5.1, more accurate, affordable, and faster genome sequencing technologies such as Illumina NovaSeq[361] drive the explosion in data generated by high-throughput experiments. In the case of NGS data, the daily production is in the range of terabytes by only one state-of-the-art instrument[362]. In addition to the large amount of data generated solely by omics technologies such as NGS, multi-omics approaches aim at concluding information of multiple layers, for instance by the integration of data from genome and proteome level. Further, experiments in such studies, especially in biomedical research, are based on several replicates to increase statistical power. Therefore, it is of great importance to record experimental variables, biological properties and to grasp the connection between patients, extracted tissue, as well as generated raw data. Collaborations with several project partners across different labs, and

the integration of large-scale databases, such as the *1000 Genomes Project*, further increase the complexity and demand stringent mapping procedures of multiple sample identifiers[175]. Thus, digital management platforms that support the entire project and data lifecycle continuously gain importance within multi-omics projects. Support mechanisms include sophisticated experimental design procedures and data integration strategies. Research consortia, such as the International Cancer Genome Consortium (ICGC), try to employ centralized means of coordination and publicly available data access points to large-scale collections of biomedical data for researchers on a global scale[363,364]. Critical requirements for such efforts, which promote big data in biomedicine[365], are elaborated approaches for data standardization and experimental metadata storage. The fundamental demand for annotating new experimental data presented through centralized portals comes with several advantages. Most importantly, it can drastically improve the opportunities for sharing data with a broader scientific community. As a consequence thereof, new options for data mining and new big data approaches that benefit from the more correlative power of leveraged data are provided[366]. However, such centralized solutions also introduce new hurdles on the infrastructure and computational side. Further challenges include the preservation of data access and availability of analysis pipelines for different data types of various data sources. In the case of clinical data, data security and possibilities to check for data integrity are especially important. Despite the complexity of these tasks, elaborated backend systems have to ensure fast and efficient access to resources. Web-based solutions are an established approach to provide collaborators access to data, metadata, and analysis tools through a centralized interface. In order to provide analysis capabilities for high-throughput biomedical data, web-based solutions require connections to large-scale computing resources, such as HPC clusters, grids, or compute clouds. Here, workflow systems can provide interfaces for the implementation and scheduling of established and standardized analysis pipelines. The portal interfaces and workflow management systems such as gUSE[272] also provide new solutions to scientists with different proficiency levels related to computational skills.

5.1.1 Related Work

Since biomedical research fields usually entail particular requirements to centralized interfaces, in recent years, portal-based solutions have been mainly developed for domain-specific applications. A web-based solution for proteomics research, including identification and quantification, is offered by the Swiss Grid Proteomics Portal (iPortal)[367,368]. Its functionality is based on the Swiss Protein Identification Toolbox swissPIT[369]. One of the most prevalent solutions in genomic research is Galaxy[264] which combines an open web-based platform and workflow system, initially developed for the analysis of genomic data. In 2006, GenePattern[47] 2.0 was initially published. The web-based platform provides tools for gene expression, sequence variation, copy number variation, flow cytometry, and network analysis. Due to the recognized

importance of capturing additional data and metadata, efforts on augmenting such platforms with additional means of metadata management are made. Various Laboratory Information Management System (LIMS)[370] solutions or even larger automated systems[371] have been proposed. Other platforms have a specific focus on big data, such as the cBio Cancer Genomics Portal[280]. cBioPortal was developed for cancer research and in particular the visualization of large-scale genomics data. As cancer research projects and biomedical projects in general often include sensitive data when it comes to data privacy, solutions such as mediGRID have been implemented[372]. In medical research, such solutions additionally have to deal with the integration of heterogeneous data. The collection of web-based platforms covers a variety of solutions for areas of research other than proteomics and genomics. The Molecular Simulation Grid (MoSGrid) offers users access to computational pipelines in the context of computer-aided drug design[373]. Common tasks include docking and virtual screening. Other solutions include the neuroscience gateway e-BioInfra[374] and the web-based portal for phylogenetic analyses CIPRES[375].

5.1.2 Project Outline

The presented solutions encompass a wide range of purposes, ranging from data management of different omics technologies and bioinformatics workflow management to specialized visualization applications. We implemented *qPortal* in order to provide a web-based platform which serves various purposes for quantitative biological data. qPortal provides users with intuitive ways for the management and analysis of large-scale data. Implemented backend solutions employ a variety of established concepts and technologies, including relational databases, data stores, data models, and data transfer capabilities. The backend builds the foundation of implemented front-end solutions empowering users to conduct data management and data analysis. The implemented data models guarantee efficient and standardized ways of annotating data and querying metadata through the integrated data management system *open Biological Information System* (openBIS)[376]. Efficient data query mechanisms allow for the computation of statistics and future re-analysis via coupled workflow management systems on HPC systems. This integration of project and data management, as well as workflow management systems, tackles the issues of decentralized data generation, storage of experimental metadata, and the need for easy-to-use means of data analysis. Although qPortal shares features of some previously published platforms, the integration in one place presents clear advantages over existing solutions. Our portal serves similar purposes as Galaxy, yet both implement different concepts. Throughout the complete project cycle, qPortal empowers users to conduct experiments by offering options from experimental design to visualization of results. These features for all-digital project management especially differentiate the data-driven approach (*qPortal*) from a workflow-driven approach (*Galaxy*).

As domain-specific applications might still need tailored implementations of user interfaces, data management, and analysis pipelines, we implemented our system in a modular manner to allow for easy extension. One example of such a domain-specific application is the development of personalized/individualized (cancer) vaccines in immunotherapy. Such projects typically include multiple data sources, data types, collaboration partners, and a variety of analysis pipelines. In order to implement a one-stop solution for these projects, we implemented *iVacPortal*. Our central integration platform for all these efforts – and central point of access for the clinical researchers – qPortal builds the foundation for iVacPortal. iVacPortal serves as the central web-based workbench for data management and analysis in personalized cancer vaccine studies as illustrated in Figure 5.1. Necessary data processing and analysis steps during

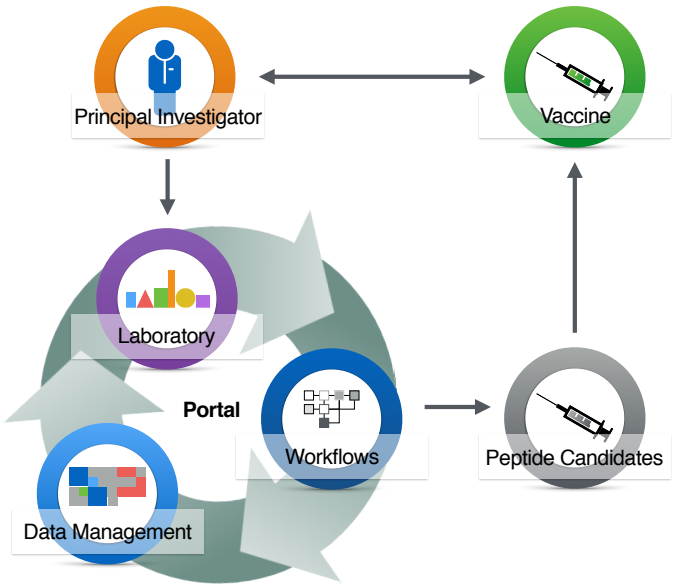

Figure 5.1: Schematic flow diagram of *iVacPortal*. Principal investigators add patients, the experimental setup, and metadata through qPortal. Automated means of data transfer from laboratories enables efficient registration of measured data. Subsequently, data management is done via qNavigator. Implemented workflows are applied on registered data sets to compute possible peptide candidates which are then selected for the final vaccine.

this kind of study, such as variant calling, HLA typing, and epitope prediction were implemented as workflows. Means of data transfer and defined data processing steps enable entry points for clinical researchers at any stage of the defined personalized (cancer) vaccine design pipeline. Simplified user interfaces for the setup and management of these projects further increase the usability of iVacPortal for users with different computer knowledge. Thus, iVacPortal provides an end-to-end solution from the analysis of somatic mutations to the selection of peptide candidates for cancer vaccines.

5.2 Materials and Methods

All bioinformatics methods which were used for the implementation of workflows are described in the following section. Further, we provide details on the data sets employed in the case study. Software components, which are part of the infrastructure and implementation of qPortal, as well as implementations for iVacPortal, are described in Section 5.3.

5.2.1 NeoOptiTope

NeoOptiTope is a Python-based software package for RNA-Seq based neoepitope selection[i]. The selection is implemented based on three different models which are used dependent on available input data. The main immunogenicity function $O1$ incorporates expression data, HLA alleles, and immunogenicity or HLA binding predictions for each specified HLA allele:

$$(O1) \qquad \max_{S \subseteq E} \sum_{e \in E} x_e \sum_{g \in G(e)} \log_2(a_g) \sum_{h \in H} \log_2(a_h) i_{e,h}$$

s.t.

$$(C1) \qquad \sum_{x \in E} x_e = k$$

$$(C2) \qquad \sum_{x \in E_{TAA}} x_e \leq k_{TAA}$$

$$(C3) \qquad x_{e_1} + x_{e_2} \leq 0 \quad \forall e_1, e_2 \in O$$

$$(C4) \qquad \sum_{e \in I(h)} x_e \leq y_h \quad \forall h \in H$$

$$(C5) \qquad \sum_{h \in H} y_h \geq \tau^{HLA}$$

$$(C6) \qquad \sum_{e \in E(g)} x_e \leq \zeta_g \quad \forall g \in G$$

$$(C7) \qquad \sum_{g \in G} \zeta_g \geq \tau^{anti}$$

$$(C8) \qquad d_{e,h} \leq \tau^{dist} \Rightarrow x_e = 0 \quad \forall e \in E, \forall h \in H$$

Here, E denotes the set of all epitopes, E_{TAA} the set of TAA epitopes, H the set of HLA alleles, $I(h)$ the set of epitopes which bind to HLA h, $E(g)$ the set of epitopes from gene, protein, or allele g, $G(e)$ the set of genes or proteins from which epitope e could originate from, O the set of epitope pairs that share a sequence of defined length, and H the set of all given HLA alleles. The abundance of gene g (a_g), the abundance of allele a (a_h), and the predicted immunogenicity $i_{e,h}$ of epitope e bound to HLA allele h have to be available. $C1$ ensures that

[i] `https://github.com/APERIM-EU/WP3-EpitopeSelector`

exactly k epitopes are selected with up to k_{TAA} TAA epitopes, whereas two epitopes may not share a substring of a predefined length ($C3$). $C4$ determines if allele h is covered by at least one of the selected epitopes e. At least τ^{HLA} alleles have to be covered ($C5$). $C6$ determines if a gene or protein g is covered by at least one of the selected epitopes e. $C7$ ensures coverage ζ_g of at least τ^{anti} genes and proteins. If distance-to-self measures $d_{e,h}$, which can be calculated as described in Chapter 4, for epitope e based on HLA allele a are available, the objective function is optimized with constraint $C8$. The third model expects uncertainty measures of the immunogenicity predictions to give an estimate for the risk of the vaccine not being effective. In addition to objective function $O1$, the decision is based on the second objective function $O2$:

$$(O2) \qquad \min_{S \subseteq E} \sum_{e \in E} x_e \sum_{h \in H} \sigma_{e,h}$$

where $\sigma_{e,h}$ denotes the uncertainty estimate of the immunogenicity prediction of epitope e for allele h.

5.2.2 Data Sets for Case Study

The data sets used during the case study are publicly available through the 1000 Genomes Project data portal[377]. A detailed list of used data sets is given in Appendix Table E.2. All data analyzed during this study was included in a previous publication[378].

5.3 Design and Implementation

qPortal serves as the frontend and, therefore, central point of access for users (Figure 5.2). The portal is connected to a workflow engine to enable users to submit workflows to an HPC cluster and perform analysis on uploaded data sets via the portlet qNavigator. The transferred data is automatically registered in the database through defined *Extract Transform Load* (ETL) processes served by the backend system openBIS and staged to the computing infrastructure via the workflow system. Data and project management tasks can be performed through qNavigator. Moreover, we implemented resources for personalized vaccine design as part of the data managing resources of qPortal and the workflow system to provide the complete analysis pipeline from raw data generation to possible vaccine compositions. qPortal is built on top of a Liferay[284] 6.2 instance. The portal includes a collection of portlets, which are web applications written in Java. The portlets are implemented using the open-source framework Vaadin[380], that is based on Ajax and Google Web Toolkit. The main portlets, qNavigator (Project Browser), qFlow, and qWizard (Project Wizard), are written in Java 1.7 using Vaadin 7. Users have access to these portlets after successful authentication in qPortal using their credentials through the Liferay[284] UI. The Liferay instance is running on a Tomcat[281] 7 server instance.

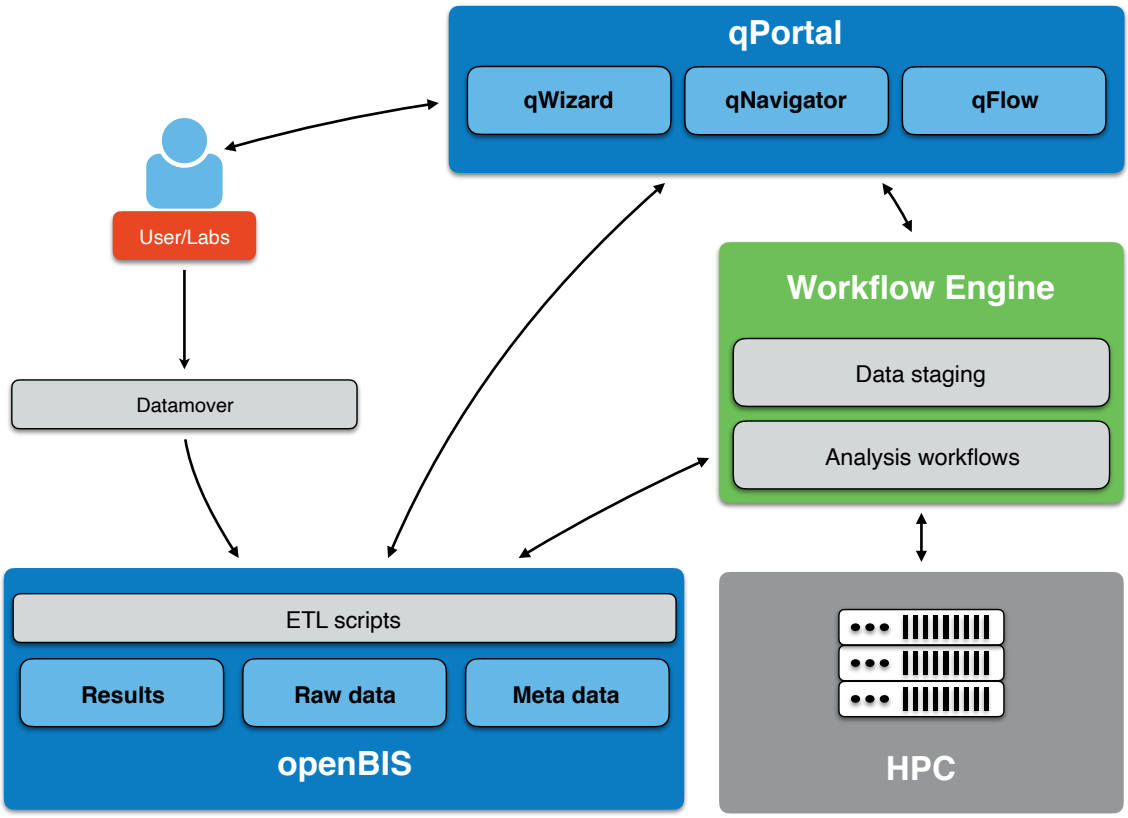

Figure 5.2: Within our setup, qPortal serves as the frontend. It is linked to openBIS and the workflow engine. The workflow engine is connected to a high-performance computing (HPC) cluster. Users may create projects, upload their data (Datamover), analyze it, and monitor workflows through qPortal (qNavigator, qWizard, and qFlow). Results are automatically written back to the database and presented on the portal through qNavigator. Figure adapted from Mohr et al.[379].

qPortal is licensed under GNU General Public License, either version 3 of the License, or any later version, as published by the Free Software Foundation[381]. The source code is available at `https://github.com/qbicsoftware`. A running qPortal instance can be accessed on `http://qbic.life`. Documentation on qPortal is available[ii] and includes a manual on how to set up qPortal at other sites and direct links to the corresponding GitHub repositories. The main components of qPortal are shown in Figure 5.3.

In the following sections, design and implementation details of the components and the two portlets, qFlow (Section 5.3.4) and qNavigator (Section 5.3.5), as well as the iVacPortal implementations (Section 5.3.6) are given. qWizard has been described previously in detail[382].

[ii] `https://portal.qbic.uni-tuebingen.de/portal/software`

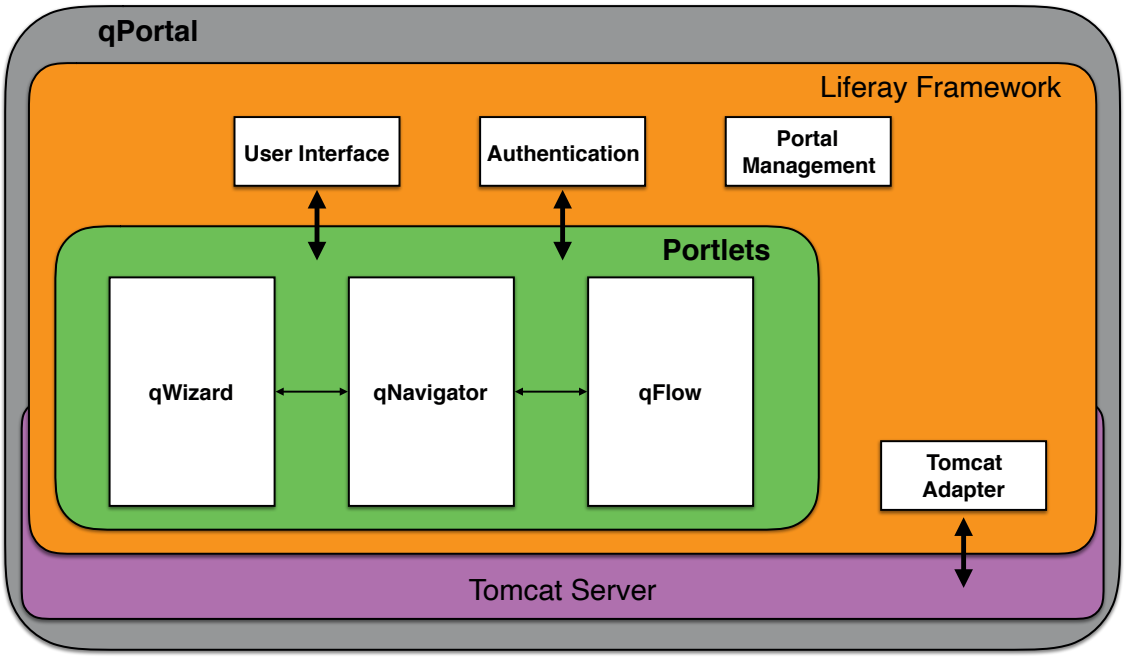

Figure 5.3: Simplified overview of the qPortal architecture. qPortal is built on top of a Liferay instance, which runs on a Tomcat server. The portal contains multiple portlets, such as qNavigator, that are deployed through the Liferay framework. Users authenticate through the user interface of Liferay.

5.3.1 Backend and Data Model

The implemented backend of qPortal uses openBIS[376]. The software package offers mechanisms for storage and management of raw data in a data store and annotation with metadata, managed via a PostgreSQL database instance. An Application server enables browsing and the management of data and metadata[383]. Management of access rights to data is implemented through a data model concept consisting of five distinct hierarchically ordered levels. On the top level, called *space*, user access roles are defined. Spaces can contain multiple projects, samples, and data sets. By definition of specific types of experiments, samples, and data sets, the openBIS data model can be customized. Common biomedical experiment types are already available[384]. Figure 5.4 illustrates the hierarchal structure of our openBIS data model implementation. Every openBIS sample has to be associated with a space. Optionally, samples can be connected to an experiment and data sets. Additionally, samples might be connected to other samples using a child or parent relationship. We implemented a data model where the *biological entity* of a project is connected to an experimental design experiment. Tissues that have been analyzed within the project are stored as *biological samples* with respective sample extraction experiments. The measured entities, including DNA, RNA, and proteins, derived from these *biological samples*, are stored as *test samples* and attached as children. Details about

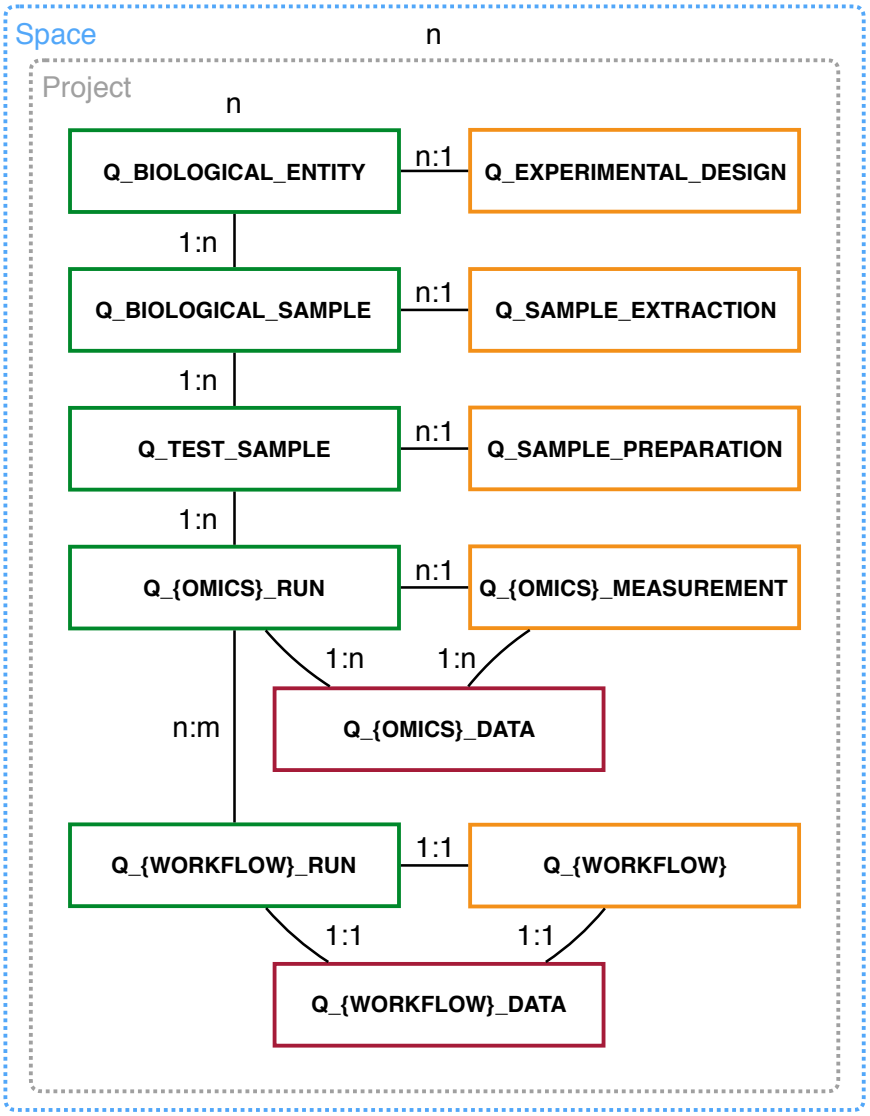

Figure 5.4: Data model as implemented in openBIS for qPortal. Spaces might contain multiple projects which usually have the depicted structure. The hierarchical data model includes different sample types (**green**), experiment types (**orange**) and data set types (**red**). Depending on the omics technology, different sample, experiment, and data set types are used. Each workflow has an implemented sample, experiment, and data set type.

the measurements, such as NGS, of the entities, are stored as experiments and samples which are connected to the corresponding *test samples*. Generated data sets are connected to those NGS measurement samples. For every available workflow, corresponding openBIS entity types, including the sample, experiment, and data set, have been implemented. Instances of these entities are connected to the openBIS entities (samples) that were used as input. The hierarchical depth is not limited to five, as illustrated in Figure 5.4, as other experiments like variant

calling might be conducted and attached to NGS samples. Results of workflow runs might also be used as input of other workflow runs. Therefore, the depth of the project structure is not limited and might increase within a project life cycle.

Sample, experiment, and data set types can be associated with multiple user-defined properties. The structured storage of metadata is an essential component of the system; metadata is attached to the representations of both the respective samples and their intangible experiments. Metadata in this context comprise information about the used protocol or similar content. The type of a *test sample* is stored as property connected to a vocabulary that contains terms such as DNA, RNA, proteins, and peptides, amongst others. Further examples include the organism specification of *biological entities* which is limited to values of the NCBI taxonomy. The use of unstructured data formats or spreadsheets is common for many scientists in the biomedical field. Since this kind of data is inherently hard to model but can contain additional metadata and is intuitive to understand, we additionally offer options to upload, display, and download unstructured data.

We implemented the connection of qPortal to the openBIS instance using the available openBIS Java API. To render queries to the openBIS datastore and the metadata database more efficient, we implemented the Java library `OpenBisClient`[iii]. The library provides wrapper functions for common data and metadata retrieval tasks, such as the retrieval of all samples for a specific project.

5.3.2 User Management

The backend (openBIS) and the Liferay instance are connected to an in-house Lightweight Directory Access Protocol (LDAP) server which stores registered users. The current setup uses the advantages of a single sign-on (SSO)[385] based solution that is already employed by other Grid web applications and portals[386]. Other protocols, such as Crowd[387] that are compatible with openBIS and Liferay, can be used to replace the user information containing resource. As described before, data and metadata access in openBIS is regulated on space level. Therefore, users have to be added to the openBIS instance by their user ID once. Through the connected user information resource, user details are added automatically. Within the associated spaces, which might include several projects and the corresponding data, roles have to be created for the users. Users or defined user groups can be assigned multiple roles. The primary login to qPortal is done on the Liferay landing page. A delegation mechanism to the backend database ensures data access. This implementation can be further extended to make use of concepts such as two-factor authentication. The information about users that are logged in to qPortal via Liferay is queried by the portlets through the implemented Java Library `LiferayAndVaadinHelpers`[iv]. This process ensures that users are only able to access the

[iii] `https://github.com/qbicsoftware/openbisclient`
[iv] `https://github.com/qbicsoftware/liferayandvaadinhelpers`

data of projects they have been granted access to. Details on the user-based data retrieval in the case of qNavigator are given in Section 5.3.5.

5.3.3 Data Transfer and Integration

To render data transfer from different sources to the central data store possible, we employ the rsync implementation of the openBIS Datamover[376,388]. During the syncing process from different locations, such as from a sequencing facility to the remote storage, checksums ensure data integrity. The data registration process is accomplished via openBIS dropboxes, which are directories on the file system, monitored for incoming files or directories and configured by property files. These property files include the incoming directory, the associated API instance, the path to an associated ETL script, as well as the data completeness condition. In our setup, marker files tag incoming data upon completed transfer. Incoming files, are forwarded to an openBIS dropbox according to their type or their origin. Every dropbox implements a Jython-based ETL routine. Depending on the implementation, ETL processes handle raw data, connected metadata, and conversions based on external tools. As outlined, our approach includes the registration of the experimental design before actual measurements take place. ETL scripts handle then the information which is collected upon sample preparation and data acquisition. This process of finalizing the experimental model usually includes the creation of additional entities, such as experiments, samples, and data sets. Besides, metadata is extracted from incoming files and stored as defined sample or experiment properties. Incoming data is then connected to data sets and moved to the data store. Listing 1 shows a simple ETL routine for the registration of files containing peptide sequences.

The association of arriving data to openBIS entities in the database is done through (sample) identifiers contained in the file or folder names. Thus, created experiments, samples, and data sets will get connected to the already existing instances by the ETL routine. We implemented ETL routines[v] for the most common file types in genomics (FASTA, FASTQ, BAM, VCF), proteomics (mzML, RAW), and others. Additionally, we developed lab-specific ETL routines to handle more complex use cases. In one case, folders including multiple FASTQ and VCF files, as well as JSON-based metadata files, are transferred at once. Therefore, the ETL routine has to include multiple processes. In this case, provided metadata such as lab-specific identifiers, the processing system, the used reference genome, and the tissue origin are stored in the database.

Furthermore, we implemented Jython-based plugins[vi] (ingestion services) for the creation of openBIS entities outside of the context of ETL routines. Ingestion services use an openBIS transaction and parameters as `java.util.Map`, with `String` keys and generic `Object` values. The transaction interface is the same as the one available in the context of dropboxes. Therefore, the same functionality as in ETL scripts can be used.

[v] `https://github.com/qbicsoftware/etl-scripts`
[vi] `https://github.com/qbicsoftware/etl-scripts/tree/master/reporting-plugins`

```
# expected code: *Q[Project Code]^4[Sample No.]^3[Sample Type][Checksum]*.*
pattern = re.compile("Q\w{4}[0-9]{3}[a-zA-Z]\w")

# Check barcode for integrity
def isExpected(identifier):
    try:
        id = identifier[0:9]
        return checksum.checksum(id) == identifier[9]
    except:
        return False

# Main function which will be triggered upon registration
def process(transaction):
    context = transaction.getRegistrationContext().getPersistentMap()

    # Get the incoming path of the transaction
    incomingPath = transaction.getIncoming().getAbsolutePath()

    ...

    # Get the name of the incoming file
    name = transaction.getIncoming().getName()

    # Parse experiment, project and sample code
    identifier = pattern.findall(name)[0]
    if isExpected(identifier):
        parentCode = identifier[:10]
    else:
        print "The identifier "+identifier+" did not match the pattern Q[A-Z]
            {4}\d{3}\w{2} or checksum"

    # Initialize search service and search for sample using the provided code
    search_service = transaction.getSearchService()
    sc = SearchCriteria()
    sc.addMatchClause(SearchCriteria.MatchClause.createAttributeMatch
        (SearchCriteria.MatchClauseAttribute.CODE, parentCode))
    foundSamples = search_service.searchForSamples(sc)

    # Get sample ID and retrieve the sample for update
    parentSampleIdentifier = foundSamples[0].getSampleIdentifier()
    parentSample = transaction.getSampleForUpdate(parentSampleIdentifier)

    # Create new peptide dataset and attach it to the found sample
    dataSet = transaction.createNewDataSet("Q_PEPTIDE_DATA")
    dataSet.setMeasuredData(False)
    dataSet.setSample(parentSample)

    # Move the file(s) to the new dataset
    transaction.moveFile(incomingPath, dataSet)
```

Listing 1: ETL routine for the registration of files containing peptide sequences. When a new file arrives, the ETL routine will get the already registered sample by the identifier contained in the filename and will attach a new data set of type `Q_PEPTIDE_DATA` to it.

5.3.4 Workflow System

The connection of qPortal to the workflow management system gUSE[272] enables the portal users to perform computations on cluster infrastructures, automating common bioinformatics workflows and data analysis steps. gUSE enables access to distributed computing infrastruc-

tures (DCIs) and a GUI through WS-PGRADE for the configuration, management, and creation of new workflows. Users might, therefore, use a local WS-PGRADE instance to create new workflows and port them to qPortal.

After submission, gUSE workflows are scheduled on the cluster and workflow jobs are managed by the workload manager of the cluster. The gUSE jobs of corresponding workflow nodes are then submitted by the cluster engine. Utilizing the workflow management system adds flexibility with respect to the underlying compute infrastructure, which can be adapted to a diverse collection of computing resources. The qPortal instance in Tübingen has been extensively used for various biomedical applications with the currently connected hardware setup. These compute resources have shown to be sufficient in around 600 conducted projects.

In order to create an interface between qPortal and the workflow system, and therefore enable the configuration and submission of workflows, we implemented the Java library `WorkflowAPI`[vii]. The structure of the `WorkflowAPI` is shown in Figure 5.5. We created a gUSE-specific implementation extending abstract classes and implementing the interface `Submitter`. gUSE workflows are submitted through the RemoteAPI. Other workflow systems can be used if classes implementing the interface `Submitter` and extending the abstract classes `Workflow` and `Node` exist. Therefore, through further extensions of the `WorkflowAPI`, the gUSE workflow system can be replaced by other workflow systems such as Snakemake[273] or Nextflow[277].

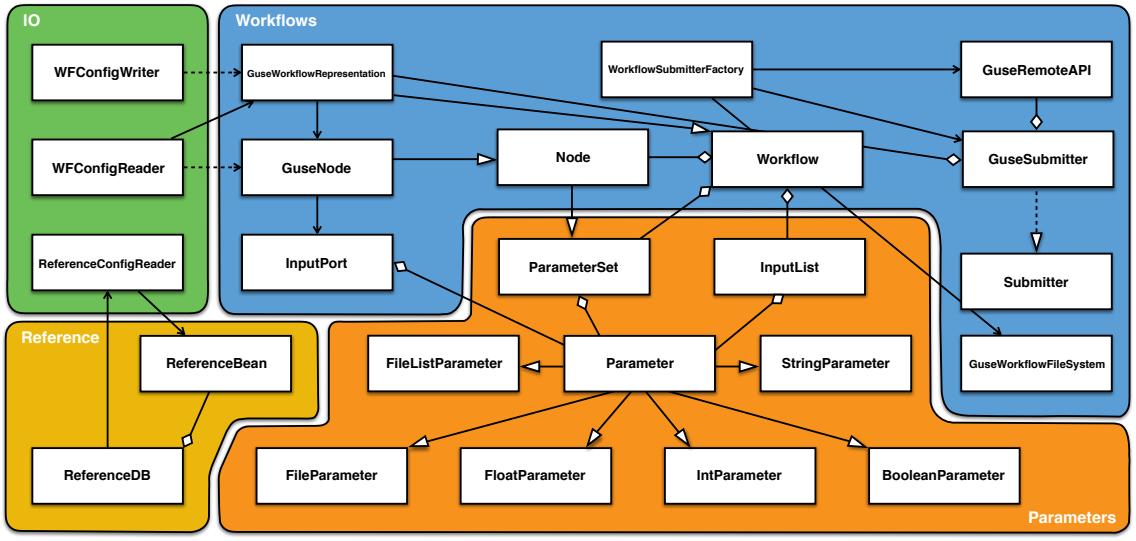

Figure 5.5: Simplified UML diagram of the `WorkflowAPI`. The classes and functionality can be broadly grouped in four packages. Implementations for new workflow engines have to extend the abstract class `Workflow` and the corresponding submitter has to implement the submitter interface.

Definition of workflow parameters is done with Common Tool Descriptors (CTDs), which are Extensible Markup Language (XML) files storing information about the execution of software

[vii] `https://github.com/qbicsoftware/workflow_api`

tools. This information typically contains specifications of parameters, input, and output files. Applications in the context of workflow conversion have previously shown the usability of CTDs[389,390]. Corresponding implementations of parameters and existing parameter types are part of the `WorkflowAPI`. Since common analysis tasks often rely on versioned reference libraries and databases, such as genome assemblies or proteomes, we implemented functionality to handle those references as part of the `WorkflowAPI`. References are defined and stored as JSON files (Listing 2). During runtime, the folder defined in the configuration file of qPortal, holding the reference configuration files, is scanned, and the JSON files are parsed (`ReferenceConfigReader`). The references are then stored as `ReferenceBean` objects and provided through the `ReferenceDB` class. Since the references have a defined type, such as `NGS`, references can displayed based on the chosen workflow. Implemented workflows are de-

```
{"reference":
   {
   "name": "Ensembl GRCh37 homo sapiens",
   "description": "Human Ensembl GRCh37 genome reference",
   "path": "/path/to/genome.fa",
   "species": "Homo sapiens",
   "version": "GRCh37",
   "type": "NGS",
   "date": "03/21/2012",
   "detailedType": "WholeGenomeFasta"
   }
}
```

Listing 2: Reference configuration file for the genome assembly *GRCh37*. The reference JSON files define the metadata of references and contain the path to the corresponding file on the file system.

fined by JSON files as well (see Appendix G). The workflow configuration files contain metadata such as the version, the description, the folder location of the corresponding gUSE workflows, and connected openBIS types. Additionally, the file contains the workflow structure with its' nodes, input ports, parameters, and input data types. The available workflow configuration files are parsed by `WFConfigReader` and corresponding `GuseWorkflowRepresentation` objects are created. In order to ease handling of workflows, we implemented the portlet collection qFlow[viii] based on the functionality provided by the `WorkflowAPI`. To enhance workflow tracking, the status of submitted workflows can be monitored through qFlow (Figure 5.6A). The status of a workflow is defined by the gUSE workflow status and the according openBIS experiment status, which is updated when a workflow has been submitted, or results have been registered. Additionally, the preparation of workflow configuration files, which has to

[viii] `https://github.com/qbicsoftware/qflow`

be carried out once for each workflow, can be done in qFlow. Available workflows can be imported and configured in the admin panel in qFlow, shown in Figure 5.6B. In this view, workflow properties, including the name, version, description, and visible parameters, can be configured. The latter specifies which parameters will be shown to the end user during the workflow submission in qNavigator. By using the information about required input data types, only workflows suitable for the types of data at hand are presented to the user. Implemented

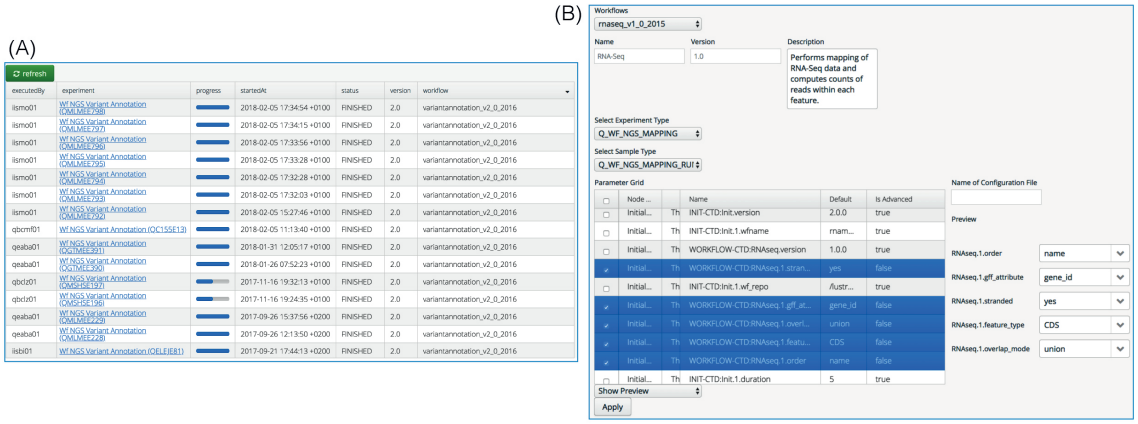

Figure 5.6: Sub-portlets of qFlow. (A) The workflow monitor shows information about workflows that belong to a user's project. Associated experiments provide a direct link to the corresponding entities in qNavigator. (B) The workflow admin panel is used to configure workflows for their usage through qNavigator. After the selection of a workflow, the name, version, and description can be set. Additionally, openBIS experiment and sample types have to be associated with this workflow. The selected workflow parameters will be shown to the user in qNavigator, as shown in the preview on the right.

gUSE workflows are composed of three workflow nodes. The data staging of selected input files occurs in the first node of these workflows. Therefore, we implemented a Python-based initialization script which parses the CTD-based input data files using `CTDopts`[ix]. Afterwards, input files, CTD-based parameter files, and additional files, such as experimental design files containing sample annotation of input files, are transferred to the newly created workspace on the cluster instance. The corresponding workflow analysis scripts are then fetched from the local GitHub repository. The actual analysis step takes place in the central node and is done through bash scripts, Python scripts, or Snakemake workflows, depending on the workflow implementation. Due to the modularity of available workflows and configuration options through parameters, many common analysis tasks can be performed by subsequent workflow runs. Appendix F provides a comprehensive list of implemented workflows with detailed descriptions. The last workflow node commits the result files to the corresponding openBIS dropbox using qProject[x]. Workflow-associated openBIS dropboxes are defined in a JSON-based configuration

[ix] `https://github.com/WorkflowConversion/CTDopts`
[x] `https://github.com/qbicsoftware/qproject`

file, and folder locations are set as node input values upon submission. The registration of results and log files is performed by workflow-specific ETL scripts, which create connected openBIS instances and update the status of the workflow.

5.3.5 Project Browser

The portlet for project management and workflow submission, *qNavigator* (Project Browser), is implemented using Vaadin 7. A simplified version of the Unified Modeling Language (UML) diagram of qNavigator is shown in Figure 5.7. All the described design and implementation details are based on release 1.6.2 (revision da6891a). The first component in the UI hierarchy

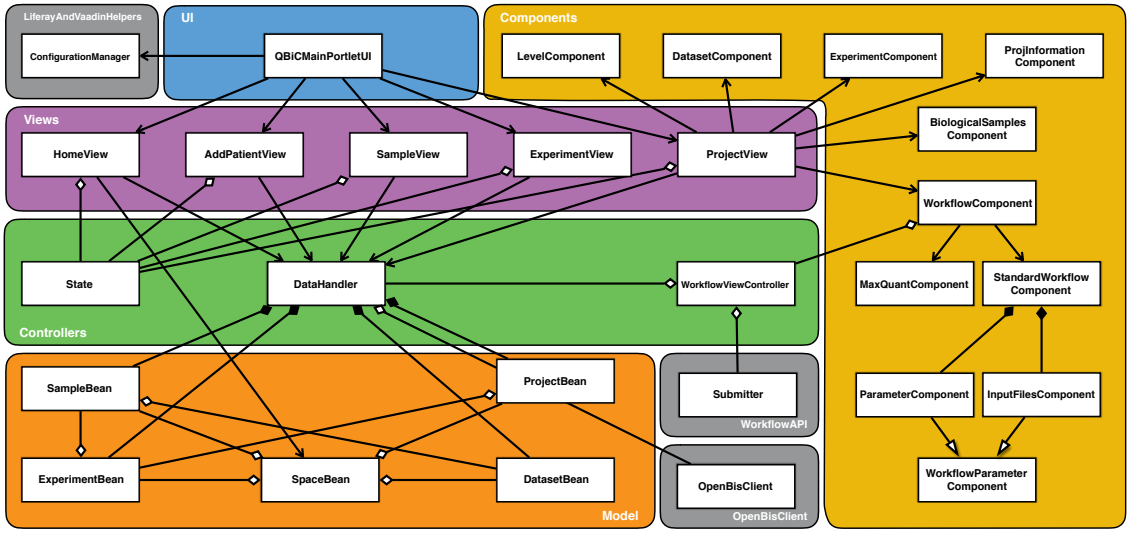

Figure 5.7: Simplified UML diagram of qNavigator. All component classes (**yellow**) extend the Vaadin class `CustomComponent`. The view classes (**purple**) extend the Vaadin class `VerticalLayout` and implement the Vaadin interface `View`. `QBicMainPortletUI` as main UI (**blue**) extends the abstract Vaadin class `UI` as the topmost component in the component hierarchy. Controller classes (**green**) enable navigation, data loading, and workflow submission. The openBIS entities are stored as JavaBeans (**orange**). Classes from external Java libraries are visualized with a **grey** background.

(`QBiCMainPortletUI`) initializes the main layout, the views, and the controllers. The main layout is a Vaadin 3×3 `GridLayout` where the vertical central component changes depending on the current user request. The different views are added to a Vaadin `Navigator` to enable the navigation between them. The navigation between different views is controlled by the `State` which extends the Java class `Observable`. Initial data retrieval is done in the `QBiCMainPortletUI` using the central data controller `DataHandler` which uses functionality provided by the `OpenBisClient`. In the beginning, users will be forwarded to the home screen (`HomeView`). Here, all projects on behalf of the currently logged-in user are shown. User identifiers are queried from openBIS; full names are given by the Liferay instance and

retrieved through `LiferayAndVaadinHelpers`. The table provides the project code, the space, the investigator, and a description, which can be filled in when the project is registered and edited later (Figure 5.8). By clicking on the corresponding project, users are forwarded

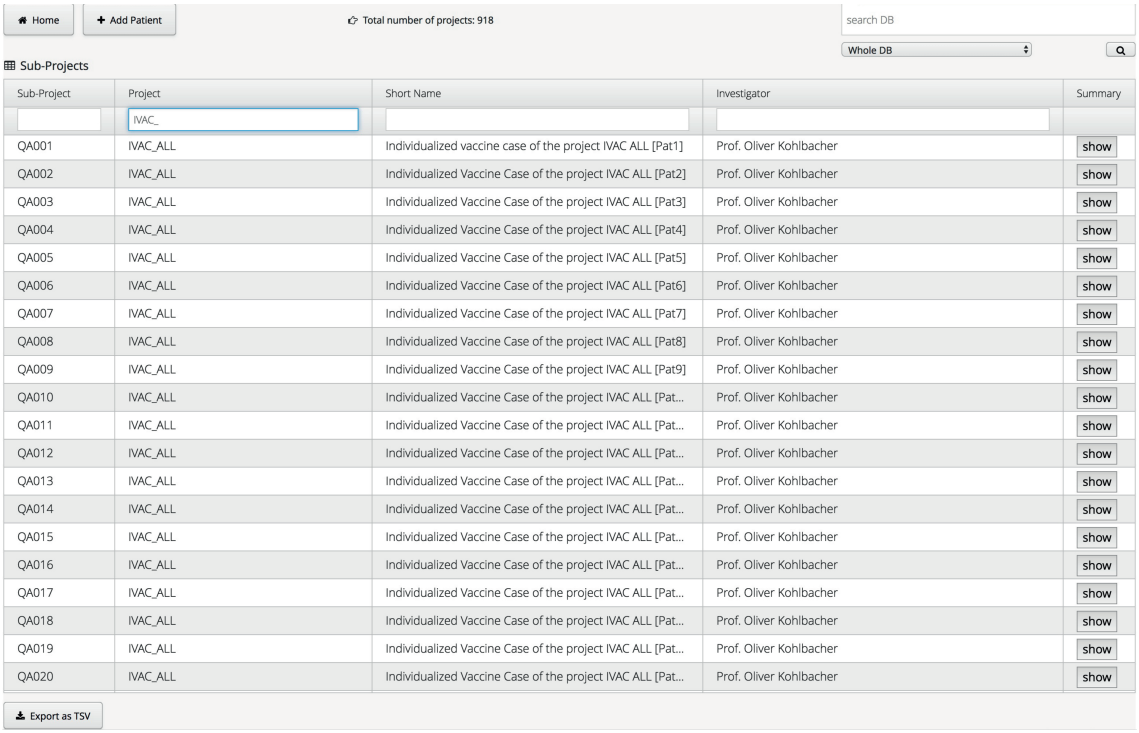

Figure 5.8: Home screen of qNavigator. The projects (sub-project) on behalf of the currently logged-in user are shown with the corresponding space (project), short name, and investigator. Users can filter their projects as shown in this case for projects starting with "IVAC_". The table content can be exported as a TSV file.

to the project view via the `State` and its implementation of the `notifyObservers` function which will call the Vaadin `Navigator` to navigate to the `ProjectView`. The view shows all information about the specific project in a horizontal view with different tabs (Figure 5.9). The first tab (`ProjInformationComponent`) contains general information about the project, such as the principal investigator and the status of defined experimental steps. Principal investigators and persons responsible for experiments are stored in an external MySQL database. Spreadsheets containing information about the sample sources, sample extracts, and sample preparations can be downloaded. A graph representation (Figure 5.10) of the first four layers of the corresponding project as registered in the database is shown in the second tab. The graph is generated using a Java graph library and describes the relationships between samples in the distinct layers (biological entities, biological samples, test samples, and measured samples). Further, the number of registered sample entities can be seen at a glance. All experimental steps and data sets are shown in the tabs "Exp. Steps" (`ExperimentComponent`) and "Datasets" (`DatasetComponent`). Every table includes the functionality to download datasets.

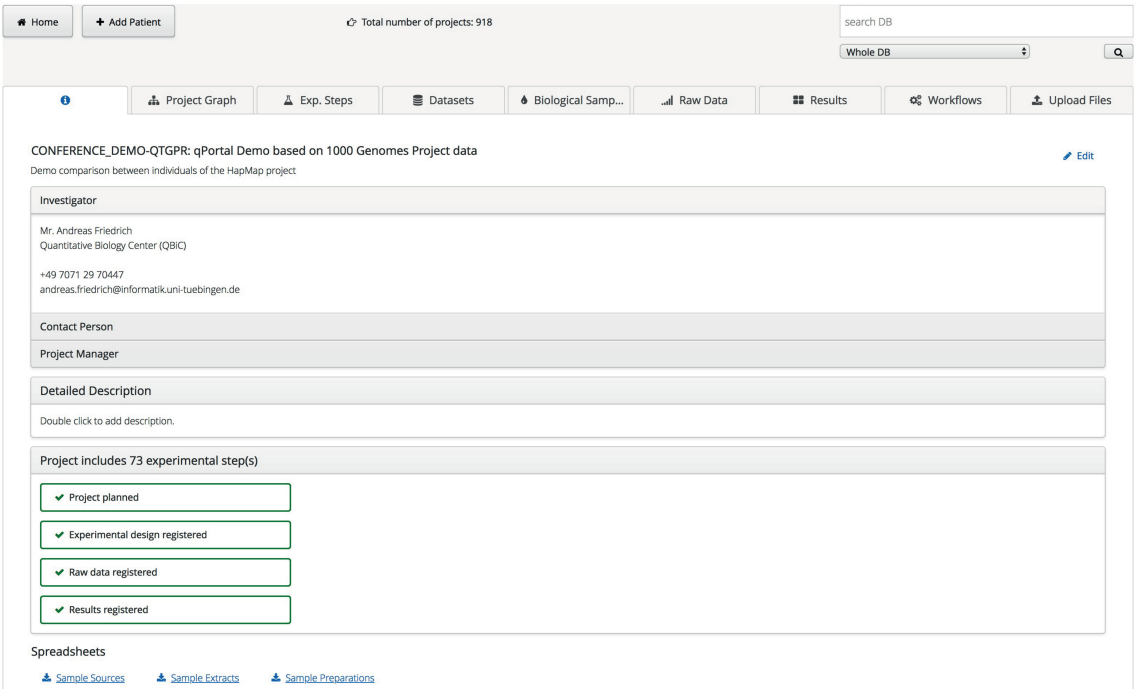

Figure 5.9: Project view of qNavigator. The view provides general information about the selected project such as the title, the investigator, and a detailed description if available. Moreover the status of the project is indicated based on the four defined steps (*project planned*, *experimental design registered*, *raw data registered*, and *results registered*).

If multiple datasets are selected, a TAR file will be generated. Files, such as HTML files and NGS or MS quality reports are directly visualized in the portlet upon selection.

In the "Biological Samples" tab, registered samples, such as organisms and their derived biological samples, are shown. Measured samples and the connected raw data are displayed in the "Raw Data" tab. OpenBIS entities are represented by corresponding JavaBeans, which are stored in Vaadin `BeanItemContainer` instances. These containers are set as data sources of tables and grids. By clicking on rows of tables that display experiments and samples, users will be navigated to the corresponding views. The `ExperimentView` and `SampleView` contains detailed information about the corresponding entities, such as registration date and property values. Additionally, it provides all data sets which are contained underneath this instance in the hierarchical project tree. Results derived from analysis runs of this project are shown in the "Results" tab. Most of those results originate from workflow runs directly triggered from the portlet and provide a direct trace to the parameter settings used to generate these results.

Workflows can be submitted from the workflows tab (`WorkflowComponent`). Depending on the available data sets in the current project, all workflows for which the mandatory input file requirements are fulfilled are presented for selection. These requirements, as well as the visualized description and version, are derived from the workflow configuration files through the

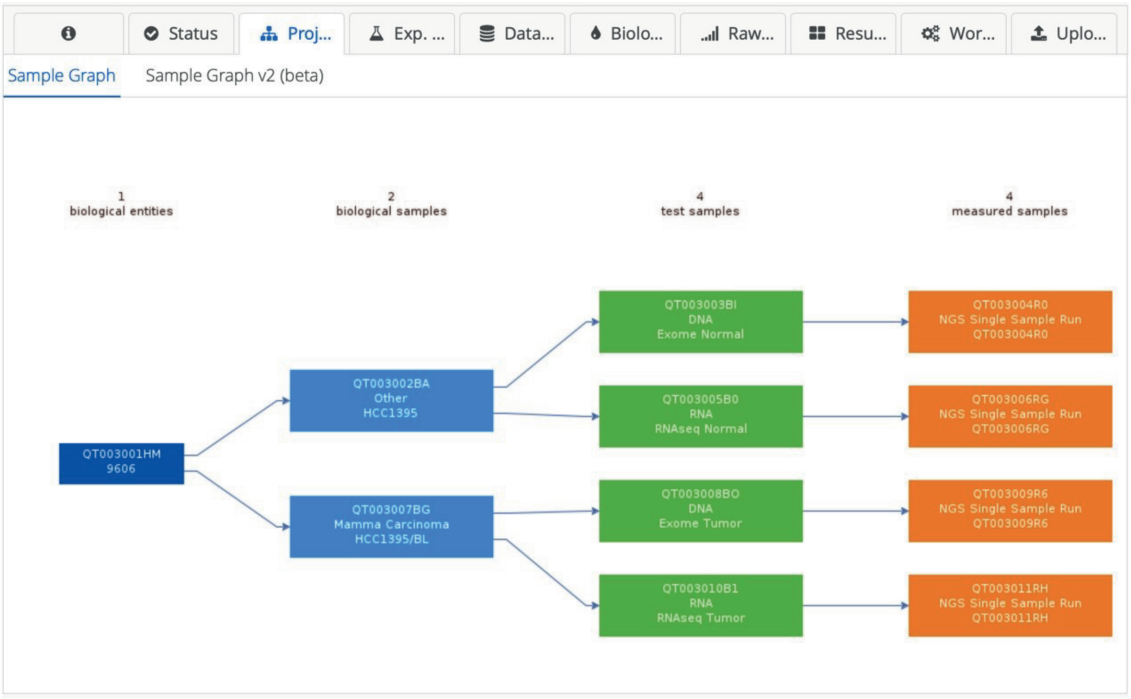

Figure 5.10: Project graph of qNavigator. The graph visualizes the first four hierarchical levels of the corresponding project. Therefore, the connection between measured samples and the biological entity can be traced back.

`WorkflowViewController`. Depending on the workflow, a specialized component, such as the `MaxQuantComponent`, or the default component (`StandardWorkflowComponent`) is initialized. The selection of a workflow will load the corresponding input file form (`InputFilesComponent`) and the parameter form (`ParameterComponent`). Upon submission of the workflow, a new openBIS sample and experiment instance of the corresponding type will be created, and the workflow will be submitted by the `WorkflowViewController`. OpenBIS entity types connected to corresponding workflows are defined in the workflow configuration files as described. Experiments and samples are registered by implemented openBIS ingestion services[xi]. Respective services are called by the `OpenBisClient`, and the information, such as the parameter values, are transferred. The openBIS experiment holds information about the workflow run, such as the specified parameter values and the execution time. Samples reflect the connection of the workflow run within the project and enables users to trace back the input of the corresponding run. Users are informed about the submission status of the workflow (Figure 5.11).

The "Upload Files" tab offers functionality to upload small unstructured data, that is directly attached to the project. Such data typically contain information from the planning stages of a project or results which have not been generated via the portal.

[xi] `https://github.com/qbicsoftware/etl-scripts/tree/master/reporting-plugins`

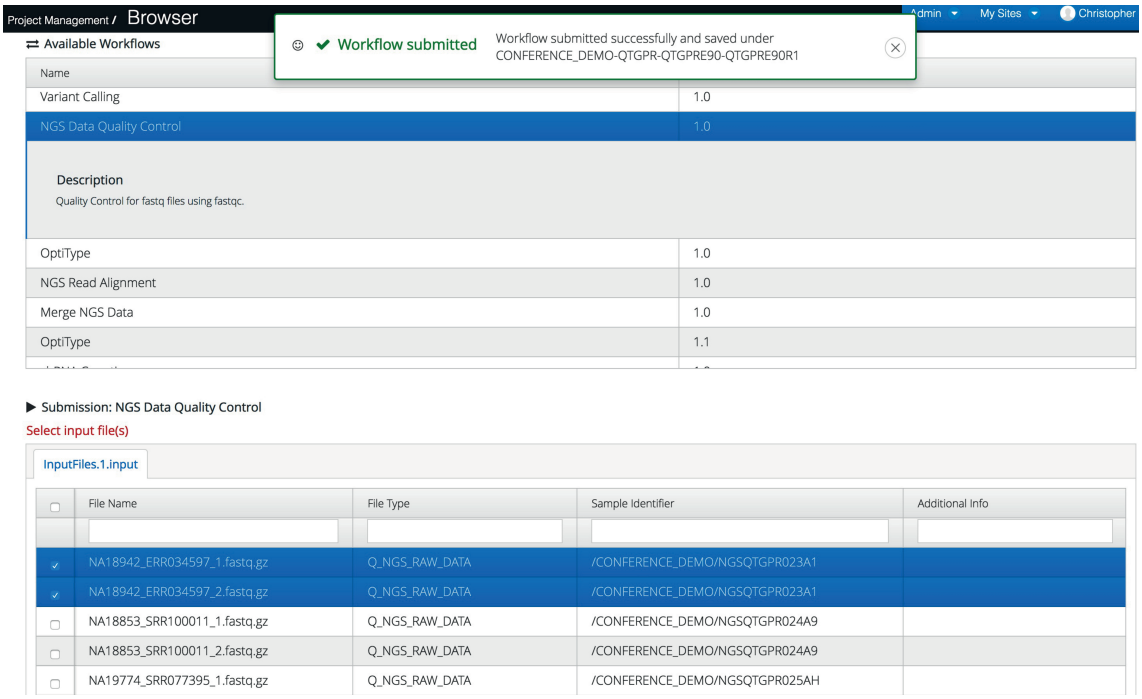

Figure 5.11: Workflow selection and submission. All available workflows for the corresponding project can be seen in the workflow tab. After selecting the workflow, users select input files and specify the parameter values. Users are notified about the submission status.

5.3.6 Resources for Personalized Vaccine Design

To facilitate the design of personalized (cancer) vaccines based on different types of omics data through a readily accessible system, we implemented functionality for data management, data analysis, as well as new user interfaces as part of qPortal. *iVacPortal* provides easy-to-use functionality and therefore enables users with different grades of computer literacy to identify potential peptide candidates for vaccines through a web-based system.

User Interface Adaptions

New user interfaces of iVacPortal include an adapted project overview page and a specialized status component. For all projects dealing with the personalized vaccine development, the project overview page contains the available HLA typing of patients and provided metadata such as the data of the initial diagnosis (Appendix Figure D.1). The status component visualizes the current overall progress, the single steps within the computational vaccine design pipeline, and offers direct links to available analysis workflows (Appendix Figure D.2).

Further, we implemented a new interface to add patients through qNavigator rapidly. The corresponding user interface can be accessed from every state in qNavigator (`Add Patient`).

The registration process is shown in Appendix Figure D.3. Underlying implementations include an openBIS ingestion service[xii]. The new project, experiment, and sample identifiers are created within qNavigator. Subsequently, the ingestion service is executed with the information entered by the user and the generated identifiers. Thereby, openBIS entities of the first four layers (from patient to NGS sequencing and optionally HLA typing as the fifth level), their connections, and the new project are registered. Additionally, provided metadata is set as property values. Users may provide information about the type of the sequenced tissues, secondary identifiers, detailed tissue information, and information about the applied sequencing technique and machine. After registration, corresponding sample identifiers can then be downloaded from qNavigator as spreadsheets and used for data registration.

Pipeline for the Design of Personalized Vaccines

The main component of the iVacPortal is the collection of computational analysis workflows which are accessible through the portal (Figure 5.12). Typically, collected patient samples originate from tumor tissue, non-malignant tissue, and blood. Data are generated by WES, RNA-Seq, and various proteomics and ligandomics measurement techniques. The core of the pipeline is the Python-based *Epitope Prediction and Annotation* (EPAA) workflow, which performs the epitope predictions and integrates results of other pipelines. These results come from different layers of the pipeline and different omics sources. The functionality of EPAA has been implemented using FRED2[26]. Annotated somatic variants and a set of HLA alleles are the mandatory input of EPAA. Somatic variants might have been provided directly or called by the somatic variant calling workflow using mapped reads. Optionally, annotated germline variants can be used as an additional input of the EPAA workflow. A suitable variant calling workflow has been implemented. Variants in VCF format can be annotated using corresponding workflows employing SnpEff[221] or ANNOVAR[220]. The mandatory HLA types can be determined using the implemented HLA typing workflow using OptiType (Chapter 3). Version 1.1 of the workflow includes an additional pre-processing step where reads are mapped against chromosome six of the human genome using the read mapper Yara[204]. Therefore, a pre-processing step, performed by the user to overcome memory issues, which might occur during the initial mapping step in the OptiType pipeline, becomes obsolete. Detailed descriptions of workflows are provided in Appendix F.

The EPAA workflow processes peptides and annotated variants in VCF format (`read_vcf`) or in a tab-separated file format with a set of mandatory columns, including chromosome, genomic position, and observed nucleotides. By parsing the variants, corresponding `FRED2.Core.Variant` objects are generated based on the provided information and annotated with further metadata such as the observed tumor depth or the tumor allele fre-

[xii] `https://github.com/qbicsoftware/etl-scripts/tree/master/reporting-plugins/register-ivac-lvl/`

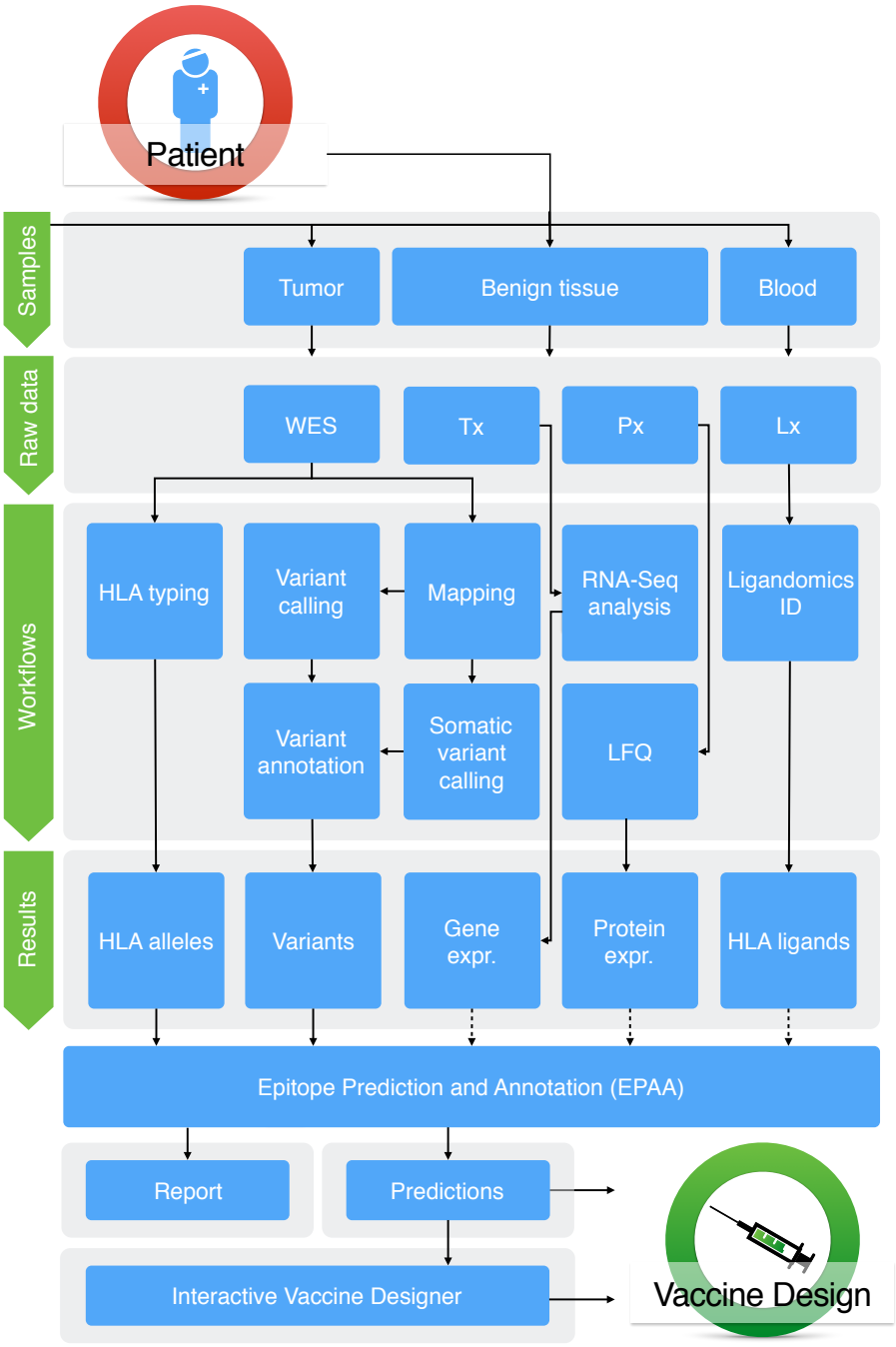

Figure 5.12: Overview of the implemented computational personalized vaccine design pipeline. The pipeline includes data processing of raw data, analysis workflows, and result generation. Raw data including WES or WGS, WTS (Tx), proteomics (Px), and ligandomics (Lx) data are used as input of available workflows for data processing. Typical tasks include read mapping and HLA typing. EPAA as the main workflow of the pipeline can be performed with additional workflow results like gene expression values (RNA-Seq Analysis) and the relative amount of proteins, using label-free quantification (LFQ). Subsequently (annotated) epitope prediction results can be used for manual selection or computational selection through the Interactive Vaccine Designer.

quency if available. Chromosome-grouped `Variant` objects are then integrated into the annotated transcripts using respective FRED2 functions. Transcript sequences will be fetched from BioMart[391] based on the user-defined (`--reference`) Ensembl[392] version using the FRED2 `MartsAdapter`. Resulting variant-containing transcript sequences are translated to respective protein sequences and sliced to peptides for a defined length interval. The generated lengths thereby depend on the user choice of HLA class I (8–11 AA) or class II (15–16 AA). If the corresponding parameter has been set (`--filter_self`), generated peptides are filtered to remove peptide sequences which exist in the human proteome and therefore are not suitable as vaccine candidates. In order to check for these peptides, a FASTA file containing protein sequences of the human proteome is parsed, and a corresponding FRED2 `UniProtAdapter.UniProtDB` object is generated. Users may also directly use peptide sequences provided in a tab-separated input file instead of annotated variants.

The remaining peptide sequences are then used as input for the epitope prediction using the FRED2 `EpitopePredictorFactory` functionality. Since the resulting `pandas.DataFrame` only contains prediction scores for every peptide and every provided HLA allele (if a corresponding model exists), it is extended by affinity values and classified into binder and non-binder. Further, peptides are annotated with information of the contained mutation (chromosome, genome position, gene, transcript, variant type) and the wild-type protein identifier. This information is fetched from BioMart using the transcript identifiers as query inputs. If specified, mutated peptides are annotated with the corresponding wild-type peptide sequence, which is generated based on the variant information.

As the choice of peptide-based vaccine candidates heavily depends on the detection of underlying mutations on multiple omics levels, the annotation with other workflow results has been implemented (Figure 5.12). In the case of results of the RNA-Seq analysis workflow, peptides are annotated with RPKM values calculated from gene feature counts for the corresponding gene. Alternatively, peptides may be annotated with results of the differential gene expression analysis workflow as $\log_2$-transformed fold changes. Results of the ligandomics identification workflows are used as intensity and score annotations if peptide sequences are present in the epitope prediction results. The identification of ligands may be performed against a customized database generated by the individualized proteome generator workflow. This workflow reads a set of annotated variants and produces the mutation-containing protein sequences (as in EPAA), which are then attached to a specified human reference proteome. Therefore, ligands can be searched against a personalized version of the proteome. Protein quantification values derived from the label-free quantification (LFQ) workflow are added as $\log_2$-transformed intensities of the peptide-originating proteins.

The `pandas.DataFrame` with all peptide sequences, predictions, and annotations is written out as a tab-separated file. Additionally, a report is generated containing general informa-

tion, such as the provided HLA alleles, used prediction methods, and statistics, including the number of provided variants and unique binding peptides.

Interactive Vaccine Designer

Since the number of HLA-binding peptides is usually significantly larger than the number of peptides used as vaccine components, the prediction of epitopes is commonly followed by the selection of peptides. The selection can thereby be done manually or by computational selection tools such as *NeoOptiTope* (Section 5.2.1). To make the computational selection available as part of our vaccine design pipeline, we implemented a GUI for NeoOptiTope. The user interface has been implemented as Vaadin 7 portlet[xiii] and was deployed as part of qPortal. NeoOptiTope has been extended to allow users to include and exclude specific peptides from the final solution explicitly. Additionally, we implemented functionality to handle given rank-based immunogenicity estimates. Concerning the execution of NeoOptiTope from the frontend, we implemented a Singularity[276] container, based on the available Docker[275] container[xiv], which is automatically started upon submission of the selection request.

The frontend is implemented as a Vaadin accordion component where tabs are arranged vertically. The vertical tabs guide users through the steps of the peptide selection process. In the beginning, users may choose between the upload of required files or the selection of data from the connected openBIS instance (Appendix Figure D.4A). The selection process requires an epitope prediction result file and corresponding HLA alleles with expression values. Since the epitope prediction result files differ between different versions of epitope prediction workflows, users have to select between two basic file structures (Appendix Figure D.4B). During the next steps of the data preparation stage, users may specify the column names of the epitope prediction results since these might differ from default values (Appendix Figure D.5). The provided HLA alleles and corresponding expression values as FPKM values can be specified manually (Appendix Figure D.6) or loaded from the openBIS instance. In the epitope pre-selection step, all peptide sequences from the used epitope predictions are displayed and can be explicitly included or excluded (Appendix Figure D.7). The parameter adjustment, as shown in Figure 5.13, enables users to adjust the parameters of NeoOptiTope. This option allows users to change parameter values quickly, rerun the selection process, and therefore interactively design vaccine solutions.

The result view (Figure 5.14) includes the parameter values, objective values, a table with the selected peptides contained in the generated solution, charts visualizing statistics, and properties of the solution. Multiple runs with different parameter settings will be organized in a horizontal tab-sheet component. All results can be downloaded and automatically registered in the database through an implemented ETL-based registration process. Therefore, the epitope

[xiii] `https://github.com/qbicsoftware/vaccine-designer-portlet`
[xiv] `https://hub.docker.com/r/aperim/epitopeselector/`

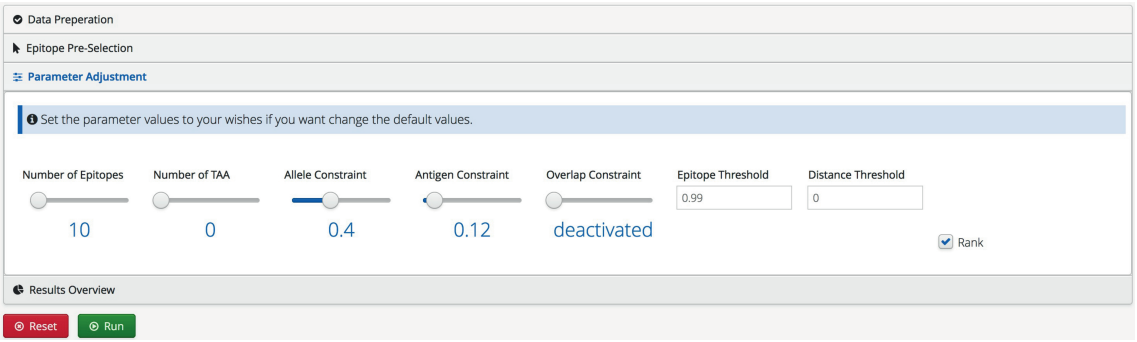

Figure 5.13: Parameter adjustment in the Interactive Vaccine Designer portlet. Users may adjust the number of epitopes, the number of TAAs, the allele constraint, antigen constraint, and overlap constraint. Epitope and distance thresholds can be specified. The rank parameter specifies if given immunogenicity estimates are rank-based.

selection results will be added as a data set to the chosen project, whereas the connected sample will be added as child instance of the corresponding HLA typing and epitope prediction workflow runs.

5.4 Results

To demonstrate the feature-rich implementation of qPortal, we conducted a case study based on publicly available data. As other web-based platforms exist in the area of data-intensive biomedical research, we performed a comparison with Galaxy instances and elaborated on the differences between its' features. We demonstrate qPortal's strength in supporting the entire project life cycle.

5.4.1 Case Study

As project management usually starts with the entry of biological entities under investigation, we registered samples of 20 individuals of the 1000 Genomes Project[378]. Based on the annotation provided by the 1000 Genomes Project, we selected a subset of male and female individuals of ten different populations (Appendix Table E.2). The new project was registered with the Project Wizard portlet. Afterward, metadata about the sex and population of the individuals, as well as the extracted blood and DNA samples, were registered. To demonstrate the features for automated registration of incoming data sets, we selected one WES run for 19 of the individuals and two runs for one individual. Automatically created identifiers in our system were mapped to the identifiers of the 1000 Genomes Project. Typically, these barcodes would be used in the lab at the time of data creation to identify processed samples uniquely. The data sets were processed by the dropbox system for raw genomic data and moved to our storage. Next, raw data and metadata were associated using the implemented ETL process.

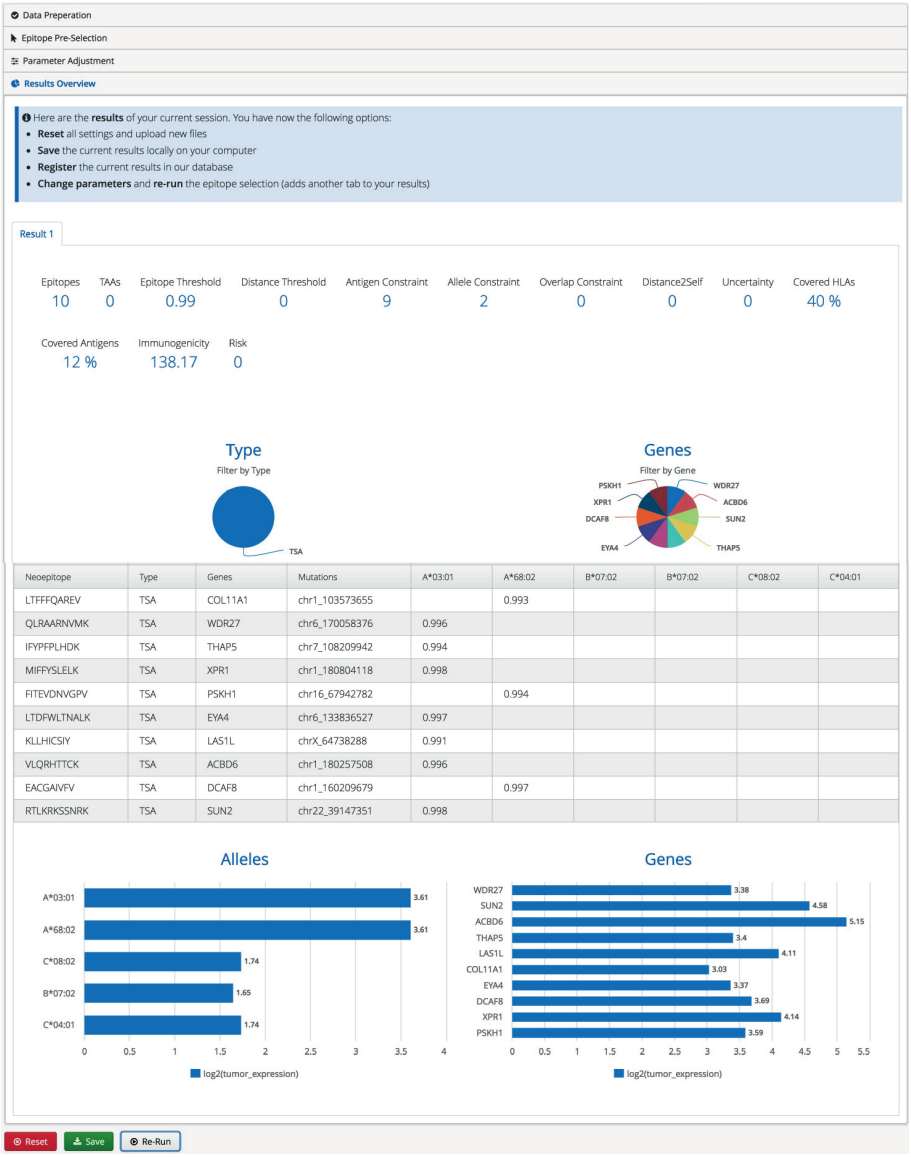

Figure 5.14: Result presentation in the Interactive Vaccine Designer portlet. Used parameter settings and objective values are shown. The type (TSA/TAA) distribution and the gene distribution are visualized as pie charts. Peptides of the final solution are presented in a table with their sequence, type, gene, mutation, and corresponding allele scores. The $\log_2$-transformed expression values of HLA alleles and genes are presented as horizontal bar plots. Multiple runs on the same input data are presented in a tab-sheet component.

Processing of technical replicates of runs above resulted in the registration of 21 NGS samples with meta information within the project. Further, we applied the quality control workflow using FastQC[184] on all samples through qNavigator. qNavigator enables direct visualization and download of the resulting quality control reports in HTML format. All workflow results can be directly visualized (Figure 5.15). Since one typically has to deal with way more complex

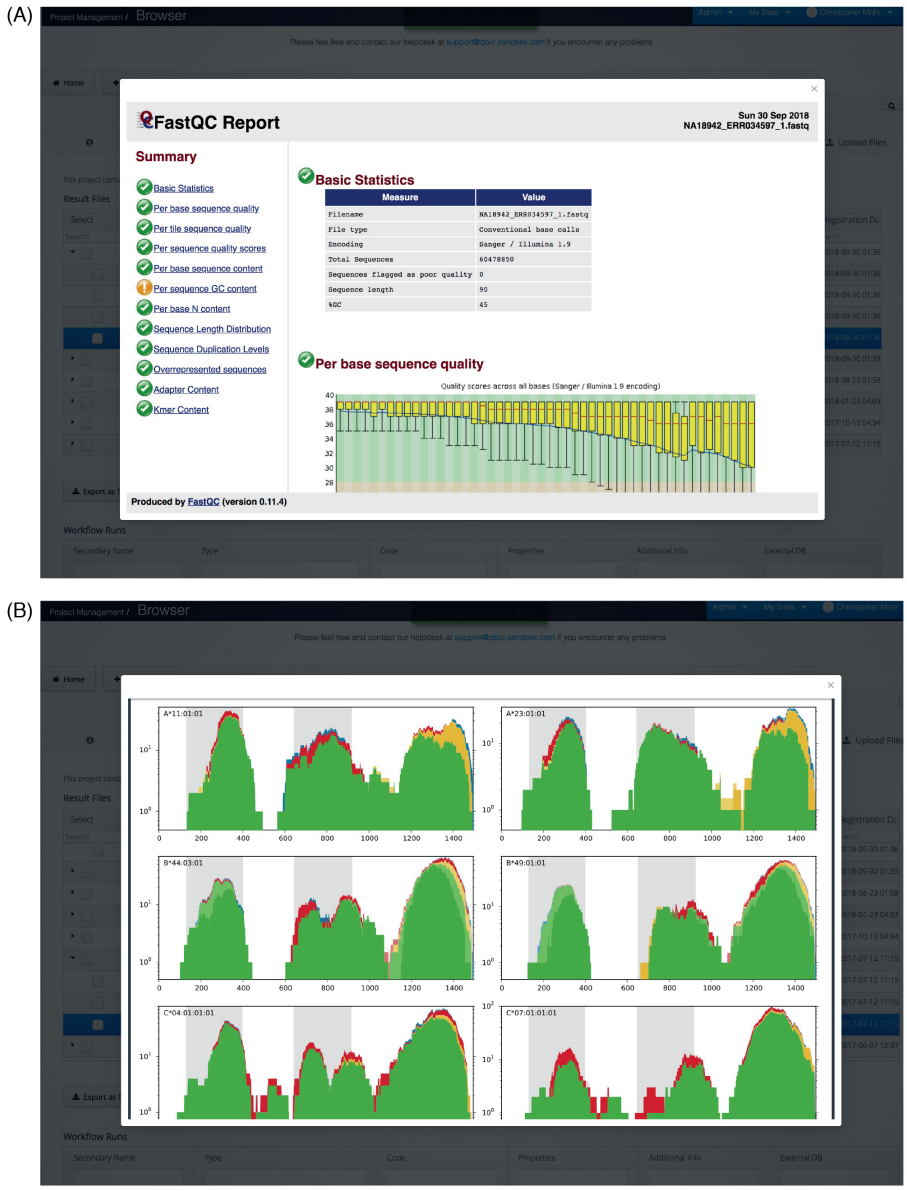

Figure 5.15: Workflow result presentation. Results of workflow runs will be directly visualized in a pop-up window when users select corresponding files: (A) HTML report of one FastQC run. (B) Coverage plot, stored in PDF format, of one OptiType run.

analysis tasks than quality control, we further performed NGS read alignment using BWA-MEM[393] and HLA typing using OptiType[301] (Chapter 3). Both analysis pipelines are available as workflows from qNavigator. Read mapping against the human reference genome (*GRCh37*; *hg19* Feb. 2009 assembly) was performed for two individuals. Resulting aligned reads were used as input for the OptiType workflow to determine the HLA type of these two patients. For the remaining 19 individuals, the OptiType workflow was directly applied on available FASTQ files.

5.4.2 Advantages of a Data-driven Research Portal

To further assess *qPortal's* usability, we draw a comparison with Galaxy instances (Table 5.1). Our implementation focuses on the specification and annotation of experimental steps in an early project phase in order to leverage this information throughout the whole project cycle and therefore facilitate data analysis. The added value of this data-driven approach is outlined below.

Table 5.1: Comparison between qPortal (data-driven) and Galaxy (workflow-driven) based on functionality criteria.

Criteria	qPortal	Galaxy
Project Management	qWizard, qNavigator	Not possible
Metadata Management	At every experiment level	Focus on workflow parameters
Experimental Design	Focus on experimental metadata	No similar functions
Data Import	Datamover upload	Upload, download link
Workflow System	Available	Available
Workflow Creator	Indirectly (WS-PGRADE)	Combination of workflows
Results	Visualization, download	Visualization, download
Data Security	Data transfer (ssh), permission based	Browser upload, user based

One main feature of qPortal, which is essential for the definition of the data-driven approach, is **project management**. This includes the functionality to register and store projects. Further, customized experimental designs can be used and maintained. The web applications qWizard and qNavigator provide the functionality to enable the annotation throughout the entire life cycle of experiments and projects, which is of prime importance. Experimental designs are built on the basis of experiment and sample instances, which ensure the automated registration of incoming data sets. In the case of the workflow-driven approach of Galaxy, implicit project management is used. This concept has a focus on file-based analysis and visualization.

Both systems offer means of **data annotation** with different emphasis. qPortal provides functionality for entering metadata, general project information, and experimental designs at the project registration stage. Furthermore, information that has been entered upon registration can be changed or extended through qNavigator. This information might include individual characteristics and an explanation of experimental steps. Two similar concepts are employed by Galaxy. Users may add notes and tags to items to make them searchable[264]. This concept focuses on the annotation of workflow runs, wheres qPortal stores metadata for each step of experiments. Extensive metadata collection throughout the whole project cycle,

even before experiments are performed, is essential for reproducibility and benchmarking of different workflows and parameter settings. Moreover, time and money can be saved by the evaluation of the statistical power of a study design before it's implementation. During a study, contingent mistakes in the design or sample handling can be traced back with higher confidence.

Since biomedical applications usually entail different data types coming from different sources, **data import** is an important criterium regarding usability. As described (Chapter 5.3.3), data import to qPortal is possible using the Datamover. We implemented automated data registration procedures based on ETL processes. These processes enable the experimental design and incorporated barcode generation, and include ID mapping and file format recognition functionality. ETL scripts can be file-type or lab-specific and include the registration of additional metadata annotations. In addition to that, small unstructured data can be uploaded through qNavigator. Similarly, Galaxy offers direct data upload through the web browser. The implemented rules and governance depend thereby on the Galaxy instance provider. Upload of larger files is possible through direct transfer from provided URLs and via encrypted FTP. Concerning auto-detection of file types, Galaxy offers similar features as qPortal, whereas more complex file operations on input data have to be implemented as workflow nodes.

The connected workflow management system gives qPortal users direct access to **workflows**. Used data sets can be directly selected within the projects and parameters can be easily adjusted in the GUI. The same holds true for Galaxy, whereas details depend on the implementation of the respective workflows. Due to our data-driven approach, the selection list of workflows is limited by the availability of required input data. Currently, qPortal does not include a feature for web-based workflow creation. However, custom workflows can be created in WS-PGRADE and ported. The most important difference results from the registration of experimental design information and the collection of metadata. Workflows can make use of this information and visualize results according to study variable values.

After the submission of workflows, users can monitor them and directly navigate to the **results** which are automatically registered by ETL processes in the corresponding project. As in Galaxy, all workflow results can be either downloaded or directly visualized in the portal.

Data security, access, and confidentiality are of prime importance, especially if biomedical data is included. This kind of data is usually bound to strictly regulated terms. The access to data from qPortal is regulated through the rule-based permission scheme of openBIS. Assigned project roles ensure that users will only be able to see their projects and corresponding data. The Datamover implementation guarantees data transfer security using Secure Shell connections. Galaxy uses a workspace concept, where users can upload data to it and optionally share it with others.

5.5 Discussion

Modern research projects commonly entail distributed data generation and many stakeholders, coordinated within larger consortia. Hence, means of efficient data sharing, central project management, and remote communication are getting more and more important. Also, fast, reliable, and easily accessible data processing and analysis tools are needed. With the continuously growing number and throughput of omics technologies, as well as an increasing number of multi-omics-based biomedical projects, the need for full automation in these tasks and data management is obvious. Web-based portals, in combination with an efficient backend system and a connection to distributed computing infrastructures, can provide such solutions. There have been big efforts towards web-based platforms in biomedical research and other domains. However, existing solutions are often limited to domain-specific applications. One established approach in the field of biomedical research that facilitates data analysis for scientists without expertise in programming is Galaxy. The *workflow-driven* approach of Galaxy eliminates the need for users to install programs or use the command line. Frequently, Galaxy instances additionally provide free storage and compute resources and offer easy-to-use means of sharing analysis results. Due to its improvements and large community, the integration level for many tools and different infrastructures in Galaxy is well established.

We implemented *qPortal*, a web-based portal that provides similar features as Galaxy concerning means of workflow submission. qPortal's workflow-based analysis module enables users to execute bioinformatics pipelines on powerful compute resources through intuitive interfaces. By hiding the complexity of distributed computing infrastructures, data analysis becomes feasible for scientists without prior scripting or command line experience. Currently, qPortal provides an interface to gUSE/WS-PGRADE. However, the interface to workflow systems is generic by design and therefore allows for the extension with other workflow systems, such as Snakemake[273] and Nextflow[277]. This enables users to create Snakemake or Nextflow workflow instances and use them through qPortal, given that respective tools are available on the cluster. Extensions of the workflow interface still can make use of existing implementations for data staging which have to be incorporated into the workflow itself.

To not only guarantee flexibility with respect to workflow interfaces, we developed qPortal using the web-framework Vaadin, which is based on the Google Web Toolkit providing a wide range of browser support. Because of the active developer community, useful tools and add-ons are frequently provided. The same holds true for the free and open source enterprise portal solution Liferay[284]. Besides, Liferay implements the Java Portlet Specification (Java Specification Request (JSR) 168[394]), standardizing the interaction between portlets and portlet containers, as well as ensuring compatibility across portal products. Consequently, given JSR 168 compliance, qPortal components can run on other portal systems such as JBoss Portal[395] with manageable effort.

The main difference of our *data-driven* approach to existing solutions, such as the workflow-driven approach of Galaxy, is its focus. qPortal provides features for experimental design, metadata handling, project management, and collaboration. Galaxy's focus on workflow annotation and collection of parameter settings aims primarily at reproducibility of computational analyses. In recent years, numerous studies support the importance of this approach to solve the reproducibility issue[262,396,397]. With qPortal, we take the notion of reproducibility one step further. Our comprehensive data-driven approach includes the extensive collection of metadata before experiments are conducted. As a consequence, time and money can be saved since the study design allows for the estimation of statistical power before experiments are performed. Further, shortcomings in study design and sample handling can be traced back more easily and with higher confidence. The comprehensive annotation of experimental data moreover increases the likelihood that this data is reused in future research. As noted by Leek and Peng[398], even computationally reproducible results have to be used with caution if no sound experimental design exists. The confidence in a scientific hypothesis is highly dependent on the replicability of studies, including data generation, given the same experimental setup. Main reasons that hinder the replicability include missing statistical considerations, missing protocol information, and poor experimental design[399,400]. Therefore, qPortal facilitates data annotation that is equally important for reproducible research, in addition to thorough logging of processing, parameters, and pipelines. Hence, qPortal follows the *Findable, Accessible, Interoperable and Reusable* (FAIR) Guiding Principles for scientific data management[401].

To summarize, we developed qPortal, a platform that enables data management and empowers scientists to analyze and navigate through large-scale biological data integratively. Its web-based nature facilitates the implementation of qPortal as a central platform in modern data-driven biomedical research. The scalable nature of the setup allows the integration of large in-house and public data sets and thereby builds an ideal ecosystem for big data analysis in biomedicine. Implemented data models and ETL processes build the backbone of the integrated data management system, enabling fully automated registration of experiments, data, and metadata. Intuitive graphical user interfaces provide the functionality to register factor-based experimental designs (qWizard). qNavigator enables users to perform project management and to run analysis pipelines. Because of its generic design and the flexible open-source components, qPortal can easily be adapted and extended as shown for iVacPortlal, a web-based portal for the management and analysis in personalized vaccine studies. Other solutions, such as pVACtools[402] provide similar functionality for identification and prioritization of neoantigens but lack metadata collection options and do not provide user interfaces. iVacPortal is implemented as part of qPortal, making use of its modularity on multiple layers. On the portal layer, modularity allows for the extension by portlets, as in the case of the Interactive Vaccine Designer. Modularity on the portlet layer made it possible to implement additional

components and features in qNavigator. Further, we extended the workflow functionality by implementing data processing and analysis pipelines commonly used within the personalized vaccine design pipeline. Implemented data transfer mechanisms, ETL processes, and data model extensions complete the integration of this pipeline, enabling researchers to generate lists of possible vaccine candidates through an easy-to-use frontend.

Chapter 6

Assessment of Personalized Vaccine Options through iVacPortal

Parts of this chapter were published in:

Multi-omics discovery of exome-derived neoantigens in hepatocellular carcinoma

Löffler, M. W.*, Mohr, C.*, Bichmann, L., Freudenmann, L. K., Walzer, M., Schroeder, C. M., Trautwein, N., Hilke, F. J., Zinser, R., Mühlenbruch, L., Kowalewski, D. J., Schuster, H., Sturm, M., Matthes, J., Riess, O., Czemmel, S., Nahnsen, S., Königsrainer, I., Thiel, K., Nadalin, S., Beckert, S., Bösmüller, H., Fend, F., Velic, A., Maček, B., Haen, S. P., Buonaguro, L., Kohlbacher, O., Stevanović, S., Königsrainer, A., HEPAVAC Consortium, and Rammensee, H.-G.

Genome medicine, 11(1), 1-16 (2019)

* Joint first authors

6.1 Introduction

Over the last decades, the discovery that tumors in cancer patients can elicit a host immune response, along with multiple other findings, has driven the progression in cancer immunotherapy[403]. To date, many approaches have been developed to exploit the immune system's ability to target cancer cells (Section 2.3.3). For an increasing number of cancer entities, immunotherapy has become a therapeutic option and is implemented in the clinics. Emerging clinical evidence for its effectiveness was mainly based on the treatment with immune checkpoint inhibitors and their demonstrated potency as reported for metastatic melanoma[404,405]. For the majority of malignancies, immune checkpoint inhibitors, and other immunotherapeutic

options, such as cancer vaccines, results remained mostly disappointing. One reason for the limited therapeutic success of cancer vaccines might be the use of TAAs, such as aberrantly expressed antigens that are often shared across cancers, and their limited anti-tumor potential due to central T-cell tolerance[15]. As previously described, non-synonymous somatic mutations give rise to tumor-specific neoantigens[406,407]. Due to their tumor-specificity accompanied by missing expression elsewhere in the body, these antigens do not possess the potential to cause toxicity. Additionally, the corresponding T cell pool is not affected by central tolerance, increasing the potential immunogenicity of these TSAs. As postulated, TSAs are of particular relevance for the therapeutic efficacy of cancer immunotherapies and have been shown to elicit tumor-specific T-cell reactivity[148,408]. Thus, corresponding neoepitopes, as targets of tumor-specific T cells, also make strong candidates for therapeutic cancer vaccines (Section 2.4), overcoming the limited efficacy of shared tumor antigens in vaccination. Earlier studies demonstrated the induction of anti-tumor immunity against a neoantigen based on a mutation that is present in a subset of gliomas[409]. However, as described[406,407,410] tumor cells possess a vast genetic heterogeneity, which is not only present between individuals with the same cancer type but even within individual tumors. Considering the diversity of HLA molecules across individuals and the associated restriction with respect to the pool of presented peptides suggest the design of therapeutic cancer vaccines in a personalized fashion. Furthermore, clonal evolution and immune escape mechanisms require cancer vaccines to target multiple neoantigens[129]. Due to the developments in genomic sequencing, the rapid identification of individual tumor mutations, i.e., the mutanome, became feasible. Further, more recent advances in computational approaches allow the identification of private neoepitopes from the mutational spectrum of a tumor, derived from cancer exome data as shown for mouse models[141,147] and single patients[148]. The prediction of T-cell neoepitopes is usually based on HLA-binding affinities predicted by established *in silico* methods[238,245,248]. Therefore, it is possible to identify candidate tumor neoantigens on a per-patient basis. Since HLA binding is a prerequisite of T-cell reactivity and HLA binding algorithms are reliable, predicted binding affinity serves as a reasonable estimate. Still, the confirmation of their presence on tumor tissue and efficacy of predicted neoepitopes remains challenging.

Related Work

Recently, studies have shown the feasibility of neoepitope identification and the analysis of their actual presence on tumor tissue within the framework of personalized cancer-specific vaccines for mouse[143] and murine tumor models using MS for confirmation[411]. Kalaora et al. showed the combination of WES analysis and *in silico* prediction methods with MS-based HLA ligandomics to identify neoantigens in one melanoma patient[412]. According to their findings, two mutated peptides could be confirmed, with one eliciting a T-cell response. In 2015, Rosenberg

et al. published the results of a next-generation sequencing approach combined with high-throughput immunological screening to demonstrate the recognition of neoepitopes by T cells in ten patients with metastatic gastrointestinal cancers[413]. Bassani-Sternberg et al. reported the identification of neoepitopes by MS[414]. They performed exome sequencing of 25 tumors from melanoma patients, following somatic SNV calling – furthermore, Bassani-Sternberg et al. performed MS analysis of purified HLA-binding peptides from 25 melanoma patients. For a subset of these patients, they compared the results with *in silico* derived HLA-binding affinities of respective neoepitopes. The generation of vaccines targeting private neoantigens has been shown by Ott et al.[150] They performed WES of matched tumor and normal DNA for ten melanoma patients, followed by somatic variant calling, prediction of neoepitopes, and the assessment of RNA expression. Further, they monitored the expansion of neoantigen-specific T cell repertoires. Sahin et al.[149] reported an RNA-based poly-neoepitope approach and its application in melanoma patients. Their approach combines comparative exome and RNA sequencing of tumor tissue and blood cells. The selection of neoepitopes is further based on binding affinity to HLA class II molecules and expression of mutation-encoding RNA. In 2017, Chang et al. presented an analytical workflow for the identification of neoepitopes using WES data of pediatric cancers[415]. Additionally, they incorporated transcriptome sequencing data to analyze expression levels of potential neoepitopes. Lately, Rubinsteyn et al. published the computational pipeline for the PGV-001 neoantigen vaccine trial[416]. Their computational pipeline combines somatic variant calling based on WES of matched tumor and normal samples, HLA typing, and epitope prediction. Peptides are then selected based on HLA binding and tumor-derived RNA expression. Most of the published studies on *in silico*-based identification of neoepitopes do not include multiple omics layers to confirm predicted neoepitopes. Additionally, a central solution to perform bioinformatics analysis for the development of personalized vaccines does not exist.

Project Outline

We performed an in-depth analysis of WES and WTS in combination with proteomics and HLA ligandome data. Based on this multi-omics data set, we assessed potential patient-specific multi-peptide vaccine candidates while investigating the evidence for the presentation of mutated naturally presented HLA ligands. To this end, we applied *in silico* approaches for the development of personalized vaccines (Chapter 5.3.6), including HLA typing (Chapter 3), through one central web-based interface (Chapter 5). Thereby, we made use of *qPortal's* data management and workflow-based analysis resources. The corresponding study design is shown in Figure 6.1. The composition of our two investigated cohorts, including *hepatocellular carcinoma* (HCC) patients and *acute lymphoblastic leukemia* (ALL) patients, renders it highly relevant in today's cancer research. HCC is one of the most common malignant tumors and among the

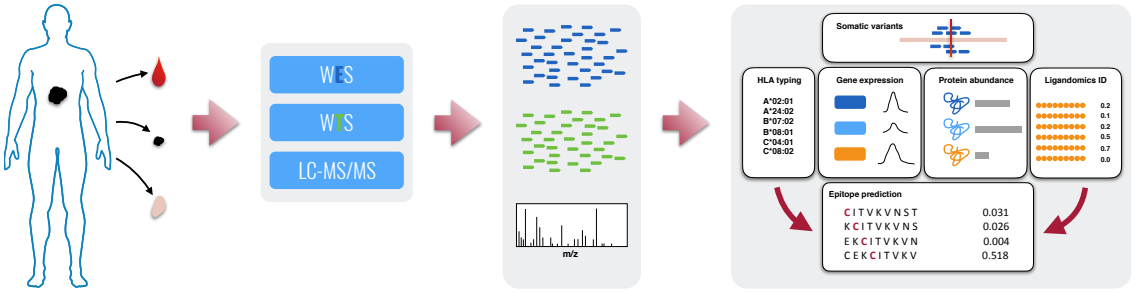

Figure 6.1: Outline of the conducted study. Malignant, non-malignant, and blood samples were extracted from patients. WES, WTS, and liquid chromatography coupled with tandem mass spectrometry (LC-MS/MS) data were analyzed using established workflows for read alignment, variant calling, RNA-Seq analysis, label-free quantification, and ligandomics identification. Resulting somatic variants and HLA genotypes were used to perform epitope predictions using EPAA. Epitope predictions were further annotated with results of gene expression, protein abundance, and ligandomics identification analysis if available. Based on the resulting integrated data set, we investigated the existence of potential neoepitopes on multiple omics layers. Human body silhouette icon obtained from Reactome Icon Library[88] and adapted.

most common cancer-related death causes[417]. Furthermore, ALL is the most common malignancy affecting children caused by genetic alterations. Although effective treatments, like chemotherapy and stem cell transplantation, exist, patients who relapse have a poor prognosis. Additionally, further chemotherapeutic treatments are often limited by toxicity. Therefore, the use of patient-specific immunogenic peptides in a multi-peptide vaccination approach might represent a viable alternative. Our analysis is based on tumor-specific somatic variants derived from sequencing of tumor and non-malignant tissue samples. Using our implemented pipeline EPAA, we generate mutation-derived peptides based on these mutations and predict potential neoepitopes for determined HLA genotypes leveraging state-of-the-art *in silico* prediction methods. Further, we annotate predicted binding neoepitopes with gene expression, protein quantification and HLA ligandomics identification data. By analyzing the obtained results, we computed statistics about potential neoepitopes and their presence on available omics layers across all patients of the investigated cohorts.

6.2 Materials and Methods

This section provides an overview of the two cohorts that were included in this study and details of the experimental data generation steps. Further, details on the employed analysis pipelines, such as parameter settings, are given. Data management, including project registration, data transfer, and collection of metadata was conducted through *iVacPortal* (Chapter 5). The analysis was performed using workflows as available in iVacPortal unless otherwise stated.

6.2.1 Experimental Data

We analyzed data from 24 patients with diagnosed ALL and 16 patients suffering from HCC. The complete list of patients with IDs, diagnosis, and HLA typing is given in Appendix Table E.3. The experimental data for these 40 patients have been generated using different protocols, which will be described here. An overview of the available experimentally derived transcriptomics, proteomics, and ligandomics data is shown in Table 6.1.

Table 6.1: Data availability on different omics' levels for the *ALL* and *HCC* cohort. WES data for non-malignant (nm) and tumor (tu) samples was available (denoted by x) for all patients. Transcriptomics (Tx), proteomics (Px), and ligandomics (Lx) data have been analyzed for all patients enclosed in the HCC cohort if available. Shotgun proteome (n=7) and HLA ligandome (n=16) data were available for tumor and non-malignant tissue in the HCC cohort. In the case of the ALL cohort, only malignant samples were available for RNA-Seq.

ALL					*HCC*				
ID	**Tx**		**Px**	**Lx**	**ID**	**Tx**		**Px**	**Lx**
	nm	tu				nm	tu		
QA001	-	x	-	-	HCC023	x	x	x	x
QA002	-	x	-	-	HCC024	x	x	x	x
QA003	-	x	-	-	HCC025	x	x	x	x
QA004	-	x	-	-	HCC026	x	x	x	x
QA005	-	x	-	-	HCC027	x	x	x	x
QA006	-	x	-	-	HCC028	x	x	-	x
QA007	-	x	-	-	HCC030	x	x	-	x
QA008	-	x	-	-	HCC034	x	x	x	x
QA009	-	x	-	-	HCC035	x	x	-	x
QA010	-	x	-	-	HCC036	-	x	x	x
QA011	-	x	-	-	HCC038	x	x	-	x
QA012	-	x	-	-	HCC040	x	x	-	x
QA013	-	x	-	-	HCC041	x	x	-	x
QA014	-	x	-	-	HCC042	x	x	-	x
QA015	-	x	-	-	HCC043	x	x	-	x
QA016	-	x	-	-	HCC045	x	x	-	x
QA017	-	-	-	-					
QA018	-	x	-	-					
QA019	-	x	-	-					
QA020	-	x	-	-					
QD003	-	x	-	-					
QD004	-	x	-	-					
QD005	-	x	-	-					
QD007	-	x	-	-					

6.2.2 Ethics and Clinical Specimens

This research was conducted in accordance with the Declaration of Helsinki and approved by the local institutional review board of the University Hospital Tübingen (Tübingen, Germany). All participants provided written informed consent before study inclusion. The ALL study was approved by the local ethical board under registration number 233/2010/BO1. Studies, including the HCC patients, were approved by the local ethical board under the registration numbers 364/2014/BO2 and 222/2015/BO2. For the HCC cohort, samples were obtained from malignant liver tissue, non-malignant liver tissue, and peripheral blood. In the case of ALL, expanded fibroblasts serve as reference tissue to avoid the usage of blood samples which might be contaminated with leukemic cells.

Experimental HLA Typing

Experimental HLA typing was performed by single specific primer-polymerase chain reaction at the Department of Transfusion Medicine (Tübingen, Germany), following clinical routines.

Next-Generation Sequencing

WES and WTS for QA001–QA020 (no WTS for QA017), QD003–QD005, QD007, HCC023–HCC027, HCC034, and HCC036 were conducted. Sample preparation for WES was performed using the SeqCap EZ v2 (Roche, Pleasanton, USA) or the SureSelectXT Human All Exon v5 kit (Agilent, Waldbronn, Germany) and the TruSeq Stranded mRNA kit (Illumina, Eindhoven, Netherlands) for WTS, respectively. Samples were sequenced on the HiSeq 2500 System (Illumina, Eindhoven, Netherlands) in paired-end mode. For HCC028, HCC030, HCC035, and HCC038 to HCC045, WES and WTS were performed by CeGaT GmbH (Tübingen, Germany). DNA and RNA were extracted from fresh frozen tissue and peripheral blood mononuclear cells (PBMCs) using the AllPrep DNA/RNA Kit (Qiagen). DNA libraries were prepared with the SureSelectXT Human All Exon v6 kit (Agilent, Waldbronn, Germany). Sequencing was performed on a HiSeq 4000 System (Illumina, Eindhoven, Netherlands) in paired-end mode, yielding 2×100 bp reads. In the case of RNA, library preparation was performed with the SMARTer Stranded Total RNA-Seq Kit v2 - Pico Input Mammalian (Clontech, Saint-Germain-en-Laye, France) and sequenced with the HiSeq 4000 System (Illumina, Eindhoven, Netherlands) in paired-end mode with a read length of 100 bp.

Analysis of HLA Ligands by Liquid Chromatography Tandem Mass Spectrometry

Isolation of HLA class I ligands from HCC and corresponding non-malignant liver tissue was performed by the Department of Immunology (Tübingen, Germany) as previously described[418]. Immunoaffinity purification was performed based on the pan-HLA class I-specific monoclonal

antibody W6/32[419]. Elution was done using 0.2% trifluoroacetic acid, separated by nanoflow uHPLC (UltiMate 3000 RSLCnano System, ThermoFisher) using a 50 μm × 25 cm column (PepMap RSLC, Thermo Fisher). A linear gradient ranging from 3 to 40% acetonitrile over the course of 90 minutes was used. An online coupled Linear Trap Quadropole (LTQ) Orbitrap XL mass spectrometer (Thermo Fisher) in automated data-dependent acquisition mode was used for measuring eluting peptides. The five most abundant precursor ions (top5) for collision-induced dissociation fragmentation were selected. Samples were analyzed using up to five technical replicates.

Shotgun Protein Tandem Mass Spectrometry

Mass spectrometry experiments were conducted by the *Proteome Center Tübingen* (Tübingen, Germany). Sodium dodecyl sulfate-polyacrylamide gel electrophoresis was used for the purification of eluted protein samples. Coomassie-stained gel pieces were digested using trypsin. Afterwards, LC-MS/MS analysis was performed using an EasyLC nano-HPLC (Proxeon Biosystems, Roskilde, Denmark) coupled to an LTQ Orbitrap Elite (ThermoFisher). The LTQ Orbitrap Elite was operated in positive ion mode. Fragmented masses were excluded for 60 seconds after MS/MS.

6.2.3 Computational Analysis

Computational analysis of all samples was performed through *qPortal* if not stated otherwise. Detailed descriptions of all applied workflows are given in Appendix F. For raw WES and RNA-Seq data, files of different lanes were merged using the corresponding workflow.

HLA Typing

HLA class I alleles with four-digit resolution were predicted with the OptiType workflow 1.1 with default settings ($\beta = 0.009$) using WES data of blood samples. Appendix Table E.3 shows the resulting HLA genotypes for all patients.

Variant Calling

Somatic variant calling for patients QA001–QA020, QD003–QD005, QD007, HCC023–HCC027, HCC034, and HCC036 was performed by the Department of Medical Genetics and Applied Genomics (Tübingen, Germany). Bioinformatic data analysis was done using the megSAP pipeline (`https://github.com/imgag/megSAP`) in combination with the ngs-bits package (`https://github.com/imgag/ngs-bits`). Briefly, the pipeline included adapter trimming using SeqPurge[187], DNA-read mapping to the Genome Reference Consortium Human Build 37 (*GRCh37*) using BWA-mem[196], and duplicate annotation by Samblaster[420]. Further trimming

of overlapping reads was done with an in-house tool to reduce false-positive variants having very low allele-frequencies. Somatic variants were called using Strelka[20]. Resulting variants were annotated with SnpEff/SnpSift[221,421], vcflib (`https://github.com/ekg/vcflib`), and dbNFSP[422]. Further, custom filter criteria were applied to obtain high-confidence variants. STAR[208] was used to map the trimmed RNA reads. DNA variants were annotated with in-house variant frequencies and (tumor) RNA depth and allele frequencies. In the case of HCC028, HCC030, HCC035, and HCC038 to HCC045, CeGaT GmbH (Tübingen, Germany) performed demultiplexing of sequenced reads with Illumina bcl2fastq 2.19 and adapter trimming with Skewer[423] 0.2.2. Read mapping was performed with an in-house version of BWA-mem[196] 0.72 against an in-house version of *hg19* and ABRA[424] for local realignment of reads in target regions. Duplicate reads were discarded using samtools[209] 0.1.18. Variant detection was performed with proprietary software. Somatic variants were filtered for minimal coverage of 30 for the corresponding variant position in tumor and matched normal tissue and an allele frequency greater than 0.05 in the tumor tissue with a three times smaller allele frequency in normal tissue. In general, somatic variants were determined on the basis of tumor and matched blood samples.

Variant Annotation

For non-annotated somatic variants (HCC028, HCC030, HCC035, and HCC038 to HCC045), we applied the variant annotation workflow 2.0. The reference genome *GRCh37.75* was used for the annotation with SnpEff[221].

Tumor Mutational Burden

The tumor mutational burden (TMB) was calculated based on coding somatic variants, including synonymous and non-synonymous variants, as previously suggested[425]. To calculate the number of somatic mutations per megabase of genome examined, the number of coding variants was divided by the number of megabases covered by the applied exon enrichment kit.

Gene Expression Analysis

Feature counts were calculated with the RNA-Seq Analysis workflow 1.1. Mapping of RNA reads was done using TopHat2[207], after removal of adapter sequences with CutAdapt[185] (`--discard-trimmed`) based on FastQC[184] results. Counts for the mapped RNA reads were calculated using HTSeq[227]. The sort mode of given alignments was set to alignment position (`--order pos`). For the identification of counts on gene basis, the attribute parameter was specified (`--gff_attribute gene_id`). The option for strand-specific assays was disabled (`--stranded no`) and only exons were considered as feature type (`--feature_type exon`). Handling of reads that overlap more than one feature was set as well (`--overlap_mode`

`union`). As a reference, a *hg19*-derived bowtie2 index has been used to directly map resulting counts for given gene symbols to variant-annotated genes in later analysis steps. We further applied differential expression analysis for corresponding RNA-Seq analysis results of non-malignant and tumor samples for the HCC cohort. The differential expression analysis workflow was used with default parameters.

Protein Quantification

Label-free protein quantification was performed using version 1.0 of the MaxQuant[426] workflow. For non-malignant- and tumor-derived raw files, parameter groups were created, respectively. The fraction of each group was set to one. Within the group-specific parameters, the label-free quantification option was enabled. The type was set to *standard* and the multiplicity to one (*label-free quantification*). *Acetyl (Protein N-term)* and *Oxidation (M)* were selected as variable modifications. The digestion mode was set to specific and *TrypsinP* was selected as the enzyme. *MaxMissedCleavages* was set to 2 and the match type was specified as *MatchFromAndTo*. As global parameter settings, we specified the human version of the Swiss-Prot reviewed UniProt proteome as reference (version UP000005640, derived: 16/02/2016). *Carbamidomethyl (C)* was chosen as fixed modification and the options *Requantifiy* and *MatchBetweenRuns* were enabled.

HLA Ligandome Analysis

The analysis of HLA ligandome data was performed with the ligandomics identification workflow 2.1. In the case of available technical replicates, the workflow with co-processing functionality of replicates was used. In both workflows, identification and post-scoring are performed using the OpenMS[427] 2.3 adapters to Comet[428] 2016.01 rev. 3 and Percolator[429,430] 3.1.1 to identify peptides. Identical parameter settings were used for both workflows and all of the samples. The precursor mass tolerance was set to 5ppm, and the precursor charge was fixed to 2:3. Neutral losses were included for the PSM. Filtering was done with a false discovery rate (FDR) of 5%. FDR calculation was based on merged identifications of all available replicates using Percolator. Identifications of replicates were treated as internal IDs, and the median intensity of consensus features was used as the final quantification value. Since, the input data was not centroided, we specified the corresponding parameter (`--centroided false`) to centroid on MS1 (`--ms_levels 1`). We further selected a digest mass range of 800-2500, a fragment bin tolerance of 1 Da, and a fragment bin offset of 0.4 Da. As protein reference, we used the personalized protein sequence databases generated by the Individualized Proteome Generator workflow (Appendix F). The resulting database contains all protein sequences of the Swiss-Prot reviewed UniProt proteome (*UP000005640*, derived: 02/16/2016) and all protein isoforms with altered amino acid sequence due to mutations. The latter is generated by the integration

of somatic variants in the corresponding transcript sequences and subsequent translation using *FRED2* functionality. The resulting FASTA files are then used as reference database respectively. The database search was performed without enzymatic restriction. Oxidation of methionine residues was set as modification (maximal number of modifications: 3).

Epitope Prediction

The epitope prediction was performed using the EPAA 1.0 and EPAA 1.1 workflow for all ALL and HCC cases respectively. Both workflows were configured to predict epitopes for MHC class I (`--mhcclass I`) for the given somatic mutations and HLA alleles. For the retrieval of transcript information via BioMart, the stable database version based on *GRCh37* (`http://feb2014.archive.ensembl.org`) was set as reference (*--reference GRCh37*). From derived mutation carrying protein sequences, all peptides with a length of 8–11 AA were generated. These peptide sequences were filtered against a human proteome database (`-filter_self`), including the reviewed Swiss-Prot proteome (UP000005640 , 02/29/16) and the Ensembl proteome reference (release 84, 04/27/2016). Remaining peptide sequences were predicted for HLA binding using SYFPEITHI[238], NetMHC[244,245] 4.0, and NetMHCpan[247] 3.0. HLA-binding affinities are computed as half-max scores for SYFPEITHI, i.e., the percentage share of the given score of the maximum possible value for a given allele and peptide length. For NetMHC and NetMHCpan, affinities are calculated as $50,000^{(1.0-s)}$ for a given score s. If available, gene expression analysis results were used for the annotation of resulting peptides, gene expression values were annotated as FPKM values, calculated as follows for gene g:

$$f(g) = \frac{10^9 \times C}{NXL},$$

where L is the length of exons in base pairs for the corresponding gene g, C is the number of reads that mapped to gene g, and N is the total number of unique mapped reads in the sample. In addition, in the case of EPAA 1.1, additional annotation options are given, which were used if ligandomics identification results and protein quantification results were available. The flag `--wild_type` was used to activate the generation of wild-type peptide sequences for the mutation-carrying peptides.

Utilized Databases

Mutation numbers of TCGA HCC (TCGA-LIHC) and melanoma cases (TCGA-SKCM) were retrieved from Genomics Data Commons Data Portal (`https://portal.gdc.cancer.gov/`, accessed 2018-09-14). Variants were filtered for missense, frameshift, inframe deletion, inframe insertion, and coding sequence variants. Variants that were called by Mutect2 were

considered. Additionally, we screened our HCC HLA class I ligandome and proteome data set against CT antigens derived from CTDatabase[431] (http://www.cta.lncc.br, 20/02/2018).

6.3 Results

Downstream analysis of the workflow results was performed with Python[432]. Peptides, assessed with an SYFPEITHI score exceeding half-max, or in the case of NetMHC and NetMHCpan with a predicted affinity less or equal to 500 nM were considered epitopes for the given HLA allele. We excluded one patient with T-ALL (QA017) from the downstream analysis since no WTS data was available. If not stated otherwise, we refer to a potential neoepitope when we use the term neoepitope since neoepitopes would have to be ultimately experimentally validated for their presentation to and recognition by T cells. Besides, we will use the following notations:

- ***Var***: Somatic variant (SNVs, InDels, frameshift variant)
- $\boldsymbol{Var^{ns}}$: Non-synonymous somatic variant
- $\boldsymbol{Var^{cns}}$: Non-synonymous variants without nonsense variants
- $\boldsymbol{Var^{exp}}$: Expressed non-synonymous somatic variant
- ***PNE***: Predicted binding neoepitope
- $\boldsymbol{PNE^{exp}}$: PNE with evidence on transcript level of tumor tissue according to the RNA depth (> 5) or in case of unavailability inferred from corresponding gene expression values given in FPKM (> 1)
- $\boldsymbol{PNE^{prot}}$: PNE originating from proteins found to be abundant in the corresponding tumor tissue
- $\boldsymbol{PNE^{lig}}$: PNE with evidence on HLA class I ligandome level
- $\boldsymbol{WT^{lig}}$: Wild-type peptide, originating from a PNE, with evidence on HLA class I ligandome level

6.3.1 HLA Typing

The predicted HLA genotypes were consistent with all available experimentally determined HLA alleles except for patient QA004, where one experimentally typed allele was different on four-digit level. Instead of the experimental result (A*30:01), OptiType predicted A*30:09. However, the other five alleles matched on four-digit level. The three most frequent HLA alleles among HCC patients were A*02:01 (n=8, 50%), C*07:01 (n=6, 37%), A*01:01, and A*03:01 (n=5, 31%). Among the ALL cohort, the most frequent HLA alleles were A*02:01

(n=7, 30%), A*01:01, A*24:02 , C*07:01, C*12:03 (n=6, 26%), B*07:02, and C*07:02 (n=5, 22%). We did not find evidence for downregulation of HLA class I expression in both cohorts based on gene expression analysis. All of the determined HLA genotypes are given in Appendix Table E.3.

6.3.2 Analysis of Somatic Variants

We analyzed the number of somatic variants and their properties across all patients of the HCC and ALL cohort (Figure 6.2). After filtering, 149.7 ± 38.7 somatic variants (Vars) remained

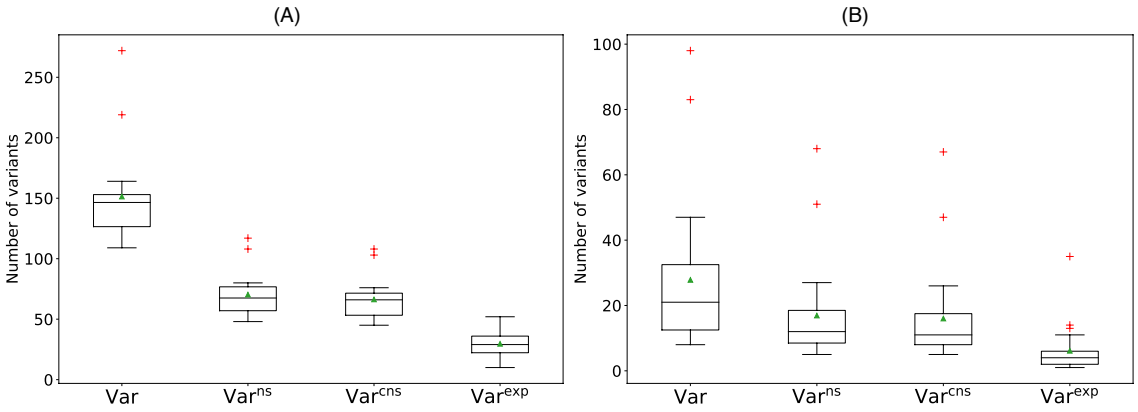

Figure 6.2: Number of observed somatic variants of different categories in the HCC (A) and ALL (B) cohort. Categories include all variants which passed initial filtering (*Var*), non-synonymous variants (Var^{ns}), non-synonymous variants without nonsense variants (Var^{cns}), and expressed Var^{cns} (Var^{exp}). Means are shown as green triangles.

on average per patient in the HCC cohort. The number of non-synonymous variants (Var^{ns}), averaged at 68.7 ± 19.1. Without nonsense mutations (Var^{cns}), we observed an average of 64.6 ± 18.1 somatic variants. RNA-based filtering resulted in 29 ± 11.1 variants (Var^{exp}) on average. We observed lower numbers of somatic variants for the ALL cohort. On average 27.8 ± 22.3 Vars, 16.9 ± 14.9 $Vars^{ns}$, and 16.0 ± 14.3 $Vars^{cns}$. Of the latter, 6.1 ± 7.1 variants were expressed on tumor. In total, we observed 1039 unique $Vars^{ns}$ in the HCC cohort (n=16), and 364 unique $Vars^{ns}$ in the ALL cohort (n=24). These $Vars^{ns}$ affected 864 and 327 unique genes respectively. 463 of these $Vars^{ns}$ had additional evidence on RNA level (Var^{exp}) in the HCC cohort and 137 in the ALL cohort. This translates to an average TMB of 1.89 ± 0.49 per Mb among the HCC patients and 1.12 ± 3.06 per Mb among the ALL patients. The patient-specific numbers are given in Appendix Table E.4 and E.5.

Further, we observed a low number of Var^{exp} that are shared across patients for both cohorts. Across the 16 HCC patients, 24 genes carried a Var^{exp} in two or more patients. This corresponds to 6.1% of all genes (n=392) carrying a Var^{exp} (Appendix Figure D.8 A). When investigating genes that carry a Var^{exp} in three or more patients, we identified six genes (Figure 6.3). The

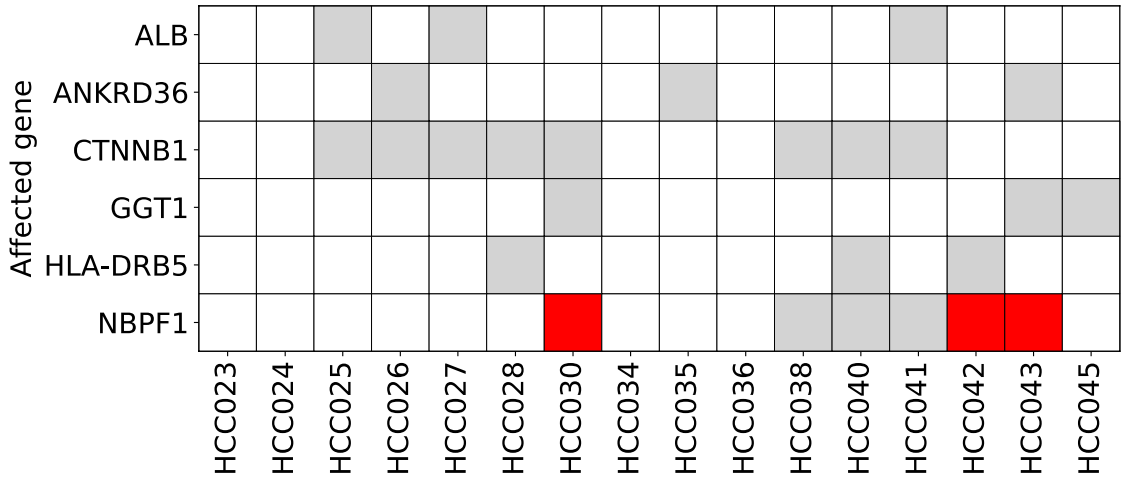

Figure 6.3: Shared genes across the HCC cohort which carry a Var^{exp}. Genes carrying expressed mutations were analyzed with respect to presence (depicted by a grey square) in multiple patients. The included six genes were affected by a Var^{exp} in $\geq$ three patients. The identical mutation was shared by three patients (red square).

genes *CTNNB1* and *NBPF1* were affected most often by an expressed mutation within our HCC cohort. Expressed mutations were observed in eight patients for *CTNNB1* (50%) and in six patients for *NBPF1* (38%). With respect to identical expressed mutations, only nine out of 463 (1.9%) expressed mutations were shared by two or more patients (Appendix Figure D.8 B). The mutation (16891365G>T) on *NBPF1* was observed in three patients (HCC030, HCC042, HCC043), which was the maximum with respect to patients sharing identical mutations. Other genes include two HLA class II genes, namely *HLA-DQA1* (13%) and *HLA-DRB5* (19%), and genes typically expressed in the liver, including *ALB* (19%), *APOB* (13%), *ABCA1* (13%), and *GGT1* (19%).

In the ALL cohort, seven genes were found to be affected by a Var^{exp} in more than one patient (Figure 6.4). This corresponds to 5.8% of unique genes affected by a *cnsTXV* (n=121). The genes *CDC27* and *PTPN11* are affected in three patients respectively. In QA005 and QA008 the mutation on *CDC27* was found to be identical (45235635A>C), whereas, in QA011, *CDC27* is affected by a different missense mutation (45214572G>A). In case of *PTPN11*, different positions are affected across the three patients (QA018, QA006, and QA009). Four identical Var^{exp} are shared across two patients each, corresponding to 2.9% of all (n=137) unique Var^{exp}. The affected genes include the aforementioned *CDC27*, *NCOR1*, *RPL19*, and *PRELID3B*. We identified ten genes (*DHX8*, *DLG1*, *DOT1L*, *EIF3B*, *HLA-DRB1*, *NCOR1*, *NOTCH1*, *RPL22*, *TP53*, *ZSWIM8*) that are affected by Var^{exp} across both cohorts.

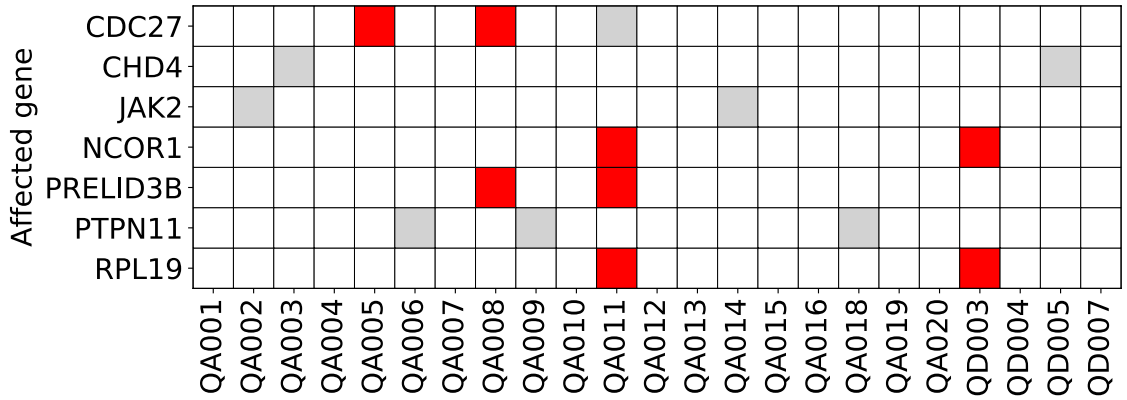

Figure 6.4: Shared genes across ALL cohort which are affected by a Var^{exp}. Genes affected by expressed mutations were analyzed with respect to presence (depicted by a grey square) in multiple patients. The included seven genes were affected by a Var^{exp} in $\geq$ two ALL patients. Identical mutations (red square), shared by two patients each, were observed in four cases.

6.3.3 Assessment of Neoepitopes

The evaluation of predicted neoepitopes was performed based on the annotated epitope prediction results of EPAA. On average, the peptide generation based on the somatic mutations resulted in a *peptide search space* (PSS) of 5,188 ± 2,140 peptides for the HCC cohort. 3,643 ± 1,196 peptides remained after filtering against the human proteome. Out of the resulting peptides, an average of 7.1 ± 2.4% were PNEs (5.2 ± 1.9% out of all peptides). The number of PNE^{exp} per patient averaged at 118 ± 40. Thus, on average 49 ± 8% of all PNEs (average: 244 ± 77) had additional evidence on transcriptome level (Figure 6.5A). In case of the ALL cohort, EPAA generated on average a PSS of 675 ± 569 peptides. Filtering against the human proteome resulted in an average of 584 ± 511 peptides. This corresponds to 86.8 ± 11.1% remaining peptides on average. The number of PNEs averaged at 35.8 ± 35.6 peptides. Thus, 5.4 ± 2.3% of all generated peptides (PSS) were predicted as PNEs. For 38.4 ± 22.5% of the PNEs, evidence on transcriptome level was given. Out of all peptides 2 ± 1.4% were PNE^{exp} on average, corresponding to an average of 13.2 ± 13.5 PNE^{exp} (Figure 6.5B). Hence, we observed on average about nine times more PNE^{exp} in the HCC cohort than in the ALL cohort.

With respect to the number of PNE^{exp}, we observed a moderate correlation ($r = 0.5$, $p = 0.049$) to the number of Var^{exp} for the HCC cases (Figure 6.6). In the ALL cohort, the results indicate a positive linear relation between the number of PNE^{exp} to the number of Var^{exp} ($r = 0.88$, $p = 3.5 \cdot 10^{-8}$). Every Var^{exp} generated on average 4.1± 2.5 PNE^{exp} among the HCC patients and 2.4 ± 1.7 PNE^{exp} in the ALL cohort. For the HCC cohort, we further investigated evidence of PNEs on proteome and ligandome level if available. In order to assess the evidence of PNEs on tumor proteome level, we annotated all PNEs with $\log_2$-transformed

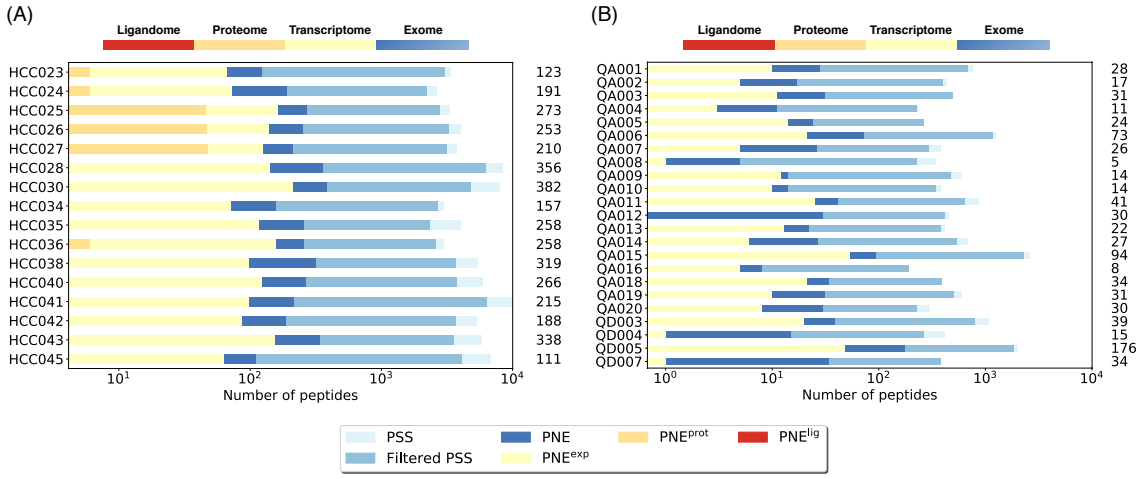

Figure 6.5: Number of peptides along the different omics layers. (A) We analyzed the HCC cohort with respect to evidence for neoepitopes on transcriptome, proteome and ligandome level. (B) In case of the ALL cohort, only transcriptome data was available. The number of potential neoepitope candidates decreases as we demand tumor expression of the corresponding mutation (PNE^{exp}) and tumor abundance of the corresponding protein (PNE^{prot}). We did not find evidence for neoepitopes on ligandome level. The number of PNEs per patient is given on the right of each plot.

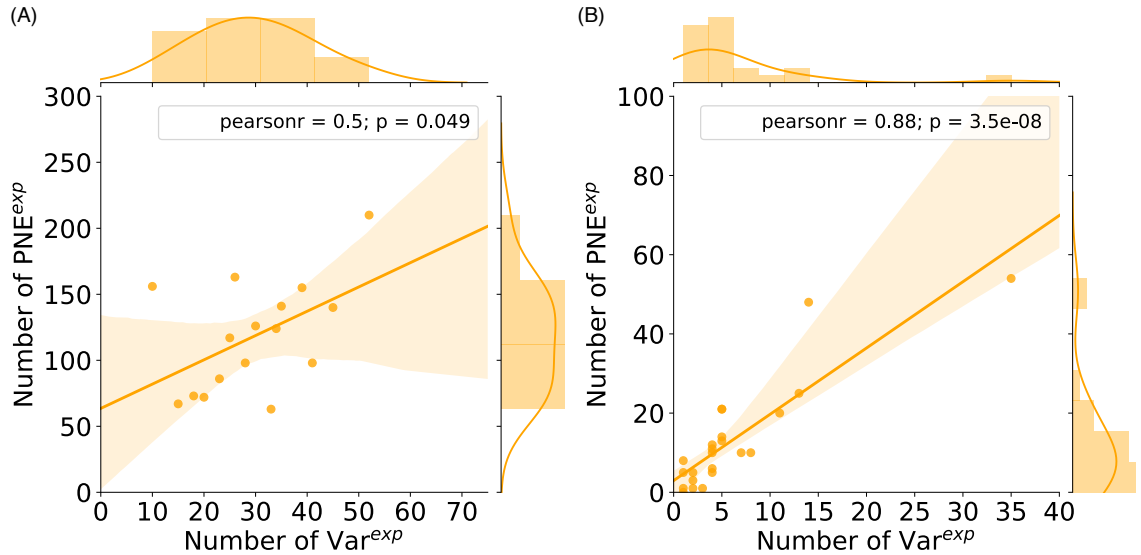

Figure 6.6: Relationship between the number of Var^{exp} and PNE^{exp} for the patients of the HCC cohort (A) and the ALL cohort (B). Data points are visualized as scatterplot with marginal histograms, as well as the linear regression model fit and kernel density fits.

intensities from shotgun proteome data, which was available for seven patients. In total, 159 PNEs were annotated as PNE^{prot}. This corresponds to 22.7 ± 21.1 PNE^{prot} and $17 \pm 14\%$ of PNE^{exp} on average. The 159 PNE^{prot} originate from 33 unique source proteins. For HCC023, three proteins were measured with an average $\log_2$-transformed intensity of 24.5 ± 0.1. Two

proteins each were measured for HCC024 and HCC036 (average $\log_2$ intensity: 26.0 ± 0.2 and 22.2 ± 3.1). Out of nine proteins measured for HCC025, eight were exclusively measured on tumor tissue. In the case of HCC026 and HCC027, five out of ten and three out of eight proteins were measured exclusively on tumor tissue. No PNE-associated proteins could be measured on tumor tissue for HCC034. On average, 9.8 ± 8.6% of PNEs were PNE^{prot}.

Further, we used HLA ligandome data in order to obtain evidence for naturally presented mutated HLA ligands. HLA ligandome data was available for all patients in the HCC cohort for non-malignant liver tissue and HCC. In total, we identified 22,443 peptide sequences (unique 15,054) of length 8–11 AA with an FDR of 5% on tumor tissue and 18,547 peptides (unique 11,775) on benign liver tissue. On average, we observed $1{,}403 \pm 621$ immunoprecipitated peptides on tumor and $1{,}159 \pm 525$ on non-malignant liver tissue. In total, 6,738 of the identified unique ligands are shared between matched tumor and non-malignant liver tissue. On a per-patient base, 568 ± 317 identified ligands were shared between tumor and non-malignant liver tissue on average, corresponding to 51 ± 11% of the unique peptides. We did not find evidence for any PNE^{lig}. However, we identified three WT^{lig} for HCC027, HCC028, and HCC041 on HCC. Manual inspection of the corresponding spectra revealed that only the identified WT^{lig} of HCC027 and HCC028 were real matches. In HCC027, the wild-type peptide (TERIIAVSF), as well as the mutated peptide, were predicted as a strong binder (NetMHC 4.0, affinity: 12.93 nM) for HLA-B*18:01, one of the patient's HLA alleles. The WT^{lig} (LPAHIPYQEL) identified in HCC028 was predicted as a weak binder for two alleles of the patient (HLA-B*07:02 and HLA-B*35:03). Both peptides were found to be known epitopes[i,ii], contained in the IEDB[251]. A condensed graphical representation of the observed numbers of neoepitopes and their evidence on multiple omics levels within the HCC cohort is given in Figure 6.7. Absolute numbers of variants and peptides per patient are given in Appendix Table E.4 and E.5.

To calculate an estimate of the probability of observing neoepitopes, we calculated the size of the theoretically presentable HLA class I ligandome as an upper estimate of the underlying search space. Therefore, we predicted the HLA-binding peptides of length 8–11 AA for all patients based on their personalized proteome, i.e., the human proteome including mutated proteins. The set of considered proteins was filtered for tumor expression according to the gene expression analysis results. The predictions results averaged at 1,219,745 unique predicted HLA-binding peptides. In addition to the number of unique peptide identifications (FDR 5%) from each HCC, we used the number of PNE^{exp} of each patient to estimate the probability of observing a neoepitope and utilized the hypergeometric distribution given by:

$$P(X = k) = \frac{\binom{n}{k}\binom{N-n}{M-k}}{\binom{N}{M}},$$

[i] http://www.iedb.org/epitope/441029
[ii] http://www.iedb.org/epitope/521537

where M is the number of predicted binding neoepitopes, N is the number of the estimated personalized tumor ligandome, n is the number of HLA class I immunoprecipitated peptides, and k is the number of observed neoepitopes. Using the above-mentioned measures resulted in an average probability of 0.15 $\pm$0.10 for observing at least one neoepitope in one run $(1-P(0))$. On average, we could, therefore, expect to find one neoepitope on average every 12th run per patient (calculated by $\frac{1}{1-P(0)}$).

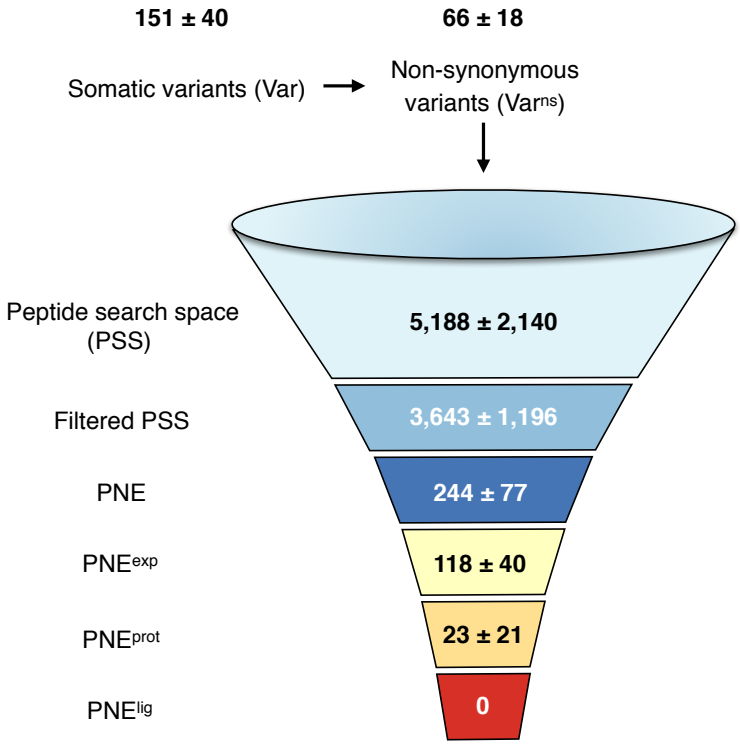

Figure 6.7: Numbers of potential neoepitopes within the HCC cohort along the analysis pipeline. Starting with non-synonymous mutations, a large number of predicted mutated peptides existed. This number decreased clearly after filtering for predicted neoepitopes and requiring evidence for their existence on transcriptome, proteome, and ligandome. We did not find evidence for neoepitopes on the ligandome level of tumor tissues.

6.3.4 Evaluation of the Neoepitope Identification Pipeline

In order to evaluate the sensitivity of our neoepitope identification pipeline, we further processed a publicly available data set of five melanoma patients, previously analyzed by Bassani-Sternberg et al.[414] (Figure 6.8). Starting with somatic variants and HLA ligandome data, we performed variant annotation, ligandomics identification, and neoepitope identification using EPAA. As expected, the average number of Var^{ns} was substantially higher than in our HCC and ALL cohort. The number of Var^{ns} averaged at 531 $\pm$ 419. Consequently, the number of PNE showed remarkable differences as well. On average, peptide predictions resulted in an

average of 1,550 ± 1,457 PNEs. Bassani-Sternberg et al. reported the identification of neoepitopes (PNE^{lig}) on human melanoma tissue[414]. We were able to reconfirm all PNE^{lig} of length 8–11 AA except one for Mel5 (ETSKQVTRW) and one for Mel15 (RIKQTARK). Both peptides were not predicted to be HLA-binding peptides. However, we discovered two additional PNE^{lig} for Mel15 that could be confirmed by manual inspection of corresponding MS spectra.

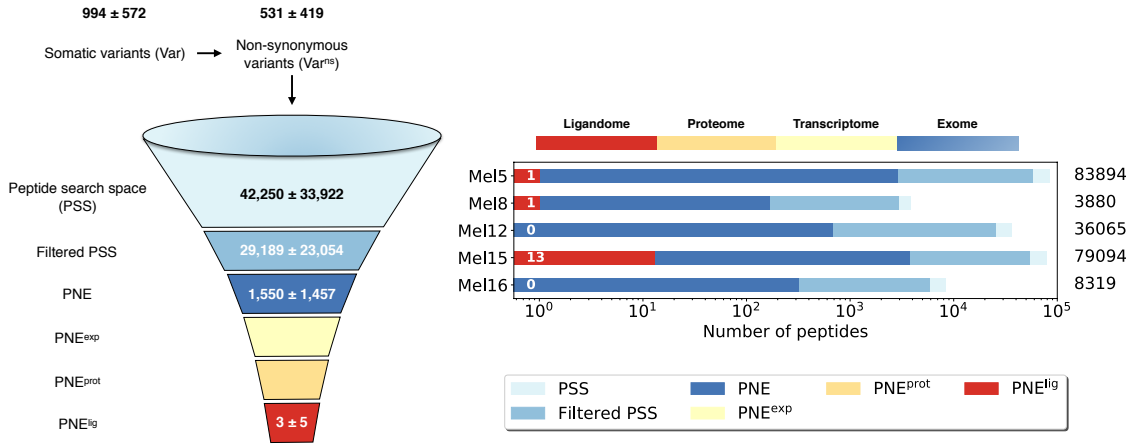

Figure 6.8: Numbers of potential neoepitopes for the publicly available melanoma data set. The data (n=5) was previously analyzed by Bassani-Sternberg et al.[414]. Starting with non-synonymous mutations, a large number of predicted mutated peptides existed. We used EPAA to predict PNE and annotated resulting peptides with HLA ligandomics identification results. Numbers are given as mean and standard deviation.

6.3.5 Assessment of Alternative Targets

Since we did not find evidence for neoepitopes on ligandome level for the HCC cohort, we analyzed the available data with respect to evidence for cancer/testis (CT) antigens in HCC. A list of known CT antigens was derived from CTDatabase[431] (accessed 2018-02-20). On proteome level, we quantified multiple members of the POTE gene family across the seven patients with available proteomics data: *POTEE* (multiple patients), *POTEH*, *POTEG* (HCC023, HCC026), *POTEC*, and *POTEB* (HCC023). Besides, *LDHC* was quantified in all patients and *TEX15* in four patients. *RBM46* was quantified in HCC023, HCC025, and HCC026. On ligandome level, we identified multiple HLA class I ligands mapping to CT antigens. We detected HLA ligands mapping to the following CT antigens: *ARMC3* (HCC045), *ATAD2* (HCC023, HCC045), *PRAME* (HCC041), and *TFDP3* (HCC045). Ligands mapping to *SSX1* were identified in HCC035 and HCC041. Besides, ligands mapping to six other members of the SSX gene family were identified in HCC041. A list of all CT antigens with evidence on proteome or ligandome level on tumor tissue and their corresponding UniProt ID is given in Appendix Table E.6.

6.3.6 Runtime Evaluation

Across all patients, the runtime of all performed workflow runs during this study was evaluated (Figure 6.9). In total, 444 workflow runs were submitted with an average runtime of 4.3 ± 8.9h. Runtimes include the data staging steps and potential idle time due to the unavailability of free cluster nodes. The 99th percentile of all workflow runtimes is 37.46h. The highest runtimes

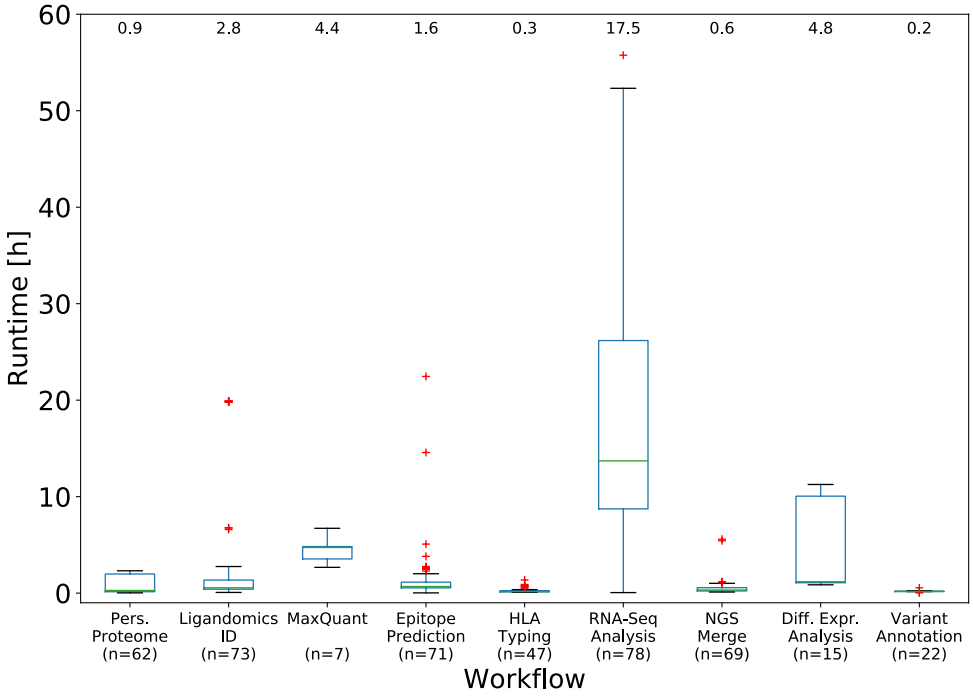

Figure 6.9: Runtime of performed workflow runs. For every workflow type (x-axis), the runtimes across all runs are plotted in hours. The maximum workflow runtime (77.4 h) of one RNA-Seq analysis workflow is not included in the plot. Average runtimes per workflow type are given within the plot.

were observed for the RNA-Seq analysis workflow. All RNA-Seq runs had an average runtime of 17.5 ± 13.6h and a maximum runtime of 77.4h, which was the maximum across all runs and caused by the unavailability of free cluster nodes. This was also the reason for outliers in terms of runtime for other analysis workflows. Variant annotation runs had an average runtime of 0.2 ± 0.1h with a minimum runtime of 1.38min, corresponding to the minimum runtime overall.

In summary, starting from somatic variants, the average runtime of the complete computational pipeline, including all omics layers, for the assessment of potential targets for personalized cancer vaccines can be estimated at 33 hours. This number is an upper estimate since some of the workflows could be submitted in parallel. In a typical setting, which includes WES and WTS data, the runtime for the complete pipeline can be estimated at 20 hours.

6.4 Discussion

Cancer vaccines are one option in targeted immunotherapy that traditionally focuses on TSAs. Recent advances in genome sequencing and in-depth analysis of cancers' mutanome revealed the existence of non-synonymous somatic mutations causing tumor-specific antigens. Due to their exclusive expression on tumors, their potency as immune targets should not be adversely affected by immune tolerance, and the risk of autoimmune side effects should be minimized. Thus, their potential to induce T cell-mediated immune responses makes them strong candidates for cancer vaccines. The availability of algorithms to identify cancer neoantigens renders their identification on a personal basis feasible in contrast to labor-intensive screening approaches[433]. Therefore, it is possible to exploit the mutational neoepitope repertoire of tumors on a full scale. In contrast to established generic therapeutic approaches, based on screening for the presence or absence of mutations and the use of off-the-shelf drugs, personalized epitope-based vaccines are based on a patients' tumor mutanome. Previous studies have shown the feasibility of *in silico*-assisted processes to derive patient-specific neoantigen candidates and their potential in human cancer treatment[149,150].

Here, we demonstrated the application of *iVacPortal* in the context of personalized epitope-based vaccines for two cohorts of different cancer entities. We utilized the portal's features for data management and the generation of peptides based on patient-specific private somatic mutations as poly-peptide cancer vaccine targets. Further, we addressed the problem of filtering and prioritizing these candidates, as well as finding evidence of neoepitopes on multiple omics layers by performing an in-depth analysis of the generated results. We applied implemented workflows for HLA typing, variant annotation, gene expression analysis, LFQ, HLA class-I ligand identification, and epitope prediction on the data of 40 patients. As a central component, the implemented neoantigen identification pipeline EPAA, with annotation functionality of previously derived results, facilitates the investigation of neoepitope evidence based on genomics, transcriptomics, proteomics, and ligandomics. Based on the available somatic variant calls, we observed quantities of non-synonymous mutations for both cancer entities, which are in the range of previously published numbers[407,434] on mutational burden (Figure 6.10). A study[434] on 27 HCC patients reported an average number of 68 somatic non-synonymous mutations. HCC cases from TCGA (TCGA-LIHC) possessed a higher average number of $Vars^{ns}$ (90 ± 100, n=363) than our HCC cohort. Still, the average numbers are substantially lower than for melanoma (461 ± 761, n=467). As expected, the observed number of somatic variants of the ALL cohort was lower than in HCC. As reported earlier[435], the application of neoantigen-based vaccines for cancer entities with lower mutational burden than in entities such as melanoma might be challenging. Nevertheless, Tran et al. proved the existence of tumor-infiltrating $CD4^+$ and $CD8^+$ cells that recognized neoepitopes in tumors with low mutation load[413]. Still, in the case of the absence of identified neoepitopes, our framework could be used to determine

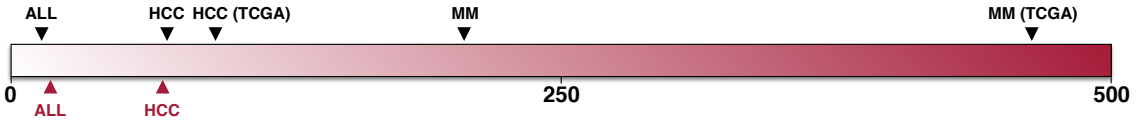

Figure 6.10: Average number of somatic mutations of different cancer entities. The numbers for ALL and melanoma (MM), were taken from Vogelstein et al.[407]. A study[434] reported 68 non-synonymous somatic mutations on average for HCC (n=27, black) which is slightly higher than in our study (n=16, 66 $Vars^{ns}$, red). The number of $Vars^{ns}$ for HCC cases from TCGA averaged at 90 $Vars^{ns}$ (HCC (TCGA), n=363). The average number of non-synonymous somatic mutations of our ALL cohort (n=23, 16 $Vars^{ns}$, red) is in line with reported numbers (black, 14 $Vars^{ns}$). Cancer entities such as MM (203 $Vars^{ns}$) usually possess a much higher mutational burden, corroborated by TCGA cases (MM (TCGA), n=467, 461 $Vars^{ns}$).

alternative targets for treatments, as shown. Such targets could include expressed CT antigens, differentiation antigens, immune checkpoint modulation targets, and identified HLA ligands, known to be immunogenic. As presented, we found evidence for multiple known CT antigens across patients. CT antigens are promising targets for immunotherapy since they are immunogenic, cancer-specific, and frequently expressed[436]. Their application includes cancer vaccination and adoptive T-cell transfer. Concerning potential neoepitope candidates, we identified at least one PNE^{exp} in all patients except one across both cohorts. Out of all predicted neoepitopes, 49% (HCC) and 38% (ALL) were found to correspond to an expressed transcript on tumor tissue. Further, we quantified 33 proteins of neoepitopes on tumor tissue. For two of these proteins, encoded by *ALB* (HCC025) and *RECQL* (HCC026), mutated proteins could be validated via shotgun proteomics[437]. Nevertheless, we were not able to identify neoepitopes in the corresponding patient-specific HLA class I ligandome. We detected one WT^{lig} for two patients each. Since both of them were predicted to be binders for the patients' HLA alleles, they could still serve as alternative targets. Especially in the case of HCC027, where the respective protein could only be measured on tumor and not on the non-malignant liver tissue. Bassani-Sternberg et al. showed that the direct identification of immunogenic mutated peptides by MS on tumor is possible[414]. We proved the sensitivity of our neoepitope identification pipeline by reconfirming their identified neoepitopes using the same data. Two reported neoepitopes[414] were excluded from the result by our pipeline since they were not predicted as binders for the HLA alleles. As shown, the number of present somatic mutations is much higher in the melanoma data set than in our investigated cohorts, affirmed by the investigation of TCGA data for both cancer entities. The resulting substantially lower amount of mutated neoepitopes might, therefore, be one reason for the absence of successfully identified naturally processed HLA-I ligands.

We reinforced this argument by calculating an estimate of the theoretically present tumor ligandome and the comparison to observed ligand identification rates for our HCC cohort. The consideration of the complete set of binding peptides (8–11 AA) as ligandome search

space is presumably an overestimation of the actual present ligandome. However, since we do not know which peptides are present and bind to the HLA molecules, it still provides a rough estimate of the odds to measure a ligand. Additionally, the resulting numbers for the probability of observing neoepitopes are in line with our past experiences. Further, the identification rate of truly-immunogenic neoepitopes might be negatively influenced by several factors. In general, the computational assessment of immunogenicity is still mostly based on HLA binding predictions. Although prediction algorithms did improve, the rate of predicted high-affinity binders that can be recognized by patient's T cells in a tumor environment seems to be low[148] and might involve inaccuracies, especially for rare alleles. Furthermore, the identification via MS on the HLA ligandome is strongly dependent on the sensitivity of the employed devices and *in silico* identification pipelines. Concerning MS, several reasons might introduce biases in HLA ligandomics data, such as the very low frequency of cysteine, the loss of too hydrophobic or too hydrophilic peptides, and poor ionizability or fragmentation[438]. Another factor that prevents the discovery of neoepitopes might be immunoediting[86]. Mechanisms such as the selection of clones with downregulated antigen processing and presentation on HLA or with silenced genes of antigens targeted by the immune system presumably limit the potential identification rate of immunogenic neoepitopes as well.

To extend the pool of potential immunogenic HLA ligands, one might consider other sources of neoepitopes such as alternative splicing[439,440] and post-translational protein splicing[441], T-cell epitopes associated with peptide processing (TEIPP)[442], tumor-specific phosphopeptides[443], and cryptic peptides[444]. Furthermore, Kreiter et al. showed that $CD4^+$ T cells recognize neoepitopes[142]. Recently, Marty et al. emphasized the importance of $CD4^+$ T cells in anti-tumor immunity and HLA-II presentation in tumor evolution[445]. Therefore, HLA class II epitopes could be considered as well. Corresponding functionality is already available in iVacPortal. However, the results of *in silico* prediction algorithms are not as reliable as HLA class I algorithms[446].

Another important factor for the assessment of potential neoepitopes and therefore finding potential constituents for targeted immunotherapy are turnaround times. The presented runtimes suggest that the computational analyses are not the main factor influencing the turnaround time since experimental steps usually take time in the range of days. However, since these processes cannot be sped up easily in most cases, it is of great importance that the computational pipeline runs as reliable and fast as possible. There are different aspects that can be addressed in order to speed up processing. As presented, the highest runtimes were observed for the RNA-Seq analysis workflow with 17.5 ± 13.6h on average. This process could be sped up by using a different RNA-Seq aligner, such as STAR[208], which has been shown to outperform TopHat2[207]. Alternatively, alignment-free methods, such as Salmon[447] and Kallisto[228] could be used for transcriptome quantification to reduce runtime. Bray et al. showed that Kallisto[228] outperforms other methods, including TopHat2[207], with respect to runtime, and the authors

of Salmon stated that their method is comparable to Kallisto in terms of speed[447]. Even higher reduction of runtimes could be achieved by using GPU accelerated sequence alignment libraries, such as the recently published GASAL2[448]. In general, different parts of the computational pipeline could benefit in terms of runtime from parallelization. One option would be to process chromosomes or even transcripts in parallel for the peptide generation and prediction in the epitope prediction step. The parallelization by chromosome has already been implemented in the nf-core[449] version of this pipeline (`https://nf-co.re/epitopeprediction`). Besides, steps of the pipeline that usually do not require any user input, such as merging sequencing reads from different lanes, could be automated and triggered directly upon data arrival to further reduce turnaround times.

To summarize, we analyzed data from 40 patients of two cancer entities with respect to the identification of potential neoepitopes. We thereby demonstrated the usefulness of iVacPortal's data management functionality and the effectiveness of implemented and integrated computational pipelines for the identification of neoepitopes based on somatic mutations. Despite the complexity of this multi-omics project and associated large amounts of data, we could provide short turnaround times by using our web-based platform, the integrated workflow system, and connected HPC resources. For multiple patients, we found evidence for neoepitope-generating mutations on transcriptome and proteome level. Still, we did not find evidence for any neoepitope in the HLA-I ligandome. Besides the presented project, *iVacPortal* has been utilized in the context of personalized cancer vaccines in more than 300 cases and is currently utilized in a prospective phase I/II study on ALL (Eudra-CT-Nr. 2015-005281-29).

Chapter 7

Conclusion and Outlook

The ultimate goal of personalized medicine is to provide the right treatment for the right patient at the right time. To achieve this goal, a comprehensive analysis of the patients and their diseases is necessary, though often hard to bring to clinical practice. Notably, in cancer treatment, generic therapy approaches are prone to be inefficient due to differences in genetic predisposition, a considerable heterogeneity across cancer types, and individual tumor mutational burdens. Here, patients can benefit from personalized therapy approaches. Personalized *epitope-based vaccines* (EVs) for tumor therapy try to account for these wide variations across tumors and patients by guiding the selection of treatment on tumor specimens and the patient's immune system. In the past years, NGS platforms, as well as the associated decreasing costs and turnaround times, were critical for the emergence of such therapeutic options in personalized medicine and will continue to advance its development.

The ever-increasing availability of patient and cancer genomes opens up unique possibilities but also presents new challenges, especially with respect to data analysis. In the context of EVs for tumor therapy, one challenge is to gain insights into the individuality of the immune system, that is among others shaped by the variability of the *human leukocyte antigen* (HLA) gene cluster. At the time of this thesis, HLA genotyping was either done using labor-intensive experimental methods or *in silico* approaches that relied on NGS data generated for the sole purpose of HLA typing. Emerging *in silico* approaches based on readily-available WGS, WES, and WTS data did not yet achieve the precision necessary for clinical diagnosis. By the development of *OptiType*, we provided a computational method to perform fully automated HLA genotyping on NGS data on a large scale with high precision. We successfully applied OptiType on data sets of state-of-the-art sequencing technologies and proved its accuracy on four-digit resolution, crucial in clinical applications such as personalized vaccine design. Since its publication, OptiType has been applied in several studies on cancer genomics[450–456]. Marty et al. successfully performed HLA typing for the majority of 9,839 cancer patients derived from TCGA, showing OptiType's applicability in large-scale cancer genomics projects and the demand in studies on immune-

regulatory mechanisms, especially in the context of cancer. Although multiple HLA genotyping methods were suggested, OptiType is still among the most accurate methods. xHLA[318] which achieved slightly higher accuracy than OptiType in a benchmark performed by Xie et al., uses a similar ILP-based approach. Future HLA genotyping methods will benefit from the increasing number of available HLA allele sequences. Further, third-generation sequencing technologies, such as Pacific Biosciences' SMRT, capable of producing sequences that span entire HLA genes, might be beneficial if acceptable error rates can be achieved in the future. The same applies to developments on HLA class II typing, that might be of even higher interest with increasing precision of HLA class II binding prediction methods and further evidence of $CD4^+$ T-cell responses caused by *EVs*.

Our second contribution is the development and evaluation of a new model for T-cell immunogenicity prediction. By implementing the underlying *distance-to-self* functionally in *ImmunoNodes*, we facilitate future efforts on similar approaches. Immunogenicity is of vital importance when it comes to EVs and other biological drugs since it affects efficacy and safety. Immunological responses, evaluated by the occurrence of peptide-specific $CD4^+$ and $CD8^+$ T cells, can be experimentally determined using flow cytometry on peripheral blood or the recently developed cytometry by time-of-flight (CyTOF). Single-cell assays include MHC-peptide tetramer staining, ELISPOT assays, and intracellular cytokine assays for IFN-γ, TNF, and IL-2. However, there is a high demand for *in silico* methods to reliably predict immunogenicity before therapy for a large number of peptide candidates. We proposed a method that models peripheral tolerance based on gut microbiome data in addition to central tolerance to improve the assessment of the T-cell repertoire. The approach is based on the assumption that peptides, which have a lower similarity (higher distance) to self-peptides, are more likely to induce immunological responses due to negative selection mechanisms. As shown, the incorporation of other immunological measures did not improve prediction performance, which might be due to the inaccurate modeling of peripheral tolerance because of insufficient information on the commensal microbiota. Certainly, the amount of data on the human gut microbiome will increase since its recognized involvement in human health and disease. Presumably, standard diagnosis and treatment evaluation will soon include the collection and analysis of gut microbiome data and thus provide new insights with respect to its role and composition. On the other hand, the proposed models still only indirectly account for the T-cell and TCR repertoire. Recent advances in TCR sequencing might close this gap and provide the basis for future developments of T-cell immunogenicity prediction to ultimately reach the precision required for clinical applications. High-throughput sequencing methods on pooled immune cell populations, and even more importantly, on a single-cell level can be used to identify TCR chains accurately, and in the latter case, even in a paired form. Together with gut microbiome analysis on a personalized level, these developments can provide the basis for personalized *in silico* approaches for the prediction of T-cell immunogenicity.

Although new methods and advances in high-throughput technologies open up new opportunities, they also present new challenges concerning data management and analysis. In the second part of this thesis (Chapter 5), we presented *qPortal* a web-based platform for data-driven biomedical research. Our portal solution combines easy-to-use user interfaces with a workflow system and provides data management functionaliy using a backend that utilizes a variety of concepts and technologies, such as relational databases, data stores, data models, and data transfer. As depicted, multi-omics projects present challenges due to multiple involved laboratories and heterogeneous data types. We tackled these challenges by providing a system for exhaustive metadata collection, central storage, and (meta)data management. The collected data is usually bound to strictly regulated terms regarding data security, access, and confidentiality. Data security is of prime importance, especially with respect to clinical data. Therefore, we implemented a two-step security process. To empower users to run analysis pipelines on HPC infrastructures, we implemented a workflow system interface. Employing this generic system, we developed a workbench for the design of personalized EVs, which facilitates common tasks in these projects from planning to analysis. Regarding data analysis, we developed a pipeline for the prediction of *neoepitopes* and their assessment as cancer vaccine candidates. To determine the presence of these candidates, our pipeline integrates genome, transcriptome, proteome, and ligandome data. More than ever, there is a need for solutions to standardize analysis pipelines and to provide reproducibility, especially with the increasing relevance of *in silico* approaches in clinical practice. With extensive metadata collection and our workflow system, we already made efforts in this direction. To take it one step further, the utilization of container solutions, such as Singularity[276] in combination with workflow systems such as Nextflow[277] should be pursued and promoted. Additionally, joint community efforts such as nf-core[449] (`https://nf-co.re/`) for the collection and curation of state-of-the-art analysis pipelines will enhance standardization and reproducibility. Ultimately, these efforts will also drive the development of cloud-based solutions to deliver software components to the data.

In the last part of this thesis (Chapter 6), we applied our contributions in two research projects on personalized EVs for cancer therapy, which demonstrated the feasibility of such efforts through *iVacPortal*. Our analysis resulted in the identification of neoepitope vaccine candidates based on patient- and tumor-specific mutations for both cancer entities. Additionally, we provided possibilities for the assessment of alternative targets. We did not find evidence for neoepitopes on ligandome level but indicated possible reasons for this, such as the insufficient sensitivity of employed devices and biases in HLA ligandomics data. Still, Bassani-Sternberg et al.[414] proved the feasibility of identifying neoepitopes on ligandome using more sensitive instrumentation. However, there are also fundamental differences between cancer entities, such as the tumor mutational burden, accompanied by differences in the number of potential neoepitopes and immunotherapy success[457].

To gain new insights into cancer immunotherapy and the effector mechanisms, it is even more critical to not only record suggested therapeutic options and composed vaccines but also the immune state of a patient, using immune monitoring and clinical outcomes. Efficient, standardized methods for the analysis of immune monitoring and immunogenicity assays are still missing. Furthermore, scalable and high-performance big data storage solutions, such as the ones used by ICGC[363] and TCGA[364] will be needed to efficiently store and query cancer genome data and provide the basis for further developments in the assessment of therapy efficacy and clinical outcome. Such systems will also be beneficial for the transition of *in silico* approaches for personalized medicine to clinical practice, especially with the emergence of molecular tumor boards. Moreover, global initiatives such as the *Global Alliance for Genomics and Health* (GA4GH)[458] will be required to define technical standards for genomic data sharing, ethics, and data security.

To summarize, the presented developments constitute a contribution to the pipeline for personalized peptide-based cancer vaccine design. With OptiType, we enable the prediction of HLA genotypes, a prerequisite for the design of personalized cancer vaccines. Our efforts on T-cell reactivity prediction assessed the potential of modeling peripheral tolerance and build the foundation for future works. The development of iVacPortal, as a web-based workbench, enables users to access and apply our developments in clinical research projects. The added value of iVacPortal was demonstrated by its application in two projects concerning the assessment of cancer vaccine targets. We anticipate that platforms, such as qPortal, will be indispensable cornerstones for state-of-the-art biomedical research infrastructures and promote the translation of *in silico* approaches into clinical practice.

Bibliography

[1] Bray F., et al. (2018). Global cancer statistics 2018: Globocan estimates of incidence and mortality worldwide for 36 cancers in 185 countries. *CA Cancer J Clin*, 68(6):394–424. 1

[2] Hanahan D. and Weinberg R. A. (2000). The hallmarks of cancer. *Cell*, 100(1):57–70. 1, 19

[3] Lazebnik Y. (2010). What are the hallmarks of cancer? *Nat Rev Cancer*, 10(4):232–3. 1

[4] Greaves M. and Maley C. C. (2012). Clonal evolution in cancer. *Nature*, 481(7381):306–13. 1

[5] Landau D. A., et al. (2013). Evolution and impact of subclonal mutations in chronic lymphocytic leukemia. *Cell*, 152(4):714–26. 1

[6] zur Hausen H. (1991). Viruses in human cancers. *Science*, 254(5035):1167–73. 2

[7] Parkin D. M. (2006). The global health burden of infection-associated cancers in the year 2002. *Int J Cancer*, 118(12):3030–44. 2

[8] Plummer M., et al. (2016). Global burden of cancers attributable to infections in 2012: a synthetic analysis. *Lancet Glob Health*, 4(9):e609–16.

[9] de Martel C., et al. (2012). Global burden of cancers attributable to infections in 2008: a review and synthetic analysis. *Lancet Oncol*, 13(6):607–15. 2

[10] Schiller J. T., Castellsague X., and Garland S. M. (2012). A review of clinical trials of human papillomavirus prophylactic vaccines. *Vaccine*, 30 Suppl 5:F123–38. 2

[11] Szmuness W., et al. (1980). Hepatitis B vaccine: demonstration of efficacy in a controlled clinical trial in a high-risk population in the United States. *N Engl J Med*, 303(15):833–41. 2

[12] Berard F., et al. (2000). Cross-priming of naive CD8 T cells against melanoma antigens using dendritic cells loaded with killed allogeneic melanoma cells. *J Exp Med*, 192(11):1535–44. 3

[13] Salcedo M., et al. (2006). Vaccination of melanoma patients using dendritic cells loaded with an allogeneic tumor cell lysate. *Cancer Immunol Immunother*, 55(7):819–29. 3

[14] Escors D. (2014). Tumour immunogenicity, antigen presentation and immunological barriers in cancer immunotherapy. *New J Sci*, 2014. 3

[15] Schumacher T. N. and Schreiber R. D. (2015). Neoantigens in cancer immunotherapy. *Science*, 348(6230):69–74. 3, 98

[16] Walter S., et al. (2012). Multipeptide immune response to cancer vaccine IMA901 after single-dose cyclophosphamide associates with longer patient survival. *Nat Med*, 18(8):1254–61. 3

[17] Slingluff, C. L. J., et al. (2013). A randomized phase II trial of multiepitope vaccination with melanoma peptides for cytotoxic T cells and helper T cells for patients with metastatic melanoma (E1602). *Clin Cancer Res*, 19(15):4228–38. 3

[18] Snyder A., et al. (2014). Genetic basis for clinical response to CTLA-4 blockade in melanoma. *N Engl J Med*, 371(23):2189–2199. 3

[19] Olsen L. R., et al. (2014). Bioinformatics for cancer immunotherapy target discovery. *Cancer Immunol Immunother*, 63(12):1235–49. 3

[20] Saunders C. T., et al. (2012). Strelka: accurate somatic small-variant calling from sequenced tumor-normal sample pairs. *Bioinformatics*, 28(14):1811–7. 4, 34, 104, 199

[21] Larson D. E., et al. (2012). SomaticSniper: identification of somatic point mutations in whole genome sequencing data. *Bioinformatics*, 28(3):311–7. 34

[22] do Valle I. F., et al. (2016). Optimized pipeline of MuTect and GATK tools to improve the detection of somatic single nucleotide polymorphisms in whole-exome sequencing data. *BMC Bioinformatics*, 17(Suppl 12):341.

[23] Reble E., Castellani C. A., Melka M. G., O'Reilly R., and Singh S. M. (2017). VarScan2 analysis of de novo variants in monozygotic twins discordant for schizophrenia. *Psychiatr Genet*, 27(2):62–70. 4

[24] Kroigard A. B., Thomassen M., Laenkholm A. V., Kruse T. A., and Larsen M. J. (2016). Evaluation of nine somatic variant callers for detection of somatic mutations in exome and targeted deep sequencing data. *PLoS One*, 11(3):e0151664. 4

[25] Feldhahn M., Donnes P., Thiel P., and Kohlbacher O. (2009). FRED–a framework for T-cell epitope detection. *Bioinformatics*, 25(20):2758–9. 4, 38

[26] Schubert B., et al. (2016). FRED 2: an immunoinformatics framework for Python. *Bioinformatics*, 32(13):2044–6. 4, 38, 59, 85

[27] Lucan C., et al. (2016). HLA Genotyping using Next Generation Sequencing. *Rom J Intern Med*, 54(2):98–104. 4

[28] Holcomb C. L., et al. (2011). A multi-site study using high-resolution HLA genotyping by next generation sequencing. *Tissue Antigens*, 77(3):206–17.

[29] Wang C., et al. (2012). High-throughput, high-fidelity HLA genotyping with deep sequencing. *Proc Natl Acad Sci U S A*, 109(22):8676–81.

[30] Shiina T., et al. (2012). Super high resolution for single molecule-sequence-based typing of classical HLA loci at the 8-digit level using next generation sequencers. *Tissue Antigens*, 80(4):305–16. 42

[31] De Santis D., et al. (2013). 16(th) IHIW : review of HLA typing by NGS. *Int J Immunogenet*, 40(1):72–6. 4

[32] Warren R. L., et al. (2012). Derivation of HLA types from shotgun sequence datasets. *Genome Med*, 4(12):95. 4, 43, 48, 52

[33] Boegel S., Scholtalbers J., Lower M., Sahin U., and Castle J. C. (2015). In silico HLA typing using standard RNA-Seq sequence reads. *Methods Mol Biol*, 1310:247–58. 43, 44

[34] Boegel S., et al. (2012). HLA typing from RNA-Seq sequence reads. *Genome Med*, 4(12):102. 52

[35] Kim H. J. and Pourmand N. (2013). HLA typing from RNA-seq data using hierarchical read weighting [corrected]. *PLoS One*, 8(6):e67885. 43

[36] Major E., Rigo K., Hague T., Berces A., and Juhos S. (2013). HLA typing from 1000 genomes whole genome and whole exome illumina data. *PLoS One*, 8(11):e78410. 4, 43, 44, 48

[37] Hakenberg J., et al. (2003). MAPPP: MHC class I antigenic peptide processing prediction. *Appl Bioinformatics*, 2(3):155–8. 4, 56

[38] Donnes P. and Kohlbacher O. (2005). Integrated modeling of the major events in the MHC class I antigen processing pathway. *Protein Sci*, 14(8):2132–40. 37

[39] Stranzl T., Larsen M. V., Lundegaard C., and Nielsen M. (2010). NetCTLpan: pan-specific MHC class I pathway epitope predictions. *Immunogenetics*, 62(6):357–68. 37

[40] Tenzer S., et al. (2005). Modeling the MHC class I pathway by combining predictions of proteasomal cleavage, TAP transport and MHC class I binding. *Cell Mol Life Sci*, 62(9):1025–37. 4, 56

[41] Gopalakrishnan V., et al. (2018). Gut microbiome modulates response to anti-PD-1 immunotherapy in melanoma patients. *Science*, 359(6371):97–103. 4, 56, 63

[42] Tung C. W., Ziehm M., Kamper A., Kohlbacher O., and Ho S. Y. (2011). POPISK: T-cell reactivity prediction using support vector machines and string kernels. *BMC Bioinformatics*, 12:446. 4, 37, 56

[43] Toussaint N. C., Feldhahn M., Ziehm M., Stevanović S., and Kohlbacher O. (2011). T-cell epitope prediction based on self-tolerance. *BCB'11: Proceedings of the 2nd ACM International Conference on Bioinformatics, Computational Biology and Biomedicine*, pages 584–588. 4, 37, 56, 57, 58, 59, 60, 62, 63

[44] Lathrop S. K., et al. (2011). Peripheral education of the immune system by colonic commensal microbiota. *Nature*, 478(7368):250–4. 4, 55, 63

[45] Honda K. and Littman D. R. (2016). The microbiota in adaptive immune homeostasis and disease. *Nature*, 535(7610):75–84. 4, 55, 63

[46] Afgan E., et al. (2016). The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2016 update. *Nucleic Acids Res*, 44(W1):W3–W10. 5, 38, 39

[47] Reich M., et al. (2006). GenePattern 2.0. *Nat Genet*, 38(5):500–1. 5, 39, 66

[48] Janeway C., Murphy K. M., and Weaver C. *Janeway's immunobiology*. Garland Science Taylor & Francis Group, New York London, 9th edition (2017). 7, 11, 12, 15

[49] Marsh S. G., et al. (2010). Nomenclature for factors of the HLA system, 2010. *Tissue Antigens*, 75(4):291–455. 11

[50] Toussaint N. C. and Kohlbacher O. (2009). Towards in silico design of epitope-based vaccines. *Expert Opin Drug Discov*, 4(10):1047–60. 12

[51] Weinberg R. A. *The biology of cancer*. Garland Science, Taylor & Francis Group, New York, second edition (2014). 17

[52] Mendelsohn J., Howley P. M., Israel M. A., Gray J. W., and Thompson C. *The molecular basis of cancer*. Saunders/Elsevier, Philadelphia, PA, edition 4 edition (2015). 17, 18

[53] McGuire S. (2016). World cancer report 2014. geneva, switzerland: World health organization, international agency for research on cancer, who press, 2015. *Adv Nutr*, 7(2):418–9. 18

[54] Stewart B. W., Wild C. P., and International Agency for Research on Cancer and World Health Organization. *World cancer report 2014*. International Agency for Research on Cancer WHO Press, Lyon, France Geneva, Switzerland (2014). 18

[55] Weinberg R. A. (1991). Tumor suppressor genes. *Science*, 254(5035):1138–46. 19

[56] Vogelstein B. and Kinzler K. W. (2004). Cancer genes and the pathways they control. *Nat Med*, 10(8):789–99. 19

[57] Kastan M. B. and Bartek J. (2004). Cell-cycle checkpoints and cancer. *Nature*, 432(7015):316–23. 19

[58] Hoeijmakers J. H. (2009). DNA damage, aging, and cancer. *N Engl J Med*, 361(15):1475–85. 19

[59] Jones S., et al. (2010). Frequent mutations of chromatin remodeling gene ARID1A in ovarian clear cell carcinoma. *Science*, 330(6001):228–31. 19

[60] Hanahan D. and Weinberg R. A. (2011). Hallmarks of cancer: the next generation. *Cell*, 144(5):646–74. 19, 20

[61] Cheng N., Chytil A., Shyr Y., Joly A., and Moses H. L. (2008). Transforming growth factor-beta signaling-deficient fibroblasts enhance hepatocyte growth factor signaling in mammary carcinoma cells to promote scattering and invasion. *Mol Cancer Res*, 6(10):1521–33. 19

[62] Sherr C. J. and McCormick F. (2002). The RB and p53 pathways in cancer. *Cancer Cell*, 2(2):103–12. 19

[63] Evan G. and Littlewood T. (1998). A matter of life and cell death. *Science*, 281(5381):1317–22. 19

[64] Ferrara N. (2002). VEGF and the quest for tumour angiogenesis factors. *Nat Rev Cancer*, 2(10):795–803. 20

[65] Vesely M. D., Kershaw M. H., Schreiber R. D., and Smyth M. J. (2011). Natural innate and adaptive immunity to cancer. *Annu Rev Immunol*, 29:235–71. 20

[66] Grivennikov S. I., Greten F. R., and Karin M. (2010). Immunity, inflammation, and cancer. *Cell*, 140(6):883–99. 20

[67] Dunn G. P., Bruce A. T., Ikeda H., Old L. J., and Schreiber R. D. (2002). Cancer immunoediting: from immunosurveillance to tumor escape. *Nat Immunol*, 3(11):991–8. 20

[68] Kim R., Emi M., and Tanabe K. (2007). Cancer immunoediting from immune surveillance to immune escape. *Immunology*, 121(1):1–14. 20

[69] Fridman W. H., Pages F., Sautes-Fridman C., and Galon J. (2012). The immune contexture in human tumours: impact on clinical outcome. *Nat Rev Cancer*, 12(4):298–306. 20

[70] Pages F., et al. (2010). Immune infiltration in human tumors: a prognostic factor that should not be ignored. *Oncogene*, 29(8):1093–102. 20

[71] Yang L., Pang Y., and Moses H. L. (2010). TGF-beta and immune cells: an important regulatory axis in the tumor microenvironment and progression. *Trends Immunol*, 31(6):220–7. 20

[72] Shields J. D., Kourtis I. C., Tomei A. A., Roberts J. M., and Swartz M. A. (2010). Induction of lymphoidlike stroma and immune escape by tumors that express the chemokine CCL21. *Science*, 328(5979):749–52. 20

[73] Mougiakakos D., Choudhury A., Lladser A., Kiessling R., and Johansson C. C. (2010). Regulatory T cells in cancer. *Adv Cancer Res*, 107:57–117. 20

[74] Chao M. P., Weissman I. L., and Majeti R. (2012). The CD47-SIRPα pathway in cancer immune evasion and potential therapeutic implications. *Curr Opin Immunol*, 24(2):225–32. 20

[75] Khanna R. (1998). Tumour surveillance: missing peptides and MHC molecules. *Immunol Cell Biol*, 76(1):20–6. 20

[76] Valastyan S. and Weinberg R. A. (2011). Tumor metastasis: molecular insights and evolving paradigms. *Cell*, 147(2):275–92. 21

[77] Gilman A. and Philips F. S. (1946). The biological actions and therapeutic applications of the B-chloroethyl amines and sulfides. *Science*, 103(2675):409–36. 21

[78] Cunningham D., et al. (2004). Cetuximab monotherapy and cetuximab plus irinotecan in irinotecan-refractory metastatic colorectal cancer. *N Engl J Med*, 351(4):337–45. 21

[79] Flaherty K. T., et al. (2010). Inhibition of mutated, activated BRAF in metastatic melanoma. *N Engl J Med*, 363(9):809–19. 21

[80] Prahallad A., et al. (2012). Unresponsiveness of colon cancer to BRAF(V600E) inhibition through feedback activation of EGFR. *Nature*, 483(7387):100–3. 21

[81] Van Cutsem E., et al. (2011). Cetuximab plus irinotecan, fluorouracil, and leucovorin as first-line treatment for metastatic colorectal cancer: updated analysis of overall survival according to tumor KRAS and BRAF mutation status. *J Clin Oncol*, 29(15):2011–9. 21

[82] Weinstein I. B. (2002). Cancer. Addiction to oncogenes–the Achilles heal of cancer. *Science*, 297(5578):63–4. 21

[83] Kantarjian H., et al. (2002). Hematologic and cytogenetic responses to imatinib mesylate in chronic myelogenous leukemia. *N Engl J Med*, 346(9):645–52. 22

[84] Druker B. J., et al. (2001). Efficacy and safety of a specific inhibitor of the BCR-ABL tyrosine kinase in chronic myeloid leukemia. *N Engl J Med*, 344(14):1031–7. 22

[85] Sharma P., Wagner K., Wolchok J. D., and Allison J. P. (2011). Novel cancer immunotherapy agents with survival benefit: recent successes and next steps. *Nat Rev Cancer*, 11(11):805–12. 22

[86] Schreiber R. D., Old L. J., and Smyth M. J. (2011). Cancer immunoediting: integrating immunity's roles in cancer suppression and promotion. *Science*, 331(6024):1565–70. 22, 118

[87] Galluzzi L., et al. (2014). Classification of current anticancer immunotherapies. *Oncotarget*, 5(24):12472–508. 22, 23

[88] Sidiropoulos K., et al. (2017). Reactome enhanced pathway visualization. *Bioinformatics*, 33(21):3461–3467. 23, 44, 100

[89] Rosenberg S. A., Restifo N. P., Yang J. C., Morgan R. A., and Dudley M. E. (2008). Adoptive cell transfer: a clinical path to effective cancer immunotherapy. *Nat Rev Cancer*, 8(4):299–308. 22

[90] Hinrichs C. S. and Rosenberg S. A. (2014). Exploiting the curative potential of adoptive T-cell therapy for cancer. *Immunol Rev*, 257(1):56–71. 22

[91] Rosenberg S. A., et al. (2011). Durable complete responses in heavily pretreated patients with metastatic melanoma using T-cell transfer immunotherapy. *Clin Cancer Res*, 17(13):4550–7. 22

[92] Sadelain M., Riviere I., and Brentjens R. (2003). Targeting tumours with genetically enhanced T lymphocytes. *Nat Rev Cancer*, 3(1):35–45. 22

[93] Gross G., Waks T., and Eshhar Z. (1989). Expression of immunoglobulin-T-cell receptor chimeric molecules as functional receptors with antibody-type specificity. *Proc Natl Acad Sci U S A*, 86(24):10024–8. 22

[94] McLaughlin P., et al. (1998). Rituximab chimeric anti-CD20 monoclonal antibody therapy for relapsed indolent lymphoma: half of patients respond to a four-dose treatment program. *J Clin Oncol*, 16(8):2825–33. 22

[95] Huhn D., et al. (2001). Rituximab therapy of patients with B-cell chronic lymphocytic leukemia. *Blood*, 98(5):1326–31. 22

[96] Leach D. R., Krummel M. F., and Allison J. P. (1996). Enhancement of antitumor immunity by CTLA-4 blockade. *Science*, 271(5256):1734–6. 23

[97] Okazaki T., Chikuma S., Iwai Y., Fagarasan S., and Honjo T. (2013). A rheostat for immune responses: the unique properties of PD-1 and their advantages for clinical application. *Nat Immunol*, 14(12):1212–8. 23

[98] Melero I., et al. (2015). Evolving synergistic combinations of targeted immunotherapies to combat cancer. *Nat Rev Cancer*, 15(8):457–72. 23

[99] Menard C., Martin F., Apetoh L., Bouyer F., and Ghiringhelli F. (2008). Cancer chemotherapy: not only a direct cytotoxic effect, but also an adjuvant for antitumor immunity. *Cancer Immunol Immunother*, 57(11):1579–87. 23

[100] Vanneman M. and Dranoff G. (2012). Combining immunotherapy and targeted therapies in cancer treatment. *Nat Rev Cancer*, 12(4):237–51. 24

[101] Shinnick T. M., Sutcliffe J. G., Green N., and Lerner R. A. (1983). Synthetic peptide immunogens as vaccines. *Annu Rev Microbiol*, 37:425–46. 25

[102] Sesardic D. (1993). Synthetic peptide vaccines. *J Med Microbiol*, 39(4):241–2.

[103] Sette A., et al. (2001). The development of multi-epitope vaccines: epitope identification, vaccine design and clinical evaluation. *Biologicals*, 29(3-4):271–6. 25

[104] Gouttefangeas C. and Rammensee H. G. (2018). Personalized cancer vaccines: adjuvants are important, too. *Cancer Immunol Immunother*. 25

[105] Blachere N. E., et al. (1997). Heat shock protein-peptide complexes, reconstituted in vitro, elicit peptide-specific cytotoxic T lymphocyte response and tumor immunity. *J Exp Med*, 186(8):1315–22. 25

[106] Aguilar J. C. and Rodriguez E. G. (2007). Vaccine adjuvants revisited. *Vaccine*, 25(19):3752–62. 25

[107] Brown L. E. and Jackson D. C. (2005). Lipid-based self-adjuvanting vaccines. *Curr Drug Deliv*, 2(4):383–93. 25

[108] Dranoff G. (2002). GM-CSF-based cancer vaccines. *Immunol Rev*, 188:147–54. 25

[109] Lee P., et al. (2001). Effects of interleukin-12 on the immune response to a multipeptide vaccine for resected metastatic melanoma. *J Clin Oncol*, 19(18):3836–47. 25

[110] Chianese-Bullock K. A., et al. (2005). MAGE-A1-, MAGE-A10-, and gp100-derived peptides are immunogenic when combined with granulocyte-macrophage colony-stimulating factor and montanide ISA-51 adjuvant and administered as part of a multipeptide vaccine for melanoma. *J Immunol*, 174(5):3080–6. 25

[111] Kool M., et al. (2008). Alum adjuvant boosts adaptive immunity by inducing uric acid and activating inflammatory dendritic cells. *J Exp Med*, 205(4):869–82. 25

[112] Kool M., Fierens K., and Lambrecht B. N. (2012). Alum adjuvant: some of the tricks of the oldest adjuvant. *J Med Microbiol*, 61(Pt 7):927–34. 25

[113] Liang M. T., Davies N. M., Blanchfield J. T., and Toth I. (2006). Particulate systems as adjuvants and carriers for peptide and protein antigens. *Curr Drug Deliv*, 3(4):379–88. 25

[114] Sanders M. T., Brown L. E., Deliyannis G., and Pearse M. J. (2005). ISCOM-based vaccines: the second decade. *Immunol Cell Biol*, 83(2):119–28. 25

[115] Kersten G. F. and Crommelin D. J. (2003). Liposomes and ISCOMs. *Vaccine*, 21(9-10):915–20. 25

[116] Taieb J., Chaput N., and Zitvogel L. (2005). Dendritic cell-derived exosomes as cell-free peptide-based vaccines. *Crit Rev Immunol*, 25(3):215–23. 25

[117] Westerfeld N. and Zurbriggen R. (2005). Peptides delivered by immunostimulating reconstituted influenza virosomes. *J Pept Sci*, 11(11):707–12. 25

[118] Bramwell V. W. and Perrie Y. (2006). Particulate delivery systems for vaccines: what can we expect? *J Pharm Pharmacol*, 58(6):717–28. 25

[119] Palucka K. and Banchereau J. (2013). Dendritic-cell-based therapeutic cancer vaccines. *Immunity*, 39(1):38–48. 25

[120] Purcell A. W., McCluskey J., and Rossjohn J. (2007). More than one reason to rethink the use of peptides in vaccine design. *Nat Rev Drug Discov*, 6(5):404–14. 25

[121] Sette A. and Fikes J. (2003). Epitope-based vaccines: an update on epitope identification, vaccine design and delivery. *Curr Opin Immunol*, 15(4):461–70. 26

[122] Rosenberg S. A. (2001). Progress in human tumour immunology and immunotherapy. *Nature*, 411(6835):380–4. 26

[123] Chaux P., et al. (1999). Identification of five MAGE-A1 epitopes recognized by cytolytic T lymphocytes obtained by in vitro stimulation with dendritic cells transduced with MAGE-A1. *J Immunol*, 163(5):2928–36.

[124] Cox A. L., et al. (1994). Identification of a peptide recognized by five melanoma-specific human cytotoxic T cell lines. *Science*, 264(5159):716–9. 26

[125] Pardoll D. M. and Topalian S. L. (1998). The role of CD4+ T cell responses in antitumor immunity. *Curr Opin Immunol*, 10(5):588–94. 26

[126] Rosenberg S. A. (2005). Cancer immunotherapy comes of age. *Nat Clin Pract Oncol*, 2(3):115. 26

[127] Zhang L., et al. (2003). Intratumoral T cells, recurrence, and survival in epithelial ovarian cancer. *N Engl J Med*, 348(3):203–13. 26

[128] Galon J., et al. (2006). Type, density, and location of immune cells within human colorectal tumors predict clinical outcome. *Science*, 313(5795):1960–4. 26

[129] Rosenberg S. A., Yang J. C., and Restifo N. P. (2004). Cancer immunotherapy: moving beyond current vaccines. *Nat Med*, 10(9):909–15. 26, 27, 98

[130] Yang J., Zhang Q., Li K., Yin H., and Zheng J. N. (2015). Composite peptide-based vaccines for cancer immunotherapy (Review). *Int J Mol Med*, 35(1):17–23. 26

[131] Kawakami Y., et al. (1994). Identification of the immunodominant peptides of the MART-1 human melanoma antigen recognized by the majority of HLA-A2-restricted tumor infiltrating lymphocytes. *J Exp Med*, 180(1):347–52. 26

[132] Bakker A. B., et al. (1994). Melanocyte lineage-specific antigen gp100 is recognized by melanoma-derived tumor-infiltrating lymphocytes. *J Exp Med*, 179(3):1005–9. 26

[133] van der Bruggen P., et al. (1991). A gene encoding an antigen recognized by cytolytic T lymphocytes on a human melanoma. *Science*, 254(5038):1643–7. 26

[134] Chomez P., et al. (2001). An overview of the MAGE gene family with the identification of all human members of the family. *Cancer Res*, 61(14):5544–51. 26

[135] Disis M. L., et al. (2002). Generation of T-cell immunity to the HER-2/neu protein after active immunization with HER-2/neu peptide-based vaccines. *J Clin Oncol*, 20(11):2624–32. 26

[136] Schmidt S. M., et al. (2003). Survivin is a shared tumor-associated antigen expressed in a broad variety of malignancies and recognized by specific cytotoxic T cells. *Blood*, 102(2):571–6. 26

[137] Vonderheide R. H., Hahn W. C., Schultze J. L., and Nadler L. M. (1999). The telomerase catalytic subunit is a widely expressed tumor-associated antigen recognized by cytotoxic T lymphocytes. *Immunity*, 10(6):673–9. 26

[138] Tran E., Robbins P. F., and Rosenberg S. A. (2017). 'Final common pathway' of human cancer immunotherapy: targeting random somatic mutations. *Nat Immunol*, 18(3):255–262. 26

[139] Guo C., et al. (2013). Therapeutic cancer vaccines: past, present, and future. *Adv Cancer Res*, 119:421–75. 26

[140] Singh-Jasuja H., Emmerich N. P., and Rammensee H. G. (2004). The Tübingen approach: identification, selection, and validation of tumor-associated HLA peptides for cancer therapy. *Cancer Immunol Immunother*, 53(3):187–95. 27

[141] Castle J. C., et al. (2012). Exploiting the mutanome for tumor vaccination. *Cancer Res*, 72(5):1081–91. 27, 98

[142] Kreiter S., et al. (2015). Mutant MHC class II epitopes drive therapeutic immune responses to cancer. *Nature*, 520(7549):692–6. 118

[143] Gubin M. M., et al. (2014). Checkpoint blockade cancer immunotherapy targets tumour-specific mutant antigens. *Nature*, 515(7528):577–81. 27, 98

[144] Lu Y. C., et al. (2013). Mutated PPP1R3B is recognized by T cells used to treat a melanoma patient who experienced a durable complete tumor regression. *J Immunol*, 190(12):6034–42. 27

[145] Zhou J., Dudley M. E., Rosenberg S. A., and Robbins P. F. (2005). Persistence of multiple tumor-specific T-cell clones is associated with complete tumor regression in a melanoma patient receiving adoptive cell transfer therapy. *J Immunother*, 28(1):53–62.

[146] Prickett T. D., et al. (2016). Durable complete response from metastatic melanoma after transfer of autologous T cells recognizing 10 mutated tumor antigens. *Cancer Immunol Res*, 4(8):669–78. 27

[147] Matsushita H., et al. (2012). Cancer exome analysis reveals a T-cell-dependent mechanism of cancer immunoediting. *Nature*, 482(7385):400–4. 27, 98

[148] van Rooij N., et al. (2013). Tumor exome analysis reveals neoantigen-specific T-cell reactivity in an ipilimumab-responsive melanoma. *J Clin Oncol*, 31(32):e439–42. 27, 98, 118

[149] Sahin U., et al. (2017). Personalized RNA mutanome vaccines mobilize poly-specific therapeutic immunity against cancer. *Nature*, 547(7662):222–226. 27, 99, 116

[150] Ott P. A., et al. (2017). An immunogenic personal neoantigen vaccine for patients with melanoma. *Nature*, 547(7662):217–221. 27, 99, 116

[151] Sanger F. and Coulson A. R. (1975). A rapid method for determining sequences in DNA by primed synthesis with DNA polymerase. *J Mol Biol*, 94(3):441–8. 28

[152] Maxam A. M. and Gilbert W. (1977). A new method for sequencing DNA. *Proc Natl Acad Sci U S A*, 74(2):560–4. 28

[153] Sanger F., Nicklen S., and Coulson A. R. (1977). DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci U S A*, 74(12):5463–7. 28

[154] Goffeau A., et al. (1996). Life with 6000 genes. *Science*, 274(5287):546, 563–7. 28

[155] Consortium C. e. S. (1998). Genome sequence of the nematode C. elegans: a platform for investigating biology. *Science*, 282(5396):2012–8. 28

[156] Venter J. C., et al. (2001). The sequence of the human genome. *Science*, 291(5507):1304–51. 28

[157] Lander E. S., et al. (2001). Initial sequencing and analysis of the human genome. *Nature*, 409(6822):860–921. 28

[158] International Human Genome Sequencing Consortium (2004). Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–45. 28

[159] Hutchison, C. A. r. (2007). DNA sequencing: bench to bedside and beyond. *Nucleic Acids Res*, 35(18):6227–37. 28

[160] Schloss J. A. (2008). How to get genomes at one ten-thousandth the cost. *Nat Biotechnol*, 26(10):1113–5. 28

[161] Wetterstrand K. A. DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP). `http://www.genome.gov/sequencingcostsdata`. Accessed: 2017-11-06. 28, 29

[162] Metzker M. L. (2010). Sequencing technologies - the next generation. *Nat Rev Genet*, 11(1):31–46. 28, 29

[163] Dressman D., Yan H., Traverso G., Kinzler K. W., and Vogelstein B. (2003). Transforming single DNA molecules into fluorescent magnetic particles for detection and enumeration of genetic variations. *Proc Natl Acad Sci U S A*, 100(15):8817–22. 28

[164] Fedurco M., Romieu A., Williams S., Lawrence I., and Turcatti G. (2006). BTA, a novel reagent for DNA attachment on glass and efficient generation of solid-phase amplified DNA colonies. *Nucleic Acids Res*, 34(3):e22. 28

[165] Bentley D. R., et al. (2008). Accurate whole human genome sequencing using reversible terminator chemistry. *Nature*, 456(7218):53–9. 29, 30

[166] Margulies M., et al. (2005). Genome sequencing in microfabricated high-density picolitre reactors. *Nature*, 437(7057):376–80. 29

[167] Valouev A., et al. (2008). A high-resolution, nucleosome position map of C. elegans reveals a lack of universal sequence-dictated positioning. *Genome Res*, 18(7):1051–63. 29

[168] Rothberg J. M., et al. (2011). An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, 475(7356):348–52. 29

[169] Clarke J., et al. (2009). Continuous base identification for single-molecule nanopore DNA sequencing. *Nat Nanotechnol*, 4(4):265–70. 29

[170] Eid J., et al. (2009). Real-time DNA sequencing from single polymerase molecules. *Science*, 323(5910):133–8. 29

[171] Reuter J. A., Spacek D. V., and Snyder M. P. (2015). High-throughput sequencing technologies. *Mol Cell*, 58(4):586–97. 29, 30

[172] Snyder M., Du J., and Gerstein M. (2010). Personal genome sequencing: current approaches and challenges. *Genes Dev*, 24(5):423–31. 31

[173] Chaisson M. J., et al. (2015). Resolving the complexity of the human genome using single-molecule sequencing. *Nature*, 517(7536):608–11. 31

[174] The 1000 Genomes Project Consortium (2010). A map of human genome variation from population-scale sequencing. *Nature*, 467(7319):1061–73. 31, 48

[175] The 1000 Genomes Project Consortium (2012). An integrated map of genetic variation from 1,092 human genomes. *Nature*, 491(7422):56–65. 31, 48, 66

[176] Wang Z., Gerstein M., and Snyder M. (2009). RNA-Seq: a revolutionary tool for transcriptomics. *Nat Rev Genet*, 10(1):57–63. 31

[177] Human Microbiome Project Consortium (2012). Structure, function and diversity of the healthy human microbiome. *Nature*, 486(7402):207–14. 31, 58

[178] Grossman R. L., et al. (2016). Toward a shared vision for cancer genomic data. *N Engl J Med*, 375(12):1109–12. 31, 39

[179] International Cancer Genome Consortium, et al. (2010). International network of cancer genome projects. *Nature*, 464(7291):993–8. 31

[180] Bao R., et al. (2014). Review of current methods, applications, and data management for the bioinformatics analysis of whole exome sequencing. *Cancer Inform*, 13(Suppl 2):67–82. 31, 32

[181] Cock P. J., Fields C. J., Goto N., Heuer M. L., and Rice P. M. (2010). The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res*, 38(6):1767–71. 32

[182] Ewing B. and Green P. (1998). Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res*, 8(3):186–94. 32

[183] Ewing B., Hillier L., Wendl M. C., and Green P. (1998). Base-calling of automated sequencer traces using phred. I. Accuracy assessment. *Genome Res*, 8(3):175–85. 32

[184] Andrews S. (2010). FastQC: a quality control tool for high throughput sequence data. `https://www.bioinformatics.babraham.ac.uk/projects/fastqc/`. 32, 90, 104

[185] Martin M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet.journal*, 17:10–12. 32, 104

[186] Bolger A. M., Lohse M., and Usadel B. (2014). Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics*, 30(15):2114–20. 32

[187] Sturm M., Schroeder C., and Bauer P. (2016). SeqPurge: highly-sensitive adapter trimming for paired-end NGS data. *BMC Bioinformatics*, 17:208. 32, 103

[188] Needleman S. B. and Wunsch C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J Mol Biol*, 48(3):443–53. 33

[189] Smith T. F. and Waterman M. S. (1981). Identification of common molecular subsequences. *J Mol Biol*, 147(1):195–7. 33

[190] Flicek P. and Birney E. (2009). Sense from sequence reads: methods for alignment and assembly. *Nat Methods*, 6(11 Suppl):S6–S12. 33

[191] Burrows M. and Wheeler D. J. (1994). A block-sorting lossless data compression algorithm. 33

[192] Li H., Ruan J., and Durbin R. (2008). Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res*, 18(11):1851–8. 33

[193] Rumble S. M., et al. (2009). SHRiMP: accurate mapping of short color-space reads. *PLoS Comput Biol*, 5(5):e1000386. 33

[194] Li R., Li Y., Kristiansen K., and Wang J. (2008). SOAP: short oligonucleotide alignment program. *Bioinformatics*, 24(5):713–4. 33

[195] Homer N., Merriman B., and Nelson S. F. (2009). Bfast: an alignment tool for large scale genome resequencing. *PLoS One*, 4(11):e7767. 33

[196] Li H. and Durbin R. (2009). Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–60. 33, 103, 104

[197] Li H. and Durbin R. (2010). Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*, 26(5):589–95. 33

[198] Langmead B., Trapnell C., Pop M., and Salzberg S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol*, 10(3):R25. 33

[199] Langmead B. and Salzberg S. L. (2012). Fast gapped-read alignment with Bowtie 2. *Nat Methods*, 9(4):357–9. 33, 46

[200] Ferragina P. and Manzini G. (2001). An experimental study of an opportunistic index. *Proceedings of the Twelfth Annual Acm-Siam Symposium on Discrete Algorithms*, pages 269–278. 33

[201] Li R., et al. (2009). SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*, 25(15):1966–7. 33

[202] Weese D., Holtgrewe M., and Reinert K. (2012). RazerS 3: faster, fully sensitive read mapping. *Bioinformatics*, 28(20):2592–9. 33, 46

[203] David M., Dzamba M., Lister D., Ilie L., and Brudno M. (2011). SHRiMP2: sensitive yet practical short read mapping. *Bioinformatics*, 27(7):1011–2. 33

[204] Siragusa E., Weese D., and Reinert K. (2013). Fast and accurate read mapping with approximate seeds and multiple backtracking. *Nucleic Acids Res*, 41(7):e78. 33, 54, 85

[205] Engstrom P. G., et al. (2013). Systematic evaluation of spliced alignment programs for RNA-seq data. *Nat Methods*, 10(12):1185–91. 33

[206] Trapnell C., Pachter L., and Salzberg S. L. (2009). TopHat: discovering splice junctions with RNA-Seq. *Bioinformatics*, 25(9):1105–11. 33

[207] Kim D., et al. (2013). TopHat2: accurate alignment of transcriptomes in the presence of insertions, deletions and gene fusions. *Genome Biol*, 14(4):R36. 33, 104, 118, 197

[208] Dobin A., et al. (2013). STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, 29(1):15–21. 33, 104, 118

[209] Li H., et al. (2009). The Sequence Alignment/Map format and SAMtools. *Bioinformatics*, 25(16):2078–9. 34, 104

[210] McKenna A., et al. (2010). The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res*, 20(9):1297–303. 34

[211] DePristo M. A., et al. (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat Genet*, 43(5):491–8. 34

[212] Garrison E. and Marth G. (2012). Haplotype-based variant detection from short-read sequencing. *arXiv preprint*, q-bio.GN(ArXiv:1207.3907). 34, 201

[213] Marth G. T., et al. (1999). A general approach to single-nucleotide polymorphism discovery. *Nat Genet*, 23(4):452–6. 34

[214] Larson D. E., Abbott T. E., and Wilson R. K. (2014). Using SomaticSniper to Detect Somatic Single Nucleotide Variants. *Curr Protoc Bioinformatics*, 45:15 5 1–8. 34

[215] Cibulskis K., et al. (2013). Sensitive detection of somatic point mutations in impure and heterogeneous cancer samples. *Nat Biotechnol*, 31(3):213–9. 34

[216] Koboldt D. C., et al. (2012). VarScan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. *Genome Res*, 22(3):568–76. 34

[217] Sandmann S., et al. (2017). Evaluating variant calling tools for non-matched next-generation sequencing data. *Sci Rep*, 7:43169. 35

[218] Alioto T. S., et al. (2015). A comprehensive assessment of somatic mutation detection in cancer using whole-genome sequencing. *Nat Commun*, 6:10001. 35

[219] Danecek P., et al. (2011). The variant call format and VCFtools. *Bioinformatics*, 27(15):2156–8. 35

[220] Wang K., Li M., and Hakonarson H. (2010). ANNOVAR: functional annotation of genetic variants from high-throughput sequencing data. *Nucleic Acids Res*, 38(16):e164. 35, 85, 200

[221] Cingolani P., et al. (2012). A program for annotating and predicting the effects of single nucleotide polymorphisms, SnpEff: SNPs in the genome of Drosophila melanogaster strain w1118; iso-2; iso-3. *Fly*, 6(2):80–92. 35, 85, 104, 201

[222] McLaren W., et al. (2010). Deriving the consequences of genomic variants with the Ensembl API and SNP Effect Predictor. *Bioinformatics*, 26(16):2069–70. 35

[223] Sherry S. T., et al. (2001). dbSNP: the NCBI database of genetic variation. *Nucleic Acids Res*, 29(1):308–11. 35

[224] Landrum M. J., et al. (2014). ClinVar: public archive of relationships among sequence variation and human phenotype. *Nucleic Acids Res*, 42(Database issue):D980–5. 35

[225] Forbes S. A., et al. (2011). COSMIC: mining complete cancer genomes in the Catalogue of Somatic Mutations in Cancer. *Nucleic Acids Res*, 39(Database issue):D945–50. 35

[226] Conesa A., et al. (2016). A survey of best practices for RNA-seq data analysis. *Genome Biol*, 17:13. 35

[227] Anders S., Pyl P. T., and Huber W. (2015). HTSeq-a Python framework to work with high-throughput sequencing data. *Bioinformatics*, 31(2):166–9. 35, 104, 197

[228] Bray N. L., Pimentel H., Melsted P., and Pachter L. (2016). Near-optimal probabilistic RNA-Seq quantification. *Nat Biotechnol*, 34(5):525–7. 35, 118

[229] Mortazavi A., Williams B. A., McCue K., Schaeffer L., and Wold B. (2008). Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Methods*, 5(7):621–8. 35

[230] Wagner G. P., Kin K., and Lynch V. J. (2012). Measurement of mRNA abundance using RNA-seq data: RPKM measure is inconsistent among samples. *Theory Biosci*, 131(4):281–5. 36

[231] Love M. I., Huber W., and Anders S. (2014). Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol*, 15(12):550. 36, 180

[232] Robinson M. D., McCarthy D. J., and Smyth G. K. (2010). edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–40. 36

[233] Robinson M. D. and Oshlack A. (2010). A scaling normalization method for differential expression analysis of RNA-seq data. *Genome Biol*, 11(3):R25. 36

[234] Kesmir C., Nussbaum A. K., Schild H., Detours V., and Brunak S. (2002). Prediction of proteasome cleavage motifs by neural networks. *Protein Eng*, 15(4):287–96. 36

[235] Nielsen M., Lundegaard C., Lund O., and Kesmir C. (2005). The role of the proteasome in generating cytotoxic T-cell epitopes: insights obtained from improved predictions of proteasomal cleavage. *Immunogenetics*, 57(1-2):33–41. 36, 57, 59

[236] Daniel S., et al. (1998). Relationship between peptide selectivities of human transporters associated with antigen processing and HLA class I molecules. *J Immunol*, 161(2):617–24. 37

[237] Peters B., Bulik S., Tampe R., Van Endert P. M., and Holzhutter H. G. (2003). Identifying MHC class I epitopes by predicting the TAP transport efficiency of epitope precursors. *J Immunol*, 171(4):1741–9. 37, 59

[238] Rammensee H., Bachmann J., Emmerich N. P., Bachor O. A., and Stevanović S. (1999). SYFPEITHI: database for MHC ligands and peptide motifs. *Immunogenetics*, 50(3-4):213–9. 37, 60, 98, 106

[239] Nielsen M. and Lund O. (2009). NN-align. An artificial neural network-based alignment algorithm for MHC class II peptide binding prediction. *BMC Bioinformatics*, 10:296. 37

[240] Nielsen M., Lundegaard C., and Lund O. (2007). Prediction of MHC class II binding affinity using SMM-align, a novel stabilization matrix alignment method. *BMC Bioinformatics*, 8:238. 37

[241] Andreatta M., et al. (2015). Accurate pan-specific prediction of peptide-MHC class II binding affinity with improved binding core identification. *Immunogenetics*, 67(11-12):641–50. 37

[242] Parker K. C., Bednarek M. A., and Coligan J. E. (1994). Scheme for ranking potential HLA-A2 binding peptides based on independent binding of individual peptide side-chains. *J Immunol*, 152(1):163–75. 37

[243] Donnes P. and Elofsson A. (2002). Prediction of MHC class I binding peptides, using SVMHC. *BMC Bioinformatics*, 3:25. 37

[244] Nielsen M., et al. (2003). Reliable prediction of T-cell epitopes using neural networks with novel sequence representations. *Protein Sci*, 12(5):1007–17. 37, 106

[245] Andreatta M. and Nielsen M. (2016). Gapped sequence alignment using artificial neural networks: application to the MHC class I system. *Bioinformatics*, 32(4):511–7. 37, 98, 106

[246] Hoof I., et al. (2009). NetMHCpan, a method for MHC class I binding prediction beyond humans. *Immunogenetics*, 61(1):1–13. 37

[247] Nielsen M. and Andreatta M. (2016). NetMHCpan-3.0; improved prediction of binding to MHC class I molecules integrating information from multiple receptor and peptide length datasets. *Genome Med*, 8(1):33. 37, 58, 59, 106

[248] Jurtz V., et al. (2017). NetMHCpan-4.0: Improved Peptide-MHC Class I Interaction Predictions Integrating Eluted Ligand and Peptide Binding Affinity Data. *J Immunol*, 199(9):3360–3368. 37, 98

[249] Karosiene E., Lundegaard C., Lund O., and Nielsen M. (2012). NetMHCcons: a consensus method for the major histocompatibility complex class I predictions. *Immunogenetics*, 64(3):177–86. 37

[250] Zhang H., Lund O., and Nielsen M. (2009). The PickPocket method for predicting binding specificities for receptors based on receptor pocket similarities: application to MHC-peptide binding. *Bioinformatics*, 25(10):1293–9. 37

[251] Vita R., et al. (2015). The immune epitope database (IEDB) 3.0. *Nucleic Acids Res*, 43(Database issue):D405–12. 37, 112

[252] Vider-Shalit T., Raffaeli S., and Louzoun Y. (2007). Virus-epitope vaccine design: informatic matching the HLA-I polymorphism to the virus genome. *Mol Immunol*, 44(6):1253–61. 37

[253] Toussaint N. C., Donnes P., and Kohlbacher O. (2008). A mathematical framework for the selection of an optimal set of peptides for epitope-based vaccines. *PLoS Comput Biol*, 4(12):e1000246. 37

[254] Lundegaard C., et al. (2010). PopCover: a method for selecting of peptides with optimal population and pathogen coverage. *In Proceedings of the First ACM International Conference on Bioinformatics and Computational Biology*, pages 658–659. 37

[255] Toussaint N. C., Maman Y., Kohlbacher O., and Louzoun Y. (2011). Universal peptide vaccines–optimal peptide vaccine design based on viral sequence conservation. *Vaccine*, 29(47):8745–53. 38

[256] Schubert B. and Kohlbacher O. (2016). Designing string-of-beads vaccines with optimal spacers. *Genome Med*, 8(1):9. 38

[257] Toussaint N. C. and Kohlbacher O. (2009). OptiTope–a web server for the selection of an optimal set of peptides for epitope-based vaccines. *Nucleic Acids Res*, 37(Web Server issue):W617–22. 38

[258] Schubert B., Brachvogel H. P., Jurges C., and Kohlbacher O. (2015). EpiToolKit–a web-based workbench for vaccine design. *Bioinformatics*, 31(13):2211–3. 38

[259] Feldhahn M., et al. (2008). EpiToolKit–a web server for computational immunomics. *Nucleic Acids Res*, 36(Web Server issue):W519–22. 38

[260] Schubert B., de la Garza L., Mohr C., Walzer M., and Kohlbacher O. (2017). ImmunoNodes–graphical development of complex immunoinformatics workflows. *BMC Bioinformatics*, 18(1):242. 38, 61, 64

[261] Ioannidis J. P., et al. (2009). Repeatability of published microarray gene expression analyses. *Nat Genet*, 41(2):149–55. 38

[262] Nekrutenko A. and Taylor J. (2012). Next-generation sequencing data interpretation: enhancing reproducibility and accessibility. *Nature Reviews Genetics*, 13(9):667–U93. 38, 95

[263] Blankenberg D., et al. (2010). Galaxy: a web–based genome analysis tool for experimentalists. *Curr Protoc Mol Biol*, Chapter 19:Unit 19 10 1–21. 38

[264] Goecks J., Nekrutenko A., Taylor J., and Galaxy T. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol*, 11(8):R86. 38, 66, 92

[265] Afgan E., Chapman B., and Taylor J. (2012). CloudMan as a platform for tool, data, and analysis distribution. *BMC Bioinformatics*, 13:315. 38

[266] Blankenberg D., et al. (2014). Dissemination of scientific software with Galaxy ToolShed. *Genome Biol*, 15(2):403. 38

[267] Oinn T., et al. (2004). Taverna: a tool for the composition and enactment of bioinformatics workflows. *Bioinformatics*, 20(17):3045–54. 38

[268] Wolstencroft K., et al. (2013). The Taverna workflow suite: designing and executing workflows of Web Services on the desktop, web or in the cloud. *Nucleic Acids Res*, 41(Web Server issue):W557–61. 38

[269] Goble C. A., et al. (2010). myExperiment: a repository and social network for the sharing of bioinformatics workflows. *Nucleic Acids Res*, 38(Web Server issue):W677–82. 38

[270] Amstutz P., et al. (2016). Common Workflow Language, v1.0. *Specification, Common Workflow Language working group*. 38

[271] Berthold M. R., et al. (2008). KNIME: The Konstanz Information Miner. *Data Analysis, Machine Learning and Applications*, pages 319–326. 38, 61

[272] Kacsuk P., et al. (2012). WS-PGRADE/gUSE Generic DCI Gateway Framework for a Large Variety of User Communities. *Journal of Grid Computing*, 10(4):601–630. 38, 66, 76

[273] Koster J. and Rahmann S. (2012). Snakemake–a scalable bioinformatics workflow engine. *Bioinformatics*, 28(19):2520–2. 39, 77, 94

[274] McGrath R. M. S. and Roland (1991). GNU Make–a program for directing recompilation. 39

[275] Docker Inc. Docker. `https://www.docker.com/`. 39, 88

[276] Kurtzer G. M., Sochat V., and Bauer M. W. (2017). Singularity: Scientific containers for mobility of compute. *PLoS One*, 12(5):e0177459. 39, 88, 123

[277] Di Tommaso P., et al. (2017). Nextflow enables reproducible computational workflows. *Nat Biotechnol*, 35(4):316–319. 39, 77, 94, 123

[278] Haider S., et al. (2009). BioMart Central Portal–unified access to biological data. *Nucleic Acids Res*, 37(Web Server issue):W23–7. 39

[279] Zhang J., et al. (2011). International Cancer Genome Consortium Data Portal–a one-stop shop for cancer genomics data. *Database (Oxford)*, 2011:bar026. 39

[280] Cerami E., et al. (2012). The cBio cancer genomics portal: an open platform for exploring multidimensional cancer genomics data. *Cancer Discov*, 2(5):401–4. 39, 67

[281] Apache Software Foundation. Apache Tomcat. `https://tomcat.apache.org/`. 39, 70

[282] Oracle Corporation. GlassFish. `https://javaee.github.io/glassfish/`. 39

[283] Red Hat Inc. WildFly. `https://wildfly.org/`. 39

[284] Liferay Inc. Liferay Portal. `https://web.liferay.com/community/liferay-projects/liferay-portal/overview`. 39, 70, 94

[285] Red Hat Inc. GateIn JBoss Portal. `https://gatein.jboss.org/`. 39

[286] Haralambieva I. H., et al. (2013). The genetic basis for interindividual immune response variation to measles vaccine: new understanding and new vaccine approaches. *Expert Rev Vaccines*, 12(1):57–70. 41

[287] Ovsyannikova I. G. and Poland G. A. (2011). Vaccinomics: current findings, challenges and novel approaches for vaccine development. *AAPS J*, 13(3):438–44. 41

[288] Bradley B. A. (1991). The role of HLA matching in transplantation. *Immunol Lett*, 29(1-2):55–9. 41

[289] Opelz G., Wujciak T., Dohler B., Scherer S., and Mytilineos J. (1999). HLA compatibility and organ transplant survival. Collaborative Transplant Study. *Rev Immunogenet*, 1(3):334–42. 41

[290] Thorsby E. and Lie B. A. (2005). HLA associated genetic predisposition to autoimmune diseases: Genes involved and possible mechanisms. *Transpl Immunol*, 14(3-4):175–82. 41

[291] Undlien D. E., Lie B. A., and Thorsby E. (2001). HLA complex genes in type 1 diabetes and other autoimmune diseases. Which genes are involved? *Trends Genet*, 17(2):93–100. 41

[292] Robinson J., Soormally A. R., Hayhurst J. D., and Marsh S. G. E. (2016). The IPD-IMGT/HLA Database - New developments in reporting HLA variation. *Hum Immunol*, 77(3):233–237. 42, 44

[293] Liu C., et al. (2013). ATHLATES: accurate typing of human leukocyte antigen through exome sequencing. *Nucleic Acids Res*, 41(14):e142. 42, 43, 49

[294] Gabriel C., et al. (2009). Rapid high-throughput human leukocyte antigen typing by massively parallel pyrosequencing for high-resolution allele identification. *Hum Immunol*, 70(11):960–4. 42

[295] Bentley G., et al. (2009). High-resolution, high-throughput HLA genotyping by next-generation sequencing. *Tissue Antigens*, 74(5):393–403. 42

[296] Lank S. M., et al. (2012). Ultra-high resolution HLA genotyping and allele discovery by highly multiplexed cDNA amplicon pyrosequencing. *BMC Genomics*, 13:378. 42

[297] Lank S. M., Wiseman R. W., Dudley D. M., and O'Connor D. H. (2010). A novel single cDNA amplicon pyrosequencing method for high-throughput, cost-effective sequence-based HLA class I genotyping. *Hum Immunol*, 71(10):1011–7.

[298] Moonsamy P. V., et al. (2013). High throughput HLA genotyping using 454 sequencing and the Fluidigm Access Array System for simplified amplicon library preparation. *Tissue Antigens*, 81(3):141–9. 42

[299] Danzer M., et al. (2013). Rapid, scalable and highly automated HLA genotyping using next-generation sequencing: a transition from research to diagnostics. *BMC Genomics*, 14:221. 43

[300] Erlich R. L., et al. (2011). Next-generation sequencing for HLA typing of class I loci. *BMC Genomics*, 12:42. 43, 52

[301] Szolek A., et al. (2014). OptiType: precision HLA typing from next-generation sequencing data. *Bioinformatics*, 30(23):3310–6. 44, 50, 51, 91, 159, 183, 184

[302] Karp R. M. (1972). Reducibility among combinatorial problems. *Complexity of Computer Computations*, pages 85–103. 45

[303] Robinson J., et al. (2013). The IMGT/HLA database. *Nucleic Acids Res*, 41(Database issue):D1222–7. 45

[304] Blasczyk R., Kotsch K., and Wehling J. (1997). The nature of polymorphism of the HLA class I non-coding regions and their contribution to the diversification of HLA. *Hereditas*, 127(1-2):7–9. 45

[305] Sievers F., et al. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using Clustal Omega. *Mol Syst Biol*, 7:539. 46

[306] Doring A., Weese D., Rausch T., and Reinert K. (2008). SeqAn an efficient, generic C++ library for sequence analysis. *BMC Bioinformatics*, 9:11. 46

[307] Gonzalez-Galarza F. F., Christmas S., Middleton D., and Jones A. R. (2011). Allele frequency net: a database and online repository for immune gene frequencies in worldwide populations. *Nucleic Acids Res*, 39(Database issue):D913–9. 46

[308] Coordinators Ncbi Resource (2018). Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res*, 46(D1):D8–D13. 46, 48

[309] Schrijver A. *Theory of linear and integer programming*. Wiley-Interscience series in discrete mathematics and optimization. Wiley (1998). 47

[310] International HapMap Consortium (2005). A haplotype map of the human genome. *Nature*, 437(7063):1299–320. 48

[311] Montgomery S. B., et al. (2010). Transcriptome genetics using second generation sequencing in a Caucasian population. *Nature*, 464(7289):773–7. 48

[312] Leinonen R., et al. (2011). The European Nucleotide Archive. *Nucleic Acids Res*, 39(Database issue):D28–31. 48

[313] Huang Y., et al. (2015). HLAreporter: a tool for HLA typing from next generation sequencing data. *Genome Med*, 7(1):25. 52

[314] Shukla S. A., et al. (2015). Comprehensive analysis of cancer-associated somatic mutations in class I HLA genes. *Nat Biotechnol*, 33(11):1152–8. 53

[315] Nariai N., et al. (2015). HLA-VBSeq: accurate HLA typing at full resolution from whole-genome sequencing data. *BMC Genomics*, 16 Suppl 2:S7.

[316] Dilthey A. T., et al. (2016). High-accuracy HLA type inference from whole-genome sequencing data using population reference graphs. *PLoS Comput Biol*, 12(10):e1005151. 52

[317] Kawaguchi S., Higasa K., Shimizu M., Yamada R., and Matsuda F. (2017). HLA-HD: An accurate HLA typing algorithm for next-generation sequencing data. *Hum Mutat*, 38(7):788–797. 53

[318] Xie C., et al. (2017). Fast and accurate HLA typing from short-read next-generation sequence data with xHLA. *Proc Natl Acad Sci U S A*, 114(30):8059–8064. 52, 53, 122

[319] Kiyotani K., Mai T. H., and Nakamura Y. (2017). Comparison of exome-based HLA class I genotyping tools: identification of platform-specific genotyping errors. *J Hum Genet*, 62(3):397–405. 52, 53

[320] Bauer D. C., Zadoorian A., Wilson L. O. W., Melbourne Genomics Health Alliance, and Thorne N. P. (2018). Evaluation of computational programs to predict HLA genotypes from genomic sequencing data. *Brief Bioinform*, 19(2):179–187. 52, 53

[321] Yewdell J. W. and Bennink J. R. (1999). Immunodominance in major histocompatibility complex class I-restricted T lymphocyte responses. *Annu Rev Immunol*, 17:51–88. 55

[322] Calis J. J., et al. (2013). Properties of MHC class I presented peptides that enhance immunogenicity. *PLoS Comput Biol*, 9(10):e1003266. 55, 56, 59

[323] Josefowicz S. Z. and Rudensky A. (2009). Control of regulatory T cell lineage commitment and maintenance. *Immunity*, 30(5):616–25. 55

[324] Calis J. J., de Boer R. J., and Kesmir C. (2012). Degenerate T-cell recognition of peptides on MHC molecules creates large holes in the T-cell repertoire. *PLoS Comput Biol*, 8(3):e1002412. 55

[325] Bresciani A., et al. (2016). T-cell recognition is shaped by epitope sequence conservation in the host proteome and microbiome. *Immunology*, 148(1):34–9. 55

[326] Schreiber T. H., Raez L., Rosenblatt J. D., and Podack E. R. (2010). Tumor immunogenicity and responsiveness to cancer vaccine therapy: the state of the art. *Semin Immunol*, 22(3):105–12. 56

[327] Schellekens H. (2002). Bioequivalence and the immunogenicity of biopharmaceuticals. *Nat Rev Drug Discov*, 1(6):457–62. 56

[328] Hu Z., Ott P. A., and Wu C. J. (2018). Towards personalized, tumour-specific, therapeutic vaccines for cancer. *Nat Rev Immunol*, 18(3):168–182. 56

[329] Backert L. and Kohlbacher O. (2015). Immunoinformatics and epitope prediction in the age of genomic medicine. *Genome Med*, 7:119. 56

[330] Ochoa-Garay J., McKinney D. M., Kochounian H. H., and McMillan M. (1997). The ability of peptides to induce cytotoxic T cells in vitro does not strongly correlate with their affinity for the H-2Ld molecule: implications for vaccine design and immunotherapy. *Mol Immunol*, 34(3):273–81. 56

[331] Bihl F., et al. (2006). Impact of HLA-B alleles, epitope binding affinity, functional avidity, and viral coinfection on the immunodominance of virus-specific CTL responses. *J Immunol*, 176(7):4094–101. 56

[332] Feltkamp M. C., Vierboom M. P., Kast W. M., and Melief C. J. (1994). Efficient MHC class I-peptide binding is required but does not ensure MHC class I-restricted immunogenicity. *Mol Immunol*, 31(18):1391–401. 56

[333] De Boer R. J. and Perelson A. S. (1994). T cell repertoires and competitive exclusion. *J Theor Biol*, 169(4):375–90. 56

[334] Wucherpfennig K. W., Call M. J., Deng L., and Mariuzza R. (2009). Structural alterations in peptide-MHC recognition by self-reactive T cell receptors. *Curr Opin Immunol*, 21(6):590–5.

[335] Rudolph M. G., Stanfield R. L., and Wilson I. A. (2006). How TCRs bind MHCs, peptides, and coreceptors. *Annu Rev Immunol*, 24:419–66.

[336] Hausmann S., et al. (1999). Peptide recognition by two HLA-A2/Tax11-19-specific T cell clones in relationship to their MHC/peptide/TCR crystal structures. *J Immunol*, 162(9):5389–97. 56

[337] Jorgensen K. W., Rasmussen M., Buus S., and Nielsen M. (2014). NetMHCstab – predicting stability of peptide-MHC-I complexes; impacts for cytotoxic T lymphocyte epitope discovery. *Immunology*, 141(1):18–26. 56

[338] Rasmussen M., et al. (2016). Pan-specific prediction of peptide-MHC class I complex stability, a correlate of T cell immunogenicity. *J Immunol*, 197(4):1517–24. 56, 59

[339] van der Burg S. H., Visseren M. J., Brandt R. M., Kast W. M., and Melief C. J. (1996). Immunogenicity of peptides bound to MHC class I molecules depends on the MHC-peptide complex stability. *J Immunol*, 156(9):3308–14. 56

[340] Lazarski C. A., et al. (2005). The kinetic stability of MHC class II:peptide complexes is a key parameter that dictates immunodominance. *Immunity*, 23(1):29–40.

[341] Busch D. H. and Pamer E. G. (1998). MHC class I/peptide stability: implications for immunodominance, in vitro proliferation, and diversity of responding CTL. *J Immunol*, 160(9):4441–8. 56

[342] Harndahl M., et al. (2012). Peptide-MHC class I stability is a better predictor than peptide affinity of CTL immunogenicity. *Eur J Immunol*, 42(6):1405–16. 56

[343] Fredkin E. (1960). Trie memory. *Communications of the ACM*, 3(9):490–499. 56, 59

[344] The UniProt C. (2017). UniProt: the universal protein knowledgebase. *Nucleic Acids Res*, 45(D1):D158–D169. 58

[345] Toussaint N. C. *New approaches to in silico design of epitope-based vaccines*. PhD thesis, Eberhard Karls Universität Tübingen (2011). 58, 62

[346] Methé B. A., et al. (2012). A framework for human microbiome research. *Nature*, 486(7402):215–21. 58

[347] Arumugam M., et al. (2011). Enterotypes of the human gut microbiome. *Nature*, 473(7346):174–80. 58

[348] Henikoff S. and Henikoff J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A*, 89(22):10915–9. 59

[349] Sonnenburg S., et al. (2010). The SHOGUN machine learning toolbox. *J Mach Learn Res*, 11:1799–1802. 60

[350] Carrasco Pro S., et al. (2018). Microbiota epitope similarity either dampens or enhances the immunogenicity of disease-associated antigenic epitopes. *PLoS One*, 13(5):e0196551. 63

[351] Bjerregaard A. M., et al. (2017). An analysis of natural T cell responses to predicted tumor neoepitopes. *Front Immunol*, 8:1566. 63

[352] Shen W.-J., Wong H.-S., Xiao Q.-W., Guo X., and Smale S. (2012). Towards a mathematical foundation of immunology and amino acid chains. *arXiv preprint*, arXiv:1205.6031. 63

[353] Bjerregaard A. M., Nielsen M., Hadrup S. R., Szallasi Z., and Eklund A. C. (2017). MuPeXI: prediction of neo-epitopes from tumor sequencing data. *Cancer Immunol Immunother*, 66(9):1123–1130. 63

[354] Routy B., et al. (2018). Gut microbiome influences efficacy of PD-1-based immunotherapy against epithelial tumors. *Science*, 359(6371):91–97. 63

[355] Matson V., et al. (2018). The commensal microbiome is associated with anti-PD-1 efficacy in metastatic melanoma patients. *Science*, 359(6371):104–108. 63

[356] Belkaid Y. and Segre J. A. (2014). Dialogue between skin microbiota and immunity. *Science*, 346(6212):954–9. 64

[357] Dash P., et al. (2017). Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature*, 547(7661):89–93. 64

[358] Glanville J., et al. (2017). Identifying specificity groups in the T cell receptor repertoire. *Nature*, 547(7661):94–98. 64

[359] Benichou J., Ben-Hamo R., Louzoun Y., and Efroni S. (2012). Rep-Seq: uncovering the immunological repertoire through next-generation sequencing. *Immunology*, 135(3):183–91. 64

[360] Shugay M., et al. (2018). VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Res*, 46(D1):D419–D427. 64

[361] Illumina Inc. Illumina NovaSeq. `https://www.illumina.com/systems/sequencing-platforms/novaseq.html`. 65

[362] Koboldt D. C., Steinberg K. M., Larson D. E., Wilson R. K., and Mardis E. R. (2013). The next-generation sequencing revolution and its impact on genomics. *Cell*, 155(1):27–38. 65

[363] Joly Y., Dove E. S., Knoppers B. M., Bobrow M., and Chalmers D. (2012). Data sharing in the post-genomic world: the experience of the International Cancer Genome Consortium (ICGC) Data Access Compliance Office (DACO). *PLoS Comput Biol*, 8(7):e1002549. 66, 124

[364] Cancer Genome Atlas Research Network, et al. (2013). The Cancer Genome Atlas Pan-Cancer analysis project. *Nat Genet*, 45(10):1113–20. 66, 124

[365] Costa F. F. (2014). Big data in biomedicine. *Drug Discov Today*, 19(4):433–40. 66

[366] Shah N., et al. (2016). A crowdsourcing approach for reusing and meta-analyzing gene expression data. *Nat Biotechnol*, 34(8):803–6. 66

[367] Kunszt P., et al. (2015). iPortal: the swiss grid proteomics portal: Requirements and new features based on experience and usability considerations. *Concurrency and Computation-Practice & Experience*, 27(2):433–445. 66

[368] Quandt A., et al. (2007). Grid-based analysis of tandem mass spectrometry data in clinical proteomics. *Stud Health Technol Inform*, 126:13–22. 66

[369] Quandt A., et al. (2009). SwissPIT: An workflow-based platform for analyzing tandem-MS spectra using the Grid. *Proteomics*, 9(10):2648–55. 66

[370] Scholtalbers J., et al. (2013). Galaxy LIMS for next-generation sequencing. *Bioinformatics*, 29(9):1233–4. 67

[371] Madduri R. K., et al. (2014). Experiences building Globus Genomics: a next-generation sequencing analysis service using Galaxy, Globus, and Amazon Web Services. *Concurr Comput*, 26(13):2266–2279. 67

[372] Krefting D., et al. (2009). MediGRID: Towards a user friendly secured grid infrastructure. *Future Generation Computer Systems-the International Journal of Grid Computing-Theory Methods and Applications*, 25(3):326–336. 67

[373] Kruger J., et al. (2014). The MoSGrid science gateway - a complete solution for molecular simulations. *J Chem Theory Comput*, 10(6):2232–45. 67

[374] Shahand S., et al. (2013). A data-centric science gateway for computational neuroscience. *In IWSG*. 67

[375] Miller M. A., et al. (2015). A RESTful API for cccess to phylogenetic tools via the CIPRES science gateway. *Evol Bioinform Online*, 11:43–8. 67

[376] Bauch A., et al. (2011). openBIS: a flexible framework for managing and analyzing complex data in biology research. *BMC Bioinformatics*, 12:468. 67, 72, 75

[377] Clarke L., et al. (2017). The international Genome sample resource (IGSR): A worldwide collection of genome variation incorporating the 1000 Genomes Project data. *Nucleic Acids Res*, 45(D1):D854–D859. 70

[378] The 1000 Genomes Project Consortium, et al. (2015). A global reference for human genetic variation. *Nature*, 526(7571):68–74. 70, 89

[379] Mohr C., et al. (2018). qPortal: A platform for data-driven biomedical research. *PLoS One*, 13(1):e0191603. 71, 159

[380] Vaadin. Vaadin Framework. `https://vaadin.com/framework`. 70

[381] GNU General Public License, version 3. `https://www.gnu.org/licenses/gpl.html`. 71

[382] Friedrich A., Kenar E., Kohlbacher O., and Nahnsen S. (2015). Intuitive web-based experimental design for high-throughput biomedical data. *Biomed Res Int*, 2015:958302. 71

[383] The PostgreSQL Global Development Group. PostgreSQL. `https://www.postgresql.org/`. 72

[384] Ramakrishnan C., et al. (2014). openBEB: open biological experiment browser for correlative measurements. *BMC Bioinformatics*, 15:84. 72

[385] De Clercq J. (2002). Single sign-on architectures. *Infrastucture Security, Proceedings*, 2437:40–58. 74

[386] Murri R., Kunszt P. Z., Maffioletti S., and Tschopp V. (2011). GridCertLib: A single sign-on solution for grid web applications and portals. *Journal of Grid Computing*, 9(4):441–453. 74

[387] Atlassian. Crowd. `https://www.atlassian.com/software/crowd`. 74

[388] Tridgell A. and Mackerras P. (1996). The rsync algorithm. *ANU Research Publications*. 75

[389] Sturm M., et al. (2008). OpenMS–an open-source software framework for mass spectrometry. *BMC Bioinformatics*, 9:163. 78

[390] de la Garza L., et al. (2016). From the desktop to the grid: scalable bioinformatics via workflow conversion. *BMC Bioinformatics*, 17:127. 78

[391] Smedley D., et al. (2009). BioMart–biological queries made easy. *BMC Genomics*, 10:22. 87

[392] Zerbino D. R., et al. (2018). Ensembl 2018. *Nucleic Acids Res*, 46(D1):D754–D761. 87

[393] H. L. (2013). Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. *arXiv preprint*, arXiv:1303.3997v1. 91

[394] Abdelnur A. and Hepper S. (2003). JSR 168: Portlet specification. *Java Specification Requests, Java Community Process, Sun Microsystems and IBM*, 15. 94

[395] Fleury M. and Reverbel F. (2003). The JBoss extensible server. *Middleware 2003, Proceedings*, 2672:344–373. 94

[396] Piccolo S. R. and Frampton M. B. (2016). Tools and techniques for computational reproducibility. *Gigascience*, 5. 95

[397] Garijo D., et al. (2013). Quantifying reproducibility in computational biology: The case of the tuberculosis drugome. *Plos One*, 8(11). 95

[398] Leek J. T. and Peng R. D. (2015). Opinion: Reproducible research can still be wrong: Adopting a prevention approach. *Proceedings of the National Academy of Sciences of the United States of America*, 112(6):1645–1646. 95

[399] Collins F. S. and Tabak L. A. (2014). NIH plans to enhance reproducibility. *Nature*, 505(7485):612–613. 95

[400] Leek J. T. and Jager L. R. (2017). Is most published research really false? *Annual Review of Statistics and Its Application, Vol 4*, 4:109–122. 95

[401] Wilkinson M. D., et al. (2016). Comment: The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data*, 3. 95

[402] Hundal J., et al. (2016). pVAC-Seq: A genome-guided in silico approach to identifying tumor neoantigens. *Genome Med*, 8(1):11. 95

[403] Decker W. K., et al. (2017). Cancer immunotherapy: Historical perspective of a clinical revolution and emerging preclinical animal models. *Front Immunol*, 8:829. 97

[404] Schadendorf D., et al. (2015). Pooled analysis of long-term survival data from phase II and phase III trials of Ipilimumab in unresectable or metastatic melanoma. *J Clin Oncol*, 33(17):1889–94. 97

[405] Larkin J., et al. (2015). Combined Nivolumab and Ipilimumab or monotherapy in untreated melanoma. *N Engl J Med*, 373(1):23–34. 97

[406] Alexandrov L. B., et al. (2013). Signatures of mutational processes in human cancer. *Nature*, 500(7463):415–21. 98

[407] Vogelstein B., et al. (2013). Cancer genome landscapes. *Science*, 339(6127):1546–58. 98, 116, 117

[408] Robbins P. F., et al. (2013). Mining exomic sequencing data to identify mutated antigens recognized by adoptively transferred tumor-reactive T cells. *Nat Med*, 19(6):747–52. 98

[409] Schumacher T., et al. (2014). A vaccine targeting mutant IDH1 induces antitumour immunity. *Nature*, 512(7514):324–7. 98

[410] Lawrence M. S., et al. (2013). Mutational heterogeneity in cancer and the search for new cancer-associated genes. *Nature*, 499(7457):214–218. 98

[411] Yadav M., et al. (2014). Predicting immunogenic tumour mutations by combining mass spectrometry and exome sequencing. *Nature*, 515(7528):572–6. 98

[412] Kalaora S., et al. (2016). Use of HLA peptidomics and whole exome sequencing to identify human immunogenic neo-antigens. *Oncotarget*, 7(5):5110–7. 98

[413] Tran E., et al. (2015). Immunogenicity of somatic mutations in human gastrointestinal cancers. *Science*, 350(6266):1387–90. 99, 116

[414] Bassani-Sternberg M., et al. (2016). Direct identification of clinically relevant neoepitopes presented on native human melanoma tissue by mass spectrometry. *Nat Commun*, 7:13404. 99, 113, 114, 117, 123

[415] Chang T. C., et al. (2017). The neoepitope landscape in pediatric cancers. *Genome Med*, 9(1):78. 99

[416] Rubinsteyn A., et al. (2017). Computational pipeline for the PGV-001 neoantigen vaccine trial. *Front Immunol*, 8:1807. 99

[417] Torre L. A., et al. (2015). Global cancer statistics, 2012. *CA Cancer J Clin*, 65(2):87–108. 100

[418] Kowalewski D. J. and Stevanović S. (2013). Biochemical large-scale identification of MHC class I ligands. *Methods Mol Biol*, 960:145–157. 102

[419] Barnstable C. J., et al. (1978). Production of monoclonal antibodies to group A erythrocytes, HLA and other human cell surface antigens-new tools for genetic analysis. *Cell*, 14(1):9–20. 103

[420] Faust G. G. and Hall I. M. (2014). SAMBLASTER: fast duplicate marking and structural variant read extraction. *Bioinformatics*, 30(17):2503–5. 103

[421] Cingolani P., et al. (2012). Using Drosophila melanogaster as a model for genotoxic chemical mutational studies with a new program, SnpSift. *Front Genet*, 3:35. 104

[422] Liu X., Wu C., Li C., and Boerwinkle E. (2016). dbNSFP v3.0: A One-Stop Database of Functional Predictions and Annotations for Human Nonsynonymous and Splice-Site SNVs. *Hum Mutat*, 37(3):235–41. 104

[423] Jiang H., Lei R., Ding S. W., and Zhu S. (2014). Skewer: a fast and accurate adapter trimmer for next-generation sequencing paired-end reads. *BMC Bioinformatics*, 15:182. 104

[424] Mose L. E., Wilkerson M. D., Hayes D. N., Perou C. M., and Parker J. S. (2014). ABRA: improved coding indel detection via assembly-based realignment. *Bioinformatics*, 30(19):2813–5. 104

[425] Chalmers Z. R., et al. (2017). Analysis of 100,000 human cancer genomes reveals the landscape of tumor mutational burden. *Genome Med*, 9(1):34. 104

[426] Cox J. and Mann M. (2008). MaxQuant enables high peptide identification rates, individualized p.p.b.-range mass accuracies and proteome-wide protein quantification. *Nat Biotechnol*, 26(12):1367–72. 105, 196

[427] Rost H. L., et al. (2016). OpenMS: a flexible open-source software platform for mass spectrometry data analysis. *Nat Methods*, 13(9):741–8. 105

[428] Eng J. K., Jahan T. A., and Hoopmann M. R. (2013). Comet: an open-source MS/MS sequence database search tool. *Proteomics*, 13(1):22–4. 105, 188, 189, 191, 192

[429] Kall L., Canterbury J. D., Weston J., Noble W. S., and MacCoss M. J. (2007). Semi-supervised learning for peptide identification from shotgun proteomics datasets. *Nat Methods*, 4(11):923–5. 105, 188, 189, 191, 192

[430] The M., MacCoss M. J., Noble W. S., and Kall L. (2016). Fast and accurate protein false discovery rates on large-scale proteomics data sets with Percolator 3.0. *J Am Soc Mass Spectrom*, 27(11):1719–1727. 105, 188, 189, 191, 192

[431] Almeida L. G., et al. (2009). CTdatabase: a knowledge-base of high-throughput and curated data on cancer-testis antigens. *Nucleic Acids Res*, 37(Database issue):D816–9. 107, 114, 178

[432] Python Software Foundation. Python language reference. `https://www.python.org`. 107

[433] Yarchoan M., Johnson, B. A. r., Lutz E. R., Laheru D. A., and Jaffee E. M. (2017). Targeting neoantigens to augment antitumour immunity. *Nat Rev Cancer*, 17(4):209–222. 116

[434] Fujimoto A., et al. (2012). Whole-genome sequencing of liver cancers identifies etiological influences on mutation patterns and recurrent mutations in chromatin regulators. *Nat Genet*, 44(7):760–4. 116, 117

[435] Martin S. D., et al. (2016). Low mutation burden in ovarian cancer may limit the utility of neoantigen-targeted vaccines. *PLoS One*, 11(5):e0155189. 116

[436] Gjerstorff M. F., Andersen M. H., and Ditzel H. J. (2015). Oncogenic cancer/testis antigens: prime candidates for immunotherapy. *Oncotarget*, 6(18):15772–87. 117

[437] Löffler M. W., et al. (2019). Multi-omics discovery of exome-derived neoantigens in hepatocellular carcinoma. *Genome Med*, 11:1–16. 117, 160

[438] Gfeller D. and Bassani-Sternberg M. (2018). Predicting antigen presentation-what could we learn from a million peptides? *Front Immunol*, 9:1716. 118

[439] Kahles A., et al. (2018). Comprehensive analysis of alternative splicing across tumors from 8,705 patients. *Cancer Cell*, 34(2):211–224 e6. 118

[440] Smart A. C., et al. (2018). Intron retention is a source of neoepitopes in cancer. *Nat Biotechnol*, 36(11):1056–1058. 118

[441] Hanada K., Yewdell J. W., and Yang J. C. (2004). Immune recognition of a human renal cancer antigen through post-translational protein splicing. *Nature*, 427(6971):252–6. 118

[442] Marijt K. A., et al. (2018). Identification of non-mutated neoantigens presented by tap-deficient tumors. *J Exp Med*, 215(9):2325–2337. 118

[443] Cobbold M., et al. (2013). Mhc class i-associated phosphopeptides are the targets of memory-like immunity in leukemia. *Sci Transl Med*, 5(203):203ra125. 118

[444] Menez-Jamet J., Gallou C., Rougeot A., and Kosmatopoulos K. (2016). Optimized tumor cryptic peptides: the basis for universal neo-antigen-like tumor vaccines. *Ann Transl Med*, 4(14):266. 118

[445] Marty R., Thompson W. K., Salem R. M., Zanetti M., and Carter H. (2018). Evolutionary pressure against MHC class II binding cancer mutations. *Cell*, 175(2):416–428 e13. 118

[446] Nielsen M., Lund O., Buus S., and Lundegaard C. (2010). MHC class II epitope predictive algorithms. *Immunology*, 130(3):319–28. 118

[447] Patro R., Duggal G., Love M. I., Irizarry R. A., and Kingsford C. (2017). Salmon provides fast and bias-aware quantification of transcript expression. *Nat Methods*, 14(4):417–419. 118, 119

[448] Ahmed N., et al. (2019). GASAL2: a GPU accelerated sequence alignment library for high-throughput NGS data. *BMC Bioinformatics*, 20(1):520. 119

[449] Ewels P. A., et al. (2020). The nf-core framework for community-curated bioinformatics pipelines. *Nat Biotechnol*, 38(3):276–278. 119, 123

[450] McGranahan N., et al. (2016). Clonal neoantigens elicit T cell immunoreactivity and sensitivity to immune checkpoint blockade. *Science*, 351(6280):1463–9. 121

[451] Anagnostou V., et al. (2017). Evolution of neoantigen landscape during immune checkpoint blockade in non-small cell lung cancer. *Cancer Discov*, 7(3):264–276.

[452] Bentzen A. K., et al. (2016). Large-scale detection of antigen-specific T cells using peptide-MHC-I multimers labeled with DNA barcodes. *Nat Biotechnol*, 34(10):1037–1045.

[453] Senbabaoglu Y., et al. (2016). Tumor immune microenvironment characterization in clear cell renal cell carcinoma identifies prognostic and immunotherapeutically relevant messenger RNA signatures. *Genome Biol*, 17(1):231.

[454] Hartmaier R. J., et al. (2017). Genomic analysis of 63,220 tumors reveals insights into tumor uniqueness and targeted cancer immunotherapy strategies. *Genome Med*, 9(1):16.

[455] McGranahan N., et al. (2017). Allele-specific HLA loss and immune escape in lung cancer evolution. *Cell*, 171(6):1259–1271 e11.

[456] Marty R., et al. (2017). MHC-I genotype restricts the oncogenic mutational landscape. *Cell*, 171(6):1272–1283 e15. 121

[457] Gubin M. M. and Schreiber R. D. (2015). The odds of immunotherapy success. *Science*, 350(6257):158–9. 123

[458] Global Alliance for Genomics and Health (2016). A federated ecosystem for sharing genomic, clinical data. *Science*, 352(6291):1278–80. 124

[459] Vågene A. J., et al. (2018). Salmonella enterica genomes from victims of a major sixteenth-century epidemic in Mexico. *Nat Ecol Evol*, 2(3):520–528. 179

[460] Weisser H. and Choudhary J. S. (2017). Targeted feature detection for data-dependent shotgun proteomics. *J Proteome Res*, 16(8):2964–2974. 188, 189, 191, 192

Appendix A: Abbreviations

AA	*Amino acids*
ABC	*ATP-binding cassette*
ALL	*Acute lymphoblastic leukaemia*
ANN	*Artificial neural network*
APC	*Antigen-presenting cell*
auROC	*area under the Receiver Operating Characteristic*
bp	*base pair*
BCR	*B-cell receptor*
BWT	*Burrows-Wheeler transform*
CDS	*Coding DNA sequence*
CNV	*Copy number variation*
CRT	*Cycle reversible termination*
CTA	*Cancer/testis antigen*
CTD	*Common tool descriptor*
CTL	*Cytotoxic lymphocyte*
DCI	*Distributed Computing Infrastructure*
dNTP	*Deoxyribonucleotide triphosphate*
emPCR	*emulsion PCR*
ER	*Endoplasmatic reticulum*
ETL	*Extract, transform, load*
EV	*Epitope-based vaccine*
FDA	*Food and Drug Administration*

FDR	*False discovery rate*
FPKM	*Fragments per kilobase per million mapped reads*
GRC	*Genome Reference Consortium*
GUI	*Graphical user interface*
GTF	*Gene transfer format*
HBV	*Hepatitis B virus*
HCC	*Hepatocellular carcinoma*
HLA	*Human leukocyte antigen*
HPC	*High-performance computing*
HPV	*Human Papillomavirus*
ICGC	*International cancer genome consortium*
IEDB	*Immune Epitope Database*
Ig	*Immunoglobulin*
ILP	*Integer linear program*
InDel	*Insertion/deletion*
LDAP	*Lightweight directory access Protocol*
LFQ	*Label-free quantification*
LIMS	*Laboratory information management system*
mAb	*Monoclonal antibody*
MAF	*Minor allele frequency*
MS	*Mass spectrometry*
MHC	*Major histocompatibility complex*
NCI	*National Cancer Institute*
NGS	*Next-generation sequencing*
openBIS	*Open biological information system*
PBMC	*Peripheral blood mononuclear cell*
PCM	*Proteasomal cleavage matrix*
PSSM	*Position-specific scoring matrix*

QC	*Quality control*
RPKM	*Reads per kilobase per million mapped reads*
RNA-Seq	*RNA sequencing*
SAM	*Sequence Alignment Map*
SBL	*Sequencing by ligation*
SBS	*Sequencing by synthesis*
SNV	*Single nucleotide variant*
SSO	*Single sign-on*
TAA	*Tumor-associated antigen*
TCR	*T-cell receptor*
TMB	*Tumor mutational burden*
TPM	*Transcript per million*
TSA	*Tumor-specific antigen*
TSP	*Travelling Salesman Problem*
UCSC	*University of California, Santa Cruz*
VCF	*Variant Call Format*
VM	*Virtual machine*
WES	*Whole-exome sequencing*
WGS	*Whole-genome sequencing*
WTS	*Whole-transcriptome sequencing*
WS-PGRADE	*Web Services Parallel Grid Runtime and Developer Environment*

Appendix B: Contributions

All ideas, approaches and results presented in this work were developed and discussed with my supervisor Prof. Dr. Oliver Kohlbacher (OK). The following contributions were made by co-workers and myself (CM) to the projects presented.

Chapter 3: HLA Genotyping from Next-Generation Sequencing Data

OK, AS, BS, and CM designed the project. AS, BS and CM designed and implemented the HLA typing pipeline. AS, BS, and CM designed, performed, and evaluated the experiments. CM, MF, AS, BS and MS prepared the data. CM performed the β parameter evaluation. AS, BS, CM and OK wrote the manuscript[301]. All authors read and approved the manuscript. OK designed the study.

Chapter 4: T-Cell Immunogenicity: Modeling Immunological Tolerance

OK and CM designed the project. CM, BS designed the models. CM implemented the method. BS, CM performed the cleavage predictions. CM, BS prepared the data. CM performed and evaluated the experiments. CM contributed code to ImmunoNodes for the *distance-to-self* calculation.

Chapter 5: iVacPortal – A Web-based Portal for Personalized Vaccine Design

OK, SN, DW, AF and CM designed the project. CM, AF, and DW implemented qPortal. ACP, CM, and AF contributed to the design of the user interface. CM and DW implemented qNavigator. DW and CM implemented the workflowAPI and qFlow. AF implemented qWizard. EK, CM, AF implemented the data model. CM and AF implemented the ETL scripts. CM, MC, SC, MS, MW, and LB implemented pipelines (detailed information about authors of workflows are given in Appendix F). CM and DW implemented the gUSE workflows. BS formulated the ILP for the neoepitope selection. BS and CM implemented the software package. CM and JS implemented the Interactive Vaccine Designer. CM, AF, and SN wrote the qPortal manuscript[379]. All authors contributed with comments and suggestions to the manuscript. All authors read and approved the final version of the manuscript.

Chapter 6: Assessment of Personalized Vaccine Options through iVacPortal

ML, OR, BM, LBu, OK, CM, AR, SS, HEPAVAC and HR made the study/project concept and design. CM implemented the pipeline for eptiope prediction and annotation (EPAA) and performed the analysis. CM implemented the HLA typing workflow and others (see Appendix F for details). LB and CM implemented the HLA ligandome identification workflow. ML, CM, LB, LF, MW, CS, FJH, RZ, LM, DK, AR, HS, MS, JM, SC, SN, IK, KT, SNa, SB, FF, AV, BM, SH, and HR conducted analysis and/or interpretation of data. LB, LF, CM performed the HLA ligandome analysis. ML, LF, NT, FH, RZ, LM, DK, HS, HB, and AV acquired the data. CM and LB prepared the data. CM, ML, and AR performed the data and project management. CM, LB, LF performed the downstream analysis. ML, CM, LB, LF, and SC drafted/wrote the manuscript for the project on HCC[437]. MW, CS, NT, FH, RZ, LM, DJK, HS, MS, JM, OR, SC, SN, IK, KT, SNa, SB, HB, FF, AV, BM, SPH, LBu, OK, SS, AK, and HR revised the manuscript. All authors read and approved the final version of the manuscript.

SB: Stefan Beckert, LB: Leon Bichmann, LBu: Luigi Buonaguro, HB: Hans Bösmüller, MC: Marius Cosmin Codrea, SC: Stefan Czemmel, MF: Magdalena Feldhahn, FF: Falko Fend, LF: Lena Katharina Freudenmann, AF: Andreas Friedrich, SH: Sebastian P. Haen, FH: Franz J. Hilke, EK: Erhan Kenar, OK: Oliver Kohlbacher, DK: Daniel J. Kowalewski, ML: Markus W. Löffler, BM: Boris Maček, JM: Jakob Matthes, **CM: Christopher Mohr**, LM: Lena Mühlenbruch, SNa: Silvio Nadalin, SN: Sven Nahnsen, ACP: Aydin Can Polatkan, AR: Armin Rabsteyn, HR: Hans-Georg Rammensee, OR: Olaf Riess, BS: Benjamin Schubert, HS: Heiko Schuster, CS: Christopher M. Schröder, JS: Julian Späth, SS: Stefan Stevanović, MS: Marc Sturm, AS: András Szolek, NT: Nico Trautwein, AV: Ana Velic, MW: Mathias Walzer, DW: David Wojnar, RZ: Raphael S. Zinser

Appendix C: Publications

2019

Schneider, L., Kehl, T., Thedinga, K., Grammes, N.L., Backes, C., **Mohr, C.**, Schubert, B., Lenhof, K., Gerstner, N., Hartkopf, A. D., Wallwiener, M., Kohlbacher, O., Keller, A., Meese, E., Graf, N., and Lenhof, H.-P.
ClinOmicsTrail^{bc}: a visual analytics tool for breast cancer treatment stratification
Bioinformatics **35** (24), 5171-5181

Löffler, M. W.*, **Mohr, C.***, Bichmann, L., Freudenmann, L. K., Walzer, M., Schroeder, C. M., Trautwein, N., Hilke, F. J., Zinser, R., Mühlenbruch, L., Kowalewski, D. J., Schuster, H., Sturm, M., Matthes, J., Riess, O., Czemmel, S., Nahnsen, S., Königsrainer, I., Thiel, K., Nadalin, S., Beckert, S., Bösmüller, H., Fend, F., Velic, A., Maček, B., Haen, S. P., Buonaguro, L., Kohlbacher, O., Stevanović, S., Königsrainer, A., HEPAVAC Consortium, and Rammensee, H.-G.
Multi-omics discovery of exome-derived neoantigens in hepatocellular carcinoma
Genome Medicine **11** (1), 28
* Joint first authors

Bichmann, L., Nelde ,A., Ghosh, M., Heumos, L., **Mohr, C.**, Peltzer, A., Kuchenbecker, L., Sachsenberg, T., Walz, J. S., Stevanović, S., Rammensee, H.-G., and Kohlbacher, O.
MHCquant: Automated and reproducible data analysis for immunopeptidomics
Journal of Proteome Research **18** (11), 3876-3884

Blaeschke, F., Paul, M. C., Schuhmann, M. U., Rabsteyn, A., Schroeder, C., Casadei, N., Matthes, J., **Mohr, C.**, Lotfi, R., Wagner, B., Kaeuferle, T., Feucht, J., Willier, S., Handgretinger, R., Stevanović, S., Lang, P., and Feuchtinger, T.

Low mutational load in pediatric medulloblastoma still translates into neoantigens as targets for specific T-cell immunotherapy
Cytotherapy **21** (9), 973-986

Rabsteyn, A., Kyzirakos, C., Schroeder, C., Sturm, M., **Mohr, C.**, Matthes, J., Feldhahn, M., Casadei, N., Ebinger, M., Stevanović, S., Bauer, P., Kohlbacher, O., Gouttefangeas, C., Schaefer, J., Rammensee, H.-G. , Handgretinger, R., and Lang, P.
Abstract B124: Personalized peptide vaccination based on patient-individual tumor-specific variants induces T-cell responses in pediatric patients
Cancer Immunology Research **7** (2 Supplement), B124-B124

2018

Haen, S., Löffler, M. W., Kohlbacher, O., Nahnsen, S., **Mohr, C.**, Stieglbaueren, M., Hrstic, P., Buonaguro, L., Martus, P., Häntschel, M., Gouttefangeas, C., Beckert, S., Königsrainer, A., Stevanović, S., Kanz, L., and Rammensee, H.-G.
Abstract CT057: Phase I trial to evaluate the feasibility and safety of an individualized peptide vaccine of unmodified cancer antigens: PepIVAC-01
Cancer Research **78** (13 Supplement), CT057-CT057

Mohr, C.*, Friedrich, A.*, Wojnar, D., Kenar, E., Polatkan, A. C., Codrea, M. C., Czemmel, S., Kohlbacher, O., and Nahnsen, S.
qPortal: A platform for data-driven biomedical research
PLoS ONE **13**(1), e0191603
* Joint first authors

2017

Schubert, B., de la Garza, L., **Mohr, C.**, Walzer, M., and Kohlbacher, O.
ImmunoNodes – graphical development of complex immunoinformatics workflows
BMC Bioinformatics, **18**(1), 242

Armeanu-Ebinger, S., Hadaschik, D., Kyzirakos, C., **Mohr, C.**, Battke, F., Kohlbacher, O., Nahnsen, S., and Biskup, S.
Number of predicted tumour-neoantigens as biomarker for cancer immunotherapies
Annals of Oncology, **28**(Supplement 7), 12-12

2016

Kyzirakos, C., **Mohr, C.**, Armeanu-Ebinger, S., Feldhahn, M., Hadaschik, D., Walzer, M., Döcker, D., Menzel, M., Nahnsen, S., Kohlbacher, O. and Biskup, S.
Optimized neoantigen selection based on tumor exome data
Annals of Oncology, **27**(Supplement 6), 1097P

Löffler, M.W., Chandran, P.A., Laske, K., Schroeder, C., Bonzheim, I., Walzer, M., Hilke, F.J., Trautwein, N., Kowalewski, D.J., Schuster, H., Günder, M., Cacamo Yañez, V. A., **Mohr, C.**, Sturm, M., Nguyen, H.P., Riess, O., Bauer, P., Nahnsen, S., Nadalin, S., Zieker, D., Glatzle, J., Thiel, K., Schneiderhan-Marra, N., Clasen, S., Bösmüller, H., Fend, F., Kohlbacher, O., Gouttefangeas, C., Stevanović, S., Königsrainer, A., and Rammensee, H.-G.
Personalized peptide vaccine-induced immune response associated with long-term survival of a metastatic cholangiocarcinoma patient
Journal of Hepatology, **65**(4), 849-855

Schubert, B., Walzer, M., Brachvogel, H.P., Szolek, A., **Mohr, C.** and Kohlbacher, O.
FRED 2: an immunoinformatics framework for Python. Bioinformatics
Bioinformatics, **32**(13), 2044-2046

Lang, P., Rabsteyn, A., Kyzirakos, C., Schroeder, C., Sturm, M., **Mohr, C.**, Walzer, M., Pflueckhahn, U., Walter, M., Feldhahn, M., Laske, K., Bonin, M., Ebinger, M., Stevanović, S., Bauer, P., Kohlbacher, O., Gouttefangeas, C., Handgretinger, R., and Rammensee, H.-G.
Personalized peptide-vaccination for pediatric acute lymphoblastic leukemia patients based on patient-individual tumor-specific variants (iVacALL)
Oncology Research and Treatment, **39**, 125-125

Rabsteyn, A., Kyzirakos, C., Schröder, C., Sturm, M., **Mohr, C.**, Walzer, M., Pflückhahn, U., Walter, M., Feldhahn, M., Laske, K., Bonin, M., Ebinger, M., Stevanović, S., Bauer, P., Kohlbacher, O., Gouttefangeas, C., Rammensee, H.G., Handgretinger, R., and Lang, P.
Abstract A113: iVacALL: A personalized peptide-vaccination design platform for pediatric acute lymphoblastic leukemia patients based on patient-individual tumor-specific variants
Cancer Immunology Research, **4**(1 Supplement), A113

2014

Szolek, A.*, Schubert, B.*, **Mohr, C.***, Sturm, M., Feldhahn, M., and Kohlbacher, O.
OptiType: precision HLA typing from next-generation sequencing data
Bioinformatics, **30**(23), 3310-3316
* Joint first authors

Olabarriaga, S.D., Benabdelkader, A., Caan, M.W., Jaghoori, M.M., Krüger, J., de la Garza, L., **Mohr, C.**, Schubert, B., Danezi, A. and Kiss, T.
WS-PGRADE/gUSE-based science gateways in teaching
Science Gateways for Distributed Computing Infrastructures, Springer, Cham., 223-234

2013

Kyzirakos, C., Pflückhahn, U., Sturm, M., Schroeder, C., Bauer, P., Walter, M., Feldhahn, M., Walzer, M., **Mohr, C.**, Szolek, A. Bonin, M., Kohlbacher O., Ebinger, M., Handgretinger R., Rammensee, H.G., and Lang, P.
iVacALL: utilizing next-generation sequencing for the establishment of an individual peptide vaccination approach for paediatric acute lymphoblastic leukaemia
Bone Marrow Transplantation, **48**, S401

Appendix D: Supporting Figures

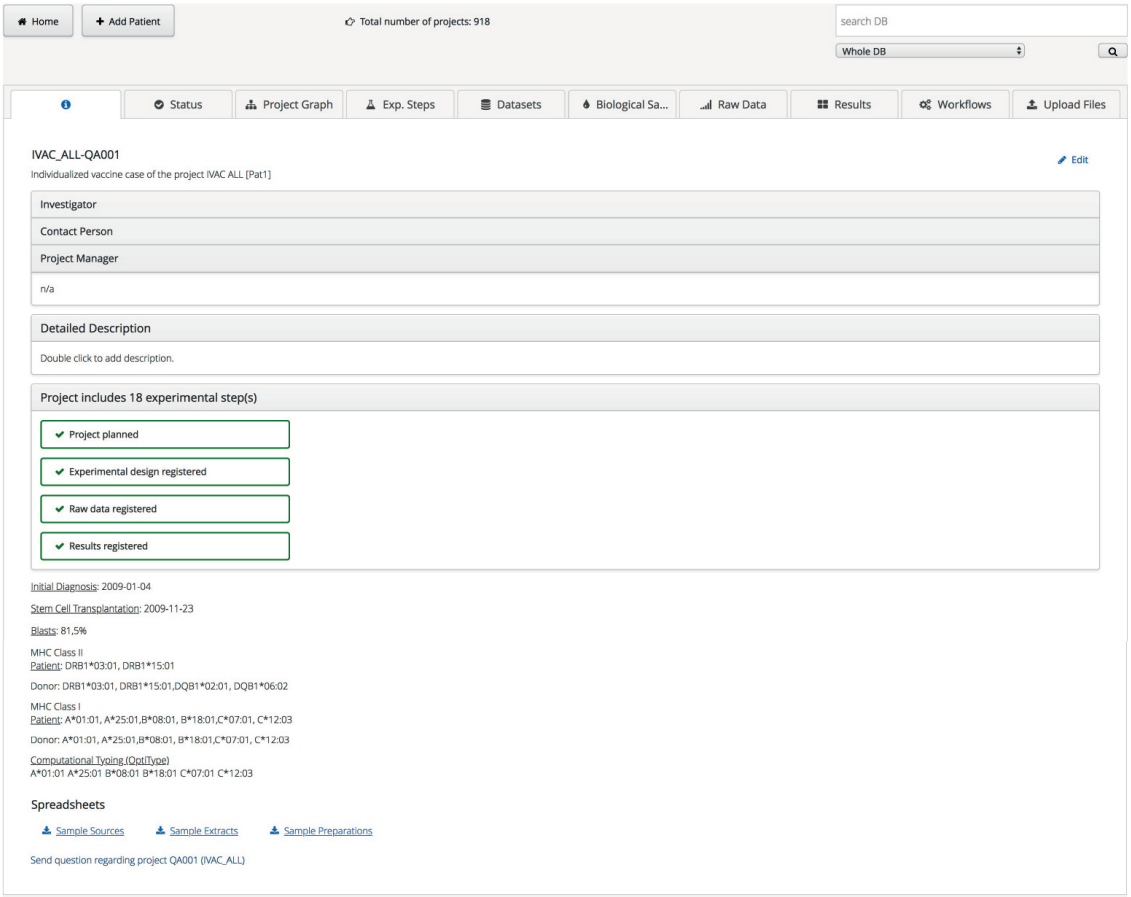

Figure D.1: Project view for projects in iVacPortal. The view additionally contains information about the HLA genotype of a patient and further metadata annotations such as the initial diagnosis date.

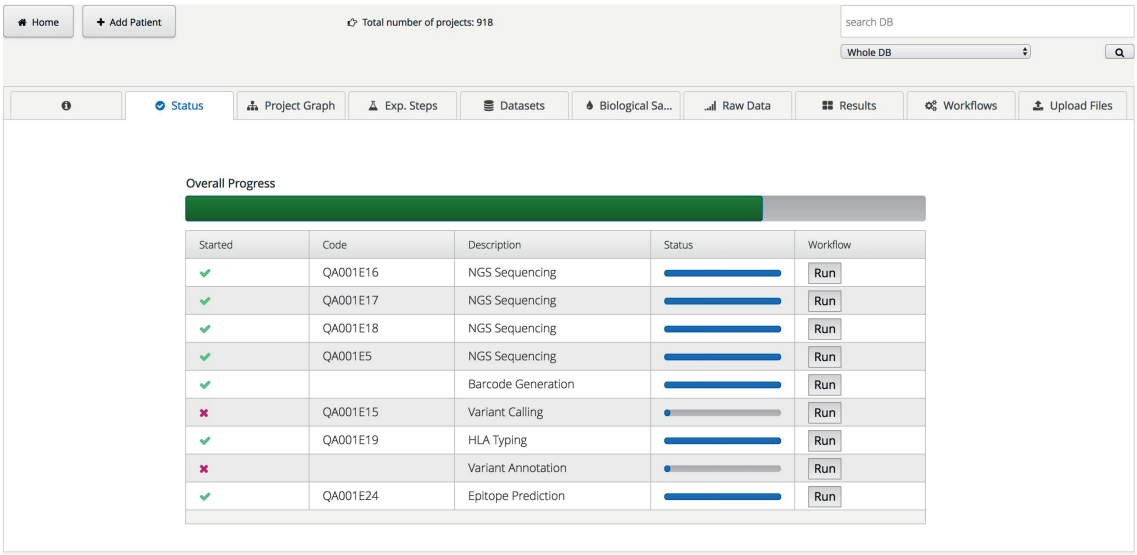

Figure D.2: Status view for projects in iVacPortal. For personalized vaccine projects, an additional status component summarizes the overall state of the project and its distinct steps.

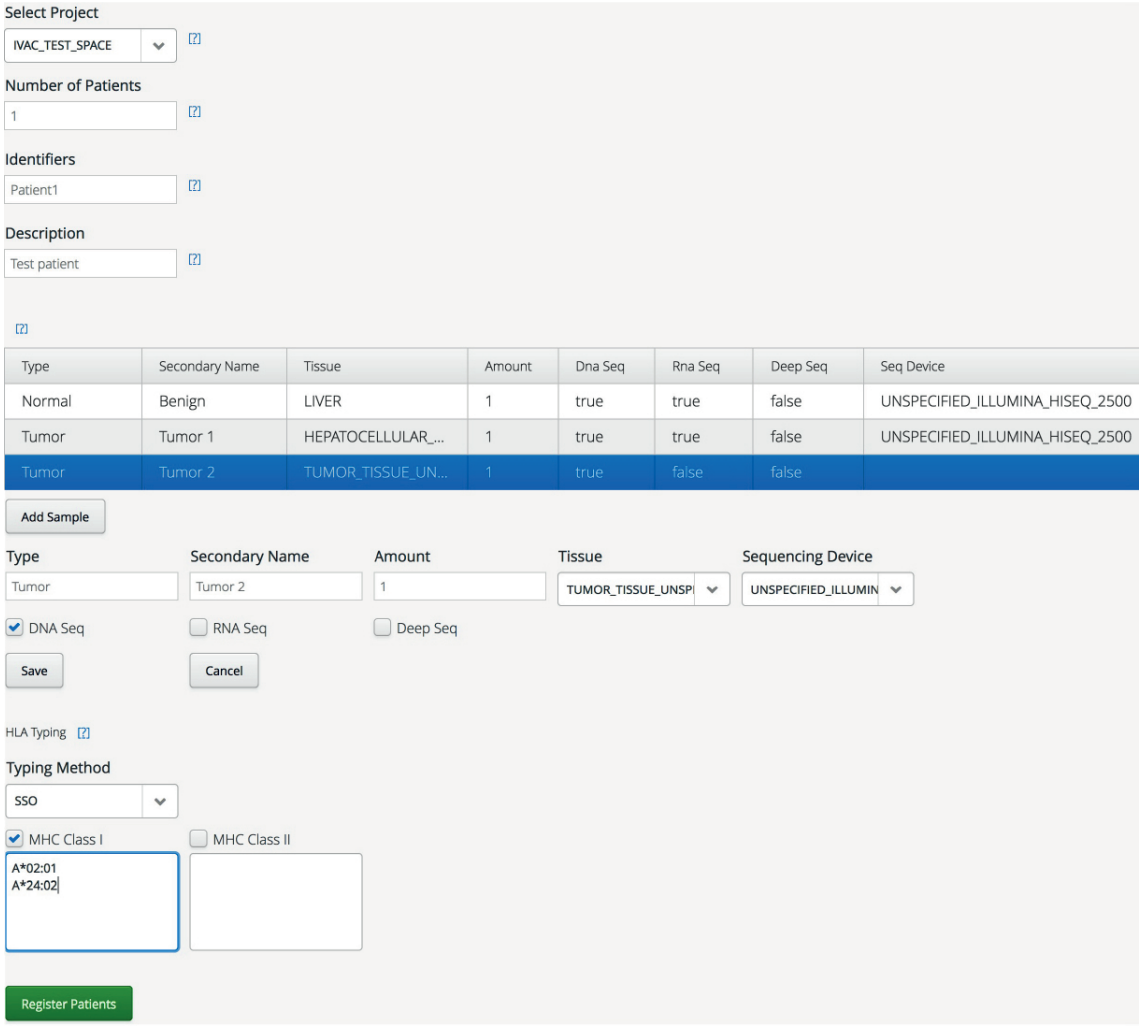

Figure D.3: iVacPortal view for the registration of new patients. The view is accessible from qNavigator and can be used to add one or multiple new patients (projects). Users may add several samples per patient and specify details such as the type of tissue, the tissue origin, the sequencing device which will be used, and the sequencing technique performed. If available, users may add HLA typing information as well. Patients will be registered upon submission through implemented services.

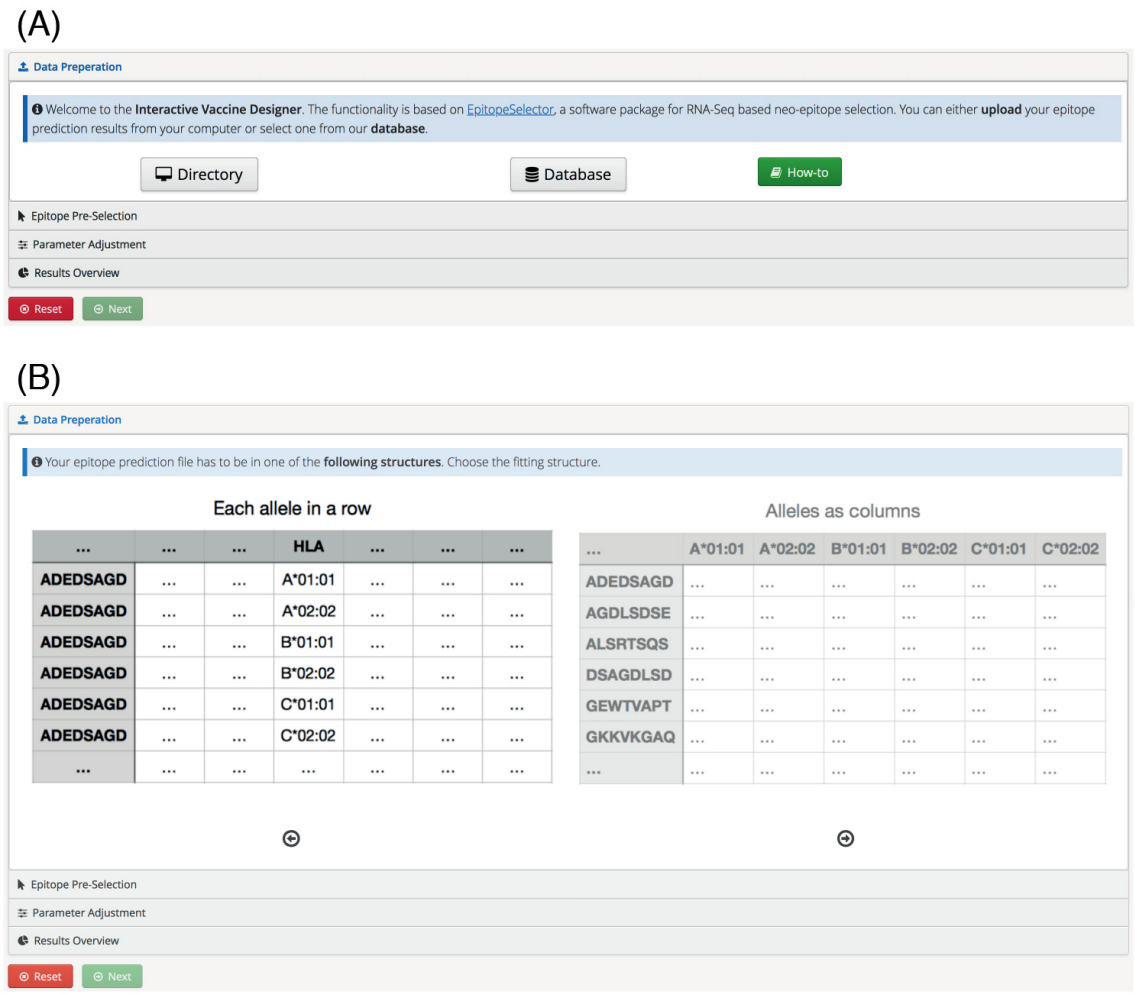

Figure D.4: Data preparation in the Interactive Vaccine Designer. (A) Users may choose between file upload and the selection of files stored in the openBIS datastore server. The downloadable how-to describes all the steps in detail. (B) Users have the option to select between two file structures for the provided epitope predictions.

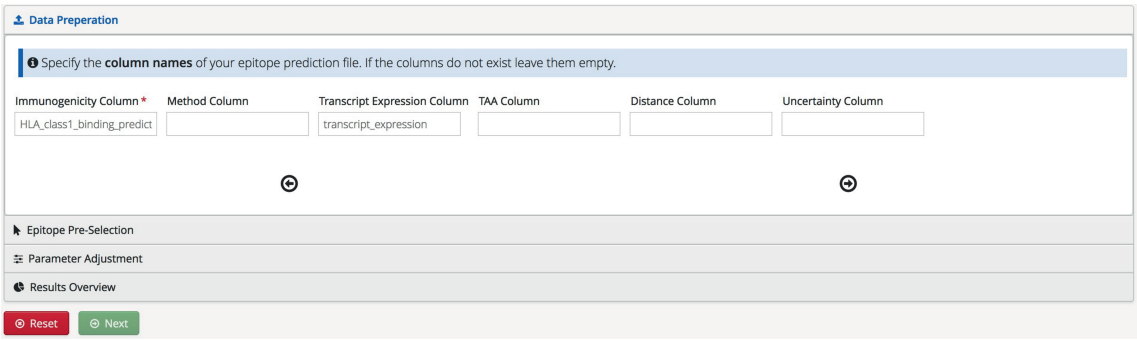

Figure D.5: Data preparation in the Interactive Vaccine Designer. Users may specify the column names as given in the provided epitope prediction results.

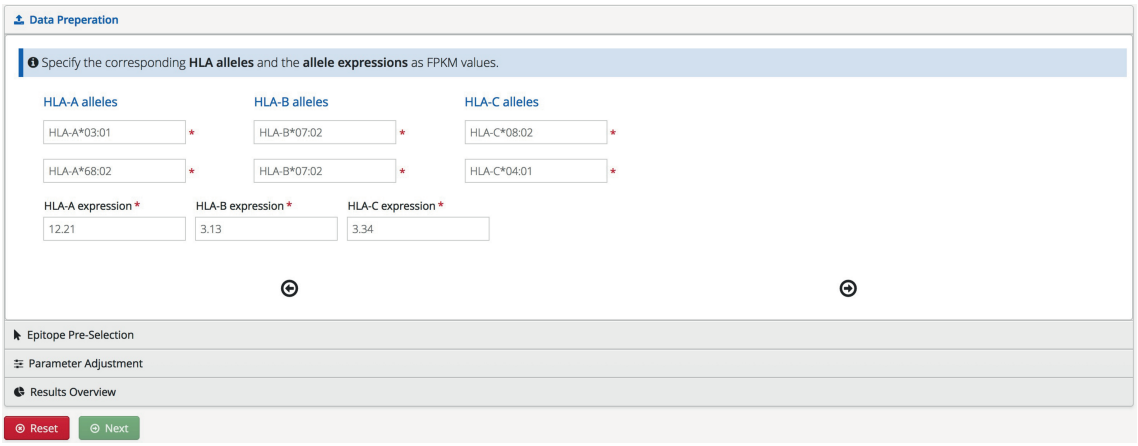

Figure D.6: Data preparation of HLA alleles in the Interactive Vaccine Designer. If users choose to specify HLA alleles manually, the alleles and the corresponding HLA loci expression values (as FPKM) have to be specified.

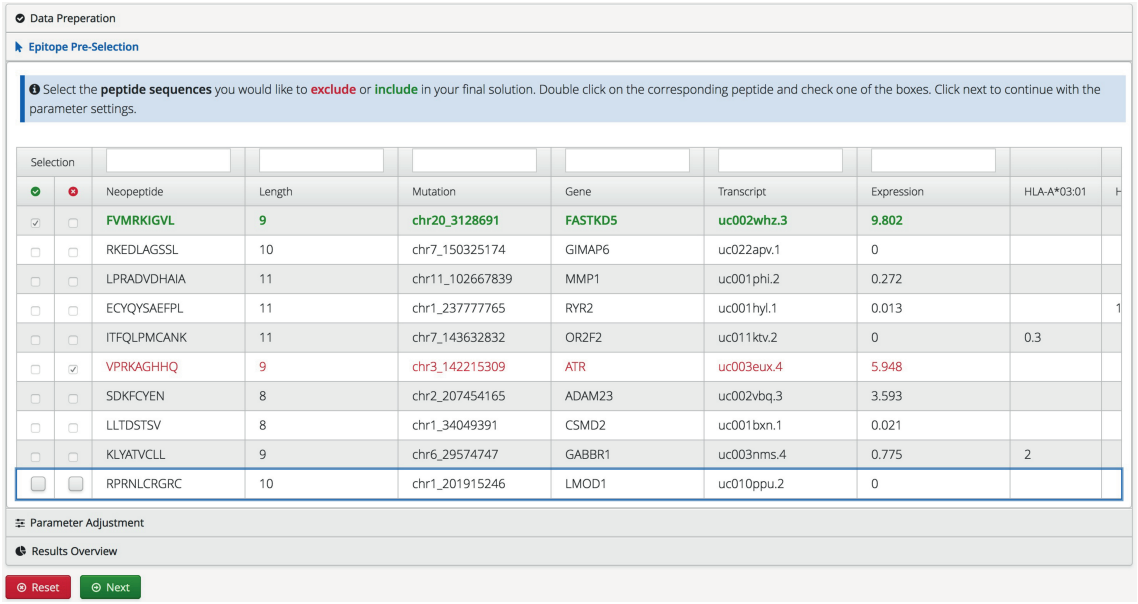

Figure D.7: Peptide preselection in the Interactive Vaccine Designer. Users may explicitly include (green) or exclude (red) specific peptide sequences for the final solution.

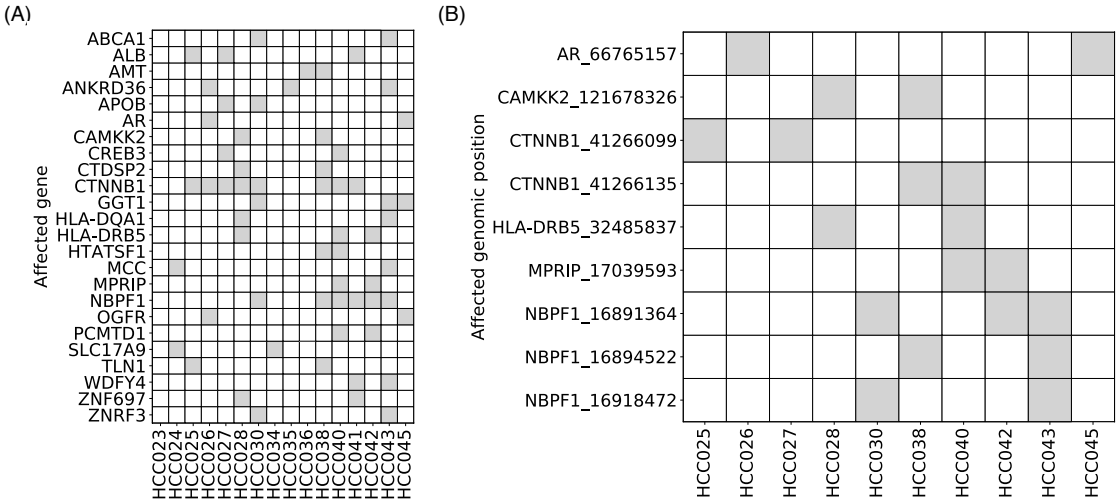

Figure D.8: Shared Vars^{exp} across the investigated HCC cohort. Expressed mutations were analyzed with respect to presence (depicted by a grey square) in multiple patients. (A) 24 genes carried a Var^{exp} in $\geq$ two patients. (B) Nine Vars^{exp} were shared by $\geq$ two patients. The mutation in gene *NBPF1* on position 16891364 was found to be shared by three patients. The genomic positions are zero-based.

Appendix E: Supporting Tables

Table E.1: Sample identifiers and run accession identifiers used for the evaluation of OptiType.

CRC		1000 Genomes Project (exome)							
Sample ID	Run ID	Sample ID	Run ID	Sample ID	Run ID	Sample ID	Run ID	Sample ID	Run ID
17	SRR396926	NA06985	SRR709972	NA18537	ERR032033	NA18971	SRR078842	NA19207	SRR081256
	SRR396998	NA06994	SRR070528	NA18537	ERR032034	NA18972	SRR077490	NA19209	SRR077489
	SRR397070	NA06994	SRR070819	NA18542	ERR031855	NA18972	SRR081255	NA19209	SRR077859
	SRR397142	NA07000	SRR766039	NA18545	ERR031856	NA18973	SRR077861	NA19210	SRR078845
		NA07048	SRR099452	NA18547	ERR031957	NA18973	SRR078846	NA19210	SRR081222
20	SRR396928	NA07056	SRR764718	NA18550	ERR031958	NA18974	SRR077456	NA19222	SRR748214
	SRR397000	NA07357	SRR764689	NA18552	ERR031959	NA18974	SRR081248	NA19223	SRR071186
	SRR397072	NA07357	SRR764690	NA18555	ERR031857	NA18975	SRR078849	NA19223	SRR071193
	SRR397144	NA10847	SRR070531	NA18558	ERR031960	NA18975	SRR081225	NA19238	SRR071173
		NA10847	SRR070823	NA18561	ERR031858	NA18976	SRR077451	NA19238	SRR071195
42	SRR396942	NA10851	SRR766044	NA18562	ERR031859	NA18976	SRR077757	NA19238	SRR792121
	SRR397014	NA11829	SRR710128	NA18563	ERR031860	NA18978	SRR716650	NA19238	SRR792165
	SRR397086	NA11830	SRR766026	NA18564	ERR031861	NA18980	SRR716652	NA19239	SRR792097
	SRR397158	NA11831	SRR709975	NA18566	ERR031862	NA18980	SRR716653	NA19239	SRR792159
		NA11832	SRR766003	NA18570	ERR031863	NA18981	SRR077477	NA19240	SRR792091
49	SRR396946	NA11840	SRR070532	NA18571	ERR031868	NA18981	SRR077751	NA19240	SRR792767
	SRR397018	NA11840	SRR070809	NA18572	ERR031869	NA18987	SRR077491	NA20313	SRR359098
	SRR397090	NA11881	SRR766021	NA18573	ERR031870	NA18987	SRR077853	NA20313R	SRR359108
	SRR397162	NA11992	SRR701474	NA18576	ERR031871	NA18990	SRR077454	HG01756	SRR359102
		NA11994	SRR701475	NA18577	ERR032035	NA18990	SRR077486	HG01757	SRR359103
53	SRR396949	NA11995	SRR766010	NA18577	ERR032036	NA18991	SRR077450	HG01872	SRR359298
	SRR397021	NA12003	SRR766061	NA18579	ERR032037	NA18991	SRR077855	HG01873	SRR359295
	SRR397093	NA12004	SRR766059	NA18579	ERR032038	NA18992	SRR716428	HG01886	SRR360655
	SRR397165	NA12005	SRR718067	NA18582	ERR031961	NA18994	SRR716431	HG01953	SRR360288
		NA12006	SRR716422	NA18592	ERR031962	NA18995	SRR764775	HG01968	SRR360391
65	SRR396959	NA12043	SRR716423	NA18593	ERR034531	NA18997	SRR702078	HG02014	SRR360148
	SRR397031	NA12043	SRR716424	NA18603	ERR031872	NA18998	SRR766013	HG02057	SRR359301
	SRR397103	NA12044	SRR766060	NA18605	ERR031873	NA18999	SRR112297		
	SRR397175	NA12144	SRR766058	NA18608	ERR031874	NA19000	SRR099528		
		NA12154	SRR702067	NA18609	ERR031875	NA19003	SRR099532		

66	SRR397206	NA12155	SRR702068	NA18611	ERR031876	NA19005	SRR715906
	SRR397266	NA12156	SRR764691	NA18612	ERR034593	NA19007	SRR099549
	SRR397326	NA12234	SRR716435	NA18620	ERR031877	NA19012	SRR112294
	SRR397386	NA12249	SRR070525	NA18621	ERR034595	NA19092	SRR100012
		NA12249	SRR070798	NA18622	ERR032027	NA19093	SRR100033
70	SRR397210	NA12716	SRR081269	NA18622	ERR032028	NA19098	SRR077453
	SRR397270	NA12716	SRR081274	NA18623	ERR032008	NA19098	SRR077460
	SRR397330	NA12717	SRR071172	NA18624	ERR031928	NA19099	SRR748771
	SRR397390	NA12717	SRR071177	NA18632	ERR031929	NA19099	SRR748772
		NA12750	SRR077449	NA18633	ERR031878	NA19102	SRR100034
75	SRR397214	NA12750	SRR081238	NA18635	ERR031879	NA19116	SRR100021
	SRR397274	NA12750	SRR794547	NA18636	ERR031930	NA19119	SRR077471
	SRR397334	NA12750	SRR794550	NA18637	ERR031931	NA19119	SRR081271
	SRR397394	NA12751	SRR071136	NA18853	SRR100011	NA19129	ERR034558
		NA12751	SRR071139	NA18856	SRR098533	NA19130	SRR107026
81	SRR397217	NA12760	SRR081223	NA18858	ERR034553	NA19131	SRR070494
	SRR397277	NA12760	SRR081251	NA18861	ERR034554	NA19131	SRR070783
	SRR397337	NA12761	SRR077753	NA18870	SRR100031	NA19137	SRR081226
	SRR397397	NA12761	SRR081267	NA18871	SRR100029	NA19137	SRR081237
		NA12762	SRR718076	NA18912	SRR111960	NA19137	SRR792542
83	SRR397218	NA12763	SRR077752	NA18940	ERR034596	NA19137	SRR792560
	SRR397278	NA12763	SRR081230	NA18942	ERR034597	NA19138	SRR070472
	SRR397338	NA12812	SRR715913	NA18943	ERR034598	NA19138	SRR070776
	SRR397398	NA12813	SRR718077	NA18944	ERR034599	NA19141	SRR077433
		NA12813	SRR718078	NA18945	ERR034600	NA19141	SRR077464
88	SRR397222	NA12814	SRR715914	NA18947	ERR034601	NA19143	SRR077445
	SRR397282	NA12815	SRR716646	NA18948	ERR034602	NA19143	SRR081272
	SRR397342	NA12872	SRR716647	NA18949	ERR034603	NA19144	SRR077392
	SRR397402	NA12873	SRR702070	NA18951	ERR034604	NA19144	SRR077468
		NA12874	SRR764692	NA18952	ERR034605	NA19152	SRR071135
90	SRR397224	NA12878	SRR098401	NA18953	SRR099546	NA19152	SRR071167
	SRR397284	NA12891	SRR098359	NA18956	SRR766028	NA19153	SRR070660
	SRR397344	NA12892	ERR034529	NA18959	SRR099545	NA19153	SRR070846
	SRR397404	NA18501	SRR100022	NA18960	SRR099533	NA19159	SRR070478
		NA18502	SRR764722	NA18961	SRR099544	NA19159	SRR070786
95	SRR397229	NA18502	SRR764723	NA18964	SRR099539	NA19160	SRR077482
	SRR397289	NA18504	SRR100028	NA18965	SRR764771	NA19160	SRR081250
	SRR397349	NA18505	SRR716648	NA18965	SRR764772	NA19171	SRR077492
	SRR397409	NA18505	SRR716649	NA18966	SRR071175	NA19171	SRR077493
		NA18507	SRR764745	NA18966	SRR071180	NA19172	SRR111962
97	SRR397231	NA18507	SRR764746	NA18967	SRR071192	NA19200	SRR077432
	SRR397291	NA18508	SRR716637	NA18967	SRR071196	NA19200	SRR078847
	SRR397351	NA18508	SRR716638	NA18968	SRR077480	NA19201	SRR077439
	SRR397411	NA18516	SRR100026	NA18968	SRR081231	NA19201	SRR077462

		NA18517	ERR034551	NA18969	SRR081266	NA19204	SRR077857
99	SRR397233	NA18522	SRR107025	NA18969	SRR081273	NA19204	SRR081263
	SRR397293	NA18523	ERR034552	NA18970	SRR071116	NA19206	SRR070491
	SRR397353	NA18526	ERR031854	NA18970	SRR071127	NA19206	SRR070781
	SRR397413	NA18532	ERR031956	NA18971	SRR077447	NA19207	SRR081254

Low coverage HapMap WGS		**CEU**					
Sample ID	Run ID	Sample ID	Run ID	Sample ID	Run ID	Sample ID	Run ID
NA06985	SRR400039	NA06985	ERR009159	NA12003	ERR009121	NA12812	ERR009104
NA11832	SRR385763	NA06994	ERR009168	NA12004	ERR009139	NA12813	ERR009114
NA12005	SRR385767	NA07000	ERR009154	NA12005	ERR009155	NA12814	ERR009134
NA12044	SRR393991	NA07051	ERR009147	NA12006	ERR009123	NA12815	ERR009151
NA12760	SRR385773	NA07346	ERR009133	NA12043	ERR009163	NA12872	ERR009099
NA18912	SRR350153	NA07347	ERR009146	NA12044	ERR009157	NA12873	ERR009111
NA18960	SRR442016	NA07357	ERR009167	NA12045	ERR009113	NA12874	ERR009145
NA18968	SRR359062	NA10847	ERR009097	NA12144	ERR009117	NA12891	ERR009105
NA18971	SRR359095	NA10851	ERR009124	NA12154	ERR009129		
NA18974	SRR360136	NA11829	ERR009122	NA12155	ERR009115		
NA18975	SRR359070	NA11830	ERR009140	NA12156	ERR009136		
NA18976	SRR359110	NA11831	ERR009096	NA12234	ERR009144		
NA18981	SRR359083	NA11832	ERR009109	NA12249	ERR009107		
NA18991	ERR052929	NA11840	ERR009142	NA12716	ERR009118		
NA19092	SRR189830	NA11881	ERR009135	NA12717	ERR009164		
NA19119	SRR359106	NA11918	ERR009166	NA12750	ERR009137		
NA19131	SRR359096	NA11920	ERR009149	NA12751	ERR009132		
NA19152	SRR359097	NA11992	ERR009119	NA12760	ERR009130		
NA19171	SRR359061	NA11993	ERR009103	NA12761	ERR009106		
NA19204	SRR359064	NA11994	ERR009141	NA12762	ERR009156		
NA12006	SRR385760	NA11995	ERR009108	NA12763	ERR009152		

Table E.2: 1000 Genomes Project samples that have been used for the qPortal case study. The experimental variables ethnicity, sex, and information about technical replicates were used as metadata.

Sample ID	Ethnicity	Sex	Replicates
NA19000	Japanese	male	-
NA19240	Yoruba	female	1
NA18942	Japanese	male	-
NA18853	Yoruba	male	-
NA19774	Mexican-American	male	-
NA19779	Mexican-American	female	-
HG01840	Kinh Vietnamese	male	-
HG01600	Kinh Vietnamese	female	-
NA18635	Han Chinese	male	-
NA18550	Han Chinese	female	-
HG00119	British	male	-
HG00121	British	female	-
NA12144	Utah/Mormon	male	-
NA12044	Utah/Mormon	female	-
HG01148	Colombian	male	-
HG01131	Colombian	female	-
NA20524	Tuscan	male	-
NA20517	Tuscan	female	-
HG00640	Puerto Rican	male	-
HG00638	Puerto Rican	female	-

Table E.3: Patients of the ALL and HCC cohort included in the project on the assessment of personalized vaccine options. Given HLA genotypes were predicted with OptiType.

ALL cohort			HCC cohort		
ID	Cancer entity	HLA alleles	ID	Cancer entity	HLA alleles
QA001	c-ALL	A*01:01 A*25:01 B*08:01 B*18:01 C*07:01 C*12:03	HCC023	HCC	A*29:02 A*24:02 B*44:03 B*37:01 C*06:02 C*16:01
QA002	c-ALL	A*24:02 A*25:01 B*18:01 B*15:01 C*03:03 C*12:03	HCC024	HHC	A*03:01 A*68:01 B*40:01 B*15:01 C*03:04 C*03:81
QA003	c-ALL	A*02:01 A*02:01 B*39:01 B*56:01 C*01:02 C*02:02	HCC025	HCC	A*02:01 A*11:01 B*44:02 B*37:01 C*07:04 C*06:02
QA004	c-ALL	A*32:01 A*30:09 B*35:02 B*78:01 C*16:01 C*04:01	HCC026	HCC	A*02:01 A*01:01 B*51:01 B*08:01 C*07:01 C*01:02
QA005	c-ALL	A*02:01 A*02:01 B*07:02 B*15:01 C*07:02 C*03:04	HCC027	HCC	A*24:02 A*03:01 B*18:01 B*27:05 C*07:01 C*02:02
QA006	c-ALL	A*24:02 A*02:01 B*35:01 B*51:01 C*12:03 C*14:02	HCC028	HCC	A*02:01 A*24:02 B*07:02 B*35:03 C*07:02 C*04:01
QA007	c-ALL	A*02:17 A*31:01 B*44:02 B*18:01 C*07:01 C*05:01	HCC030	HCC	A*02:01 A*03:01 B*14:01 B*27:05 C*01:02 C*08:02
QA008	c-ALL	A*31:01 A*03:01 B*49:01 B*44:02 C*07:01 C*05:01	HCC034	HCC	A*11:01 A*23:01 B*44:03 B*18:01 C*07:01 C*04:01
QA009	c-ALL	A*26:01 A*25:01 B*51:01 B*38:01 C*12:03 C*15:02	HCC035	HCC	A*68:01 A*02:01 B*27:05 B*35:03 C*02:02 C*04:01
QA010	c-ALL	A*01:01 A*30:01 B*57:01 B*13:02 C*06:02 C*06:02	HCC036	HCC	A*02:01 A*68:01 B*44:02 B*27:05 C*07:04 C*01:02
QA011	c-ALL	A*31:01 A*25:01 B*07:02 B*44:02 C*07:02 C*07:04	HCC038	HCC	A*02:01 A*02:01 B*07:02 B*07:02 C*07:02 C*07:02
QA012	c-ALL	A*01:01 A*68:01 B*57:01 B*51:01 C*01:02 C*06:02	HCC040	HCC	A*68:01 A*03:01 B*44:02 B*51:01 C*07:04 C*15:02
QA013	c-ALL	A*30:01 A*24:02 B*13:02 B*07:02 C*07:02 C*06:02	HCC041	HCC	A*01:01 A*01:01 B*08:01 B*08:01 C*07:01 C*07:01
QA014	c-ALL	A*01:01 A*24:02 B*14:02 B*07:02 C*07:02 C*08:02	HCC042	HCC	A*01:01 A*03:01 B*08:01 B*55:01 C*03:03 C*07:01
QA015	Cortical T-ALL	A*03:01 A*24:02 B*14:02 B*55:01 C*03:03 C*08:02	HCC043	HCC	A*02:01 A*01:01 B*08:01 B*40:01 C*03:04 C*07:01
QA016	Pro-B-ALL	A*24:02 A*03:01 B*08:01 B*08:01 C*07:01 C*07:01	HCC045	HCC	A*01:01 A*26:01 B*44:03 B*47:01 C*06:02 C*04:01
QA017	T-ALL	A*24:02 A*32:01 B*35:03 B*08:01 C*07:01 C*04:01			
QA018	c-ALL	A*30:01 A*02:01 B*13:02 B*38:01 C*06:02 C*12:03			
QA019	c-ALL	A*32:01 A*24:03 B*07:05 B*35:01 C*04:01 C*15:05			
QA020	c-ALL	A*02:01 A*68:01 B*40:01 B*38:01 C*03:04 C*12:03			
QD003	c-ALL	A*01:01 A*32:01 B*41:02 B*15:17 C*17:01 C*07:01			
QD004	c-ALL	A*01:01 A*23:01 B*18:01 B*40:02 C*07:01 C*02:02			
QD005	Pre B-ALL	A*02:01 A*23:01 B*44:03 B*07:02 C*07:02 C*04:01			
QD007	c-ALL	A*02:01 A*11:01 B*14:02 B*35:01 C*04:01 C*08:02			

Table E.4: Statistics for the HCC cohort. The category Var^{cns} does not include nonsense mutations. Variants with evidence on RNA level are denoted by Var^{exp}. After filtering of the complete peptide search space (PSS), remaining peptides (fPSS) were used as input for the HLA binding prediction. Resulting PNEs were checked for evidence on multiple omics levels (transcriptome, proteome, ligandome).

ID	Unf. Var	Var	Var^{ns}	Var^{cns}	Var^{exp}	TMB	PSS	fPSS	PNE	PNE^{exp}	PNE^{prot}	WT^{lig}	PNE^{lig}
HCC023	295	120	53	47	15	1.40	3,392	3,063	123	67	6	0	0
HCC024	228	116	48	45	18	1.40	2,634	2,207	191	73	6	0	0
HCC025	375	152	58	55	26	1.84	3,315	2,807	273	163	46	0	0
HCC026	338	147	73	68	45	1.96	4,010	3,246	253	140	47	0	0
HCC027	387	151	61	60	30	2.04	3,793	3,134	210	126	48	1	0
HCC028	194	146	69	68	35	1.85	8,385	6,258	356	141	0	1	0
HCC030	281	219	108	103	52	2.90	7,938	4,792	382	210	0	0	0
HCC034	258	122	54	51	20	1.50	2,992	2,705	157	72	0	0	0
HCC035	138	109	58	54	25	1.40	4,022	2,355	258	117	0	0	0
HCC036	293	136	52	50	10	1.72	2,984	2,605	258	156	6	0	0
HCC038	175	132	79	73	28	1.82	5,452	3,672	319	98	0	0	0
HCC040	198	156	73	68	34	1.82	5,975	3,760	266	124	0	0	0
HCC041	346	272	117	108	41	3.15	10,077	6,356	215	98	0	1	0
HCC042	168	128	66	64	23	1.48	5,317	3,668	188	86	0	0	0
HCC043	191	152	76	71	39	1.80	5,856	3,543	338	155	0	0	0
HCC045	215	164	80	76	33	2.20	6,874	4,120	111	63	0	0	0

Table E.5: Statistics for the ALL cohort. The category Var^{cns} does not include nonsense mutations. Variants with evidence on RNA level are denoted by Var^{exp}. After filtering of the complete peptide search space (PSS), remaining peptides (fPSS) were used as input for the HLA binding prediction. Resulting PNEs were checked for evidence on transcriptome level. For QA017 no WTS data was available.

ID	Unf. Var	Var	Var^{ns}	Var^{cns}	Var^{exp}	TMB	PSS	fPSS	PNE	PNE^{exp}
QA001	47	26	20	19	7	0.44	760	682	28	10
QA002	71	21	9	8	4	0.34	436	397	17	5
QA003	48	20	13	12	4	0.34	494	493	31	11
QA004	27	9	5	5	2	0.42	234	230	11	3
QA005	35	15	6	6	5	0.3	262	262	24	14
QA006	89	47	27	26	5	0.82	1,250	1,169	73	21
QA007	31	13	8	8	1	0.26	380	292	26	5
QA008	48	11	9	8	3	0.2	342	228	5	1
QA009	42	18	11	10	4	0.3	604	472	14	12
QA010	31	11	9	9	4	0.22	380	341	14	10
QA011	110	41	25	24	13	0.76	867	638	41	25
QA012	55	25	14	13	1	0.4	456	418	30	0
QA013	47	22	12	11	5	0.38	418	382	22	13
QA014	52	31	17	15	4	0.48	684	539	27	6
QA015	275	98	68	67	35	1.66	2,586	2,302	94	54
QA016	46	8	5	5	2	0.14	190	190	8	5
QA017	112	55	32	29	0	0.88	1,434	1,297	47	0
QA018	41	18	12	12	5	0.34	388	388	34	21
QA019	64	34	17	16	8	0.54	598	509	31	10
QA020	22	11	7	7	1	0.18	304	228	30	8
QD003	82	43	27	24	11	0.8	1084	800	39	20
QD004	52	23	12	11	2	0.42	418	266	15	1
QD005	142	83	51	47	14	1.42	2,011	1,840	176	48
QD007	47	12	5	5	1	0.2	380	380	34	1

Table E.6: List of cancer/testis (CT) antigens identified in HCC cohort. HLA class I ligands and quantified proteins were queried against a list of known CT antigens derived from CTDatabase[431] (accessed 2018-02-20). Proteome data was only available for seven patients.

Identifier		**Identified on**	
Gene	UniprotID	Proteome	Ligandome
ARMC3	Q5W041		HCC045
ATAD2	Q6PL18	HCC025	HCC023, HCC045
LDHC	P07864	HCC023, HCC024, HCC025, HCC026, HCC027, HCC034, HCC036	
MAEL	Q96JY0		HCC045
NR6A1	Q15406	HCC027	
POTEE	Q6S8J3	HCC034, HCC026, HCC027, HCC024, HCC025, HCC023, HCC036	
POTEB, POTEC	A0A0A6YYL3, B2RU33	HCC023	
POTEG, POTEH	Q6S5H5, Q6S545	HCC023, HCC026	
PRAME	P78395		HCC041
RBM46	Q8TBY0	HCC024, HCC025, HCC026	
SSX1	Q16384		HCC035, HCC041
SSX2, SSX3, SSX4, SSX6, SSX7, SSX9	Q16385, Q99909, O60224, Q7RTT6, Q7RTT5, Q7RTT3		HCC041
TEX15	Q9BXT5	HCC023, HCC024, HCC026, HCC027	
TFDP3	Q5H9I0		HCC045

Appendix F: Workflows

16S Taxonomic Profiling

Version: 1.0
Author: Christopher Mohr, Alexander Seitz

Description
Performs taxonomic profiling for 16S metagenomic samples using MALT[459].

Input
- FASTQ/FASTQ.gz(s) with 16S ribosomal RNA reads

Output
- FastQC output
- CSV with feature counts per sample
- TXT with merged feature counts (in case of replicates)

Additional software/data
- Snakemake
- ClipAndMerge 1.7.5
- MALT 0.3.8
- MALT reference index

Source code & report issues
`github.com/qbicsoftware/16Smetagenomics-taxonomic-profiling`

Differential Expression Analysis

Version: 1.0
Author: Christopher Mohr, Stefan Czemmel, Marius Codrea

Description
Performs differential expression Analysis using DESeq2[231] for a comparison of two groups, e.g. tumor vs. normal. Analysis is based on count data coming from an RNA-Seq analysis.

Input
- TXT(s) with count data (one file per group)

Output
- PDF/PNG(s) with plots
- TSV(s) with statistics like log-transformed read counts and primary DESeq2 output

Additional software/data
- R package DESeq 2.4

Source code & report issues
`github.com/qbicsoftware/differentialexpression-analysis-workflow`

EPAA – Epitope Prediction and Annotation

Version: 1.0
Author: Christopher Mohr

Description
Performs prediction of MHC class I and II epitopes for a list of annotated variants or peptides in the context of specified MHC alleles. Additionally predicted peptides can be annotated with protein quantification values for the corresponding proteins and differential expression values for the corresponding transcripts.

Input
- TXT(.alleles) with MHC alleles (one per line)
- TSV/GSvar/VCF with annotated variants of somatic and/or germline origin or peptide sequences
- TSV with MaxQuant results
- FASTA with protein sequences
- TXT with count data of RNA-Seq analysis

Output

- TSV with generated peptides annotated with MHC binding prediction values, corresponding genomic information and annotated information (optional)
- TXT with basic statistics

Parameters

Parameter	Value	Type	Description	Required	Restrictions
mhcclass	I	string	MHC class	true	I, II
identifier		string	Predictions will be written with this name prefix	false	-
filter_self	true	boolean	Filter peptides against human proteome	false	true, false
reference	GRCh38	boolean	Reference, retrieved information will be based on this Ensembl version	false	GRCh37/38

Additional software/data

- Python package FRED 2.0
- Syfpeithi
- NetMHC 4.0
- NetMHCpan 3.0
- NetMHCII 2.2
- NetMHCIIpan 3.1
- FASTA(s) with human reference protein sequences
- TSV with list of human coding genes and their length in base pairs

Source code & report issues

`https://github.com/qbicsoftware/epaa-workflow`

EPAA – Epitope Prediction and Annotation

Version: 1.1

Author: Christopher Mohr

Description

Performs prediction of MHC class I and II epitopes for a list of annotated variants or peptides in the context of specified MHC alleles. Additionally predicted peptides can be annotated with protein quantification values for the corresponding proteins, differential expression values for

the corresponding transcripts or MHC ligands of a ligandomics identification run. This version includes the functionality to generate the wild-type sequences of corresponding mutated peptides.

Input

- TXT (.alleles) with MHC alleles (one per line)
- TSV/GSvar/VCF with annotated variants of somatic and/or germline origin or peptide sequences
- TSV with MaxQuant results
- FASTA with protein sequences
- TXT with count data of RNA-Seq analysis
- CSV with identified ligands

Output

- TSV with generated peptides annotated with MHC binding prediction values, corresponding genomic information and annotated information (optional)
- TXT with basic statistics

Parameters

Parameter	Value	Type	Description	Required	Restrictions
mhcclass	I	string	MHC class	true	I, II
identifier		string	Predictions will be written with this name prefix	false	-
filter_self	true	boolean	Filter peptides against human proteome	false	true, false
wild_type	true	boolean	Add wild-type sequences of mutated peptides to output	false	true, false
reference	GRCh38	boolean	Reference, retrieved information will be based on this ensembl version	false	GRCh37/38

Additional software/data

- Python package FRED 2.0
- Syfpeithi
- NetMHC 4.0
- NetMHCpan 3.0
- NetMHCII 2.2
- NetMHCIIpan 3.1

- FASTA(s) with human reference protein sequences
- TSV with list of human coding genes and their length in base pairs

Source code & report issues
`https://github.com/qbicsoftware/epaa-workflow`

OptiType 1.0

Version: 1.0
Author: Christopher Mohr

Description
Performs HLA genotyping from NGS data using OptiType[301], a HLA genotyping algorithm based on integer linear programming. OptiType is capable of producing accurate four-digit HLA genotyping predictions from NGS data by simultaneously selecting all minor and major HLA-I alleles.

Input
- FASTQ/FASTQ.gz(s) with DNA or RNA reads

Output
- SAM(s) with mapped reads
- PDF with coverage plots of predicted solution
- TXT(.alleles) with HLA alleles of predicted solution
- TSV with best n solutions and corresponding objective function values

Parameters

Parameter	Value	Type	Description	Required	Restrictions
e	1	int	Number of enumerations	false	-
b	0.009	double	Value for beta	false	0:0.1
r	false	boolean	Map against RNA reference	false	true, false
d	true	boolean	Map against DNA reference	false	true, false

Additional software/data
- RazerS 3.4.0
- CBC 2.9.5

Source code & report issues
`https://github.com/qbicsoftware/optitype-workflow`

OptiType 1.1

Version: 1.1
Author: Christopher Mohr

Description
Performs HLA genotyping from NGS data using OptiType[301], a HLA genotyping algorithm based on integer linear programming. OptiType is capable of producing accurate four-digit HLA genotyping predictions from NGS data by simultaneously selecting all minor and major HLA-I alleles.

Input
- FASTQ/FASTQ.gz(s) with DNA or RNA reads
- BAM(s) with DNA or RNA reads

Output
- PDF with coverage plots of predicted solution
- TXT(.alleles) with HLA alleles of predicted solution
- TSV with best n solutions and corresponding objective function values

Parameters

Parameter	Value	Type	Description	Required	Restrictions
e	1	int	Number of enumerations	false	-
b	0.009	double	Value for beta	false	0:0.1
r	false	boolean	Map against RNA reference	false	true, false
d	true	boolean	Map against DNA reference	false	true, false

Additional software/data
- Yara 0.9.6
- CPLEX 12.6.2
- SeqAn 2.0.1

Source code & report issues
`https://github.com/qbicsoftware/optitype-workflow`

Individualized Proteome Generator

Version: 1.0
Author: Christopher Mohr

Description
Generates the individualized protein fasta containing protein sequences with integrated mutations based on a list of annotated variants.

Input
- TSV/GSvar with annotated variants of somatic and/or germline origin

Output
- FASTA with mutation-containing protein sequences

Parameters

Parameter	Value	Type	Description	Required	Restrictions
g	false	boolean	Include germline mutations	false	true, false
d		string	Sample ID of processed sample	true	-

Additional software/data
- Python package FRED
- FASTA(s) with human reference protein sequences

Source code & report issues
`https://github.com/qbicsoftware/qbic-workflow-indproteome`

Individualized Proteome Generator

Version: 2.0
Author: Christopher Mohr

Description

Generates the individualized protein fasta containing protein sequences with integrated mutations based on a list of annotated variants and attaches it to a chosen proteome reference.

Input

- TSV/GSvar/VCF with annotated variants of somatic and/or germline origin
- FASTA with human reference proteome

Output

- FASTA with human reference proteome and mutation-containing protein sequences

Parameters

Parameter	Value	Type	Description	Required	Restrictions
identifier		string	Predictions will be written with this name prefix	false	-
reference	GRCh38	string	Reference, retrieved information will be based on this ensembl version	false	GRCh37/38

Additional software/data

- Python package FRED 2.0

Source code & report issues

`https://github.com/qbicsoftware/qbic-workflow-indproteome`

IRMA – Epitope Prediction

Version: 1.0

Author: Christopher Mohr, Mathias Walzer

Description

Performs prediction of MHC-binding peptides for a list of annotated variants and specified MHC class I or II alleles.

Input

- TXT(.alleles) with MHC alleles (one per line)
- TSV(.GSvar) with annotated variants of somatic and/or germline origin

Output

- TSV with generated peptides annotated with MHC binding prediction values, corresponding genomic information
- TXT with basic statistics

Parameters

Parameter	Value	Type	Description	Required	Restrictions
g	false	boolean	Include germline mutations	false	true, false
m	I	string	MHC class	true	I, II
d		string	Sample ID of processed sample	true	-
u	true	boolean	Include prediction for wild-type peptides	false	true, false

Additional software/data

- Python package FRED
- Syfpeithi
- NetMHC 3.0
- NetMHCpan 2.4
- NetMHCII 2.2
- NetMHCIIpan 2.0
- FASTA with human protein sequences (RefSeq)

Source code & report issues

`https://github.com/qbicsoftware/qbic-workflow-epitopeprediction`

Ligandomics Identification

Version: 1.0

Author: Mathias Walzer, Christopher Mohr

Description

Performs MHC ligandomics data processing. The output will be filtered according to the given FDR value.

Input

- mzML/mzML.gz(s) with mass spectrometer output data
- FASTA with human reference proteome

Output

- idXML(s) with identifications
- CSV(s) with identified peptides

Parameters

Parameter	Value	Type	Description	Required	Restrictions
pmt	5.0	double	Precursor Mass Tolerance [ppm]	false	0.0:100.0
fmt	0.02	double	Fragment Mass Tolerance [Da]	true	0.0:100.0
fdr	0.05	double	False Discovery Rate [%]	true	0.0:1.0
variableMode	true	boolean	Variable Modification: M(ox)	false	true, false
centroided	true	boolean	Centroided input data	false	true, false

Additional software/data

- OpenMS 2.0-44ed56b
- Comet 2015024

Source code & report issues

`https://github.com/qbicsoftware/ligandomics-ID-workflow`

Ligandomics Identification

Version: 2.0

Author: Leon Bichmann, Christopher Mohr

Description

Performs MHC ligandomics data processing. The workflow uses Comet[428] and Percolator[429,430] in order to identify peptides. The FDR is calculated using Percolator, based on a competitive target decoy approach using reversed decoy sequences and merged identifications of all replicate runs if available. MapAlignerIdentification and FeatureFinderIdentification[460] are used for peptide identification.

Input

- mzML/mzML.gz(s) with mass spectrometer output data
- FASTA with human reference proteome or individualized proteome

Output

- featureXML with extracted features
- idXML with identifications
- CSV(s) with identified peptides

Parameters

Parameter	Value	Type	Description	Required	Restrictions
pmt	5.0	double	Precursor Mass Tolerance [ppm]	true	0.0:100.0
fmt	0.02	double	Fragment Mass Tolerance [Da]	true	0.0:1.0
fbo	0.0	double	Fragment Bin Offset	true	0.0:1.0
dmr	800:5000	string	Digest Mass Range	true	-
fdr	0.05	double	False Discovery Rate [%]	true	0.0:1.0
noh	1	int	Number of ranks	true	0:100
centroided	true	boolean	Centroided input data	false	true, false
ms_levels	1	int	MS Levels	false	1:2

Additional software/data

- OpenMS 2.2
- Percolator 3.1.1

Source code & report issues

`https://github.com/qbicsoftware/ligandomics-ID-workflow`

Ligandomics Identification

Version: 2.1
Author: Leon Bichmann, Christopher Mohr

Description

Performs MHC ligandomics data processing. The workflow uses Comet[428] and Percolator[429,430] in order to identify peptides. The FDR is calculated using Percolator, based on a competitive target decoy approach using reversed decoy sequences and merged identifications of all replicate runs if available. MapAlignerIdentification and FeatureFinderIdentification[460] are used for peptide identification.

Input

- mzML/mzML.gz(s) with mass spectrometer output data
- FASTA with human reference proteome or individualized proteome

Output

- featureXML with extracted features
- idXML with identifications
- CSV(s) with identified peptides

Parameters

Parameter	Value	Type	Description	Required	Restrictions
pmt	5.0	double	Precursor Mass Tolerance [ppm]	true	0.0:100.0
fmt	0.02	double	Fragment Mass Tolerance [Da]	true	0.0:1.0
fbo	0.0	double	Fragment Bin Offset	true	0.0:1.0
dmr	800:5000	string	Digest Mass Range	true	-
fdr	0.05	double	False Discovery Rate [%]	true	0.0:1.0
noh	1	int	Number of ranks	true	0:100
centroided	true	boolean	Centroided input data	false	true, false
ms_levels	1	int	MS Levels	false	1:2
maxmod	5	int	Maximum number of modifications	false	0:10
fixed_mod1	false	string	Carbamidomethyl (C) (Fixed)	false	true,false
variable_mod1	true	string	Oxidation (M) (Variable)	false	true,false
variable_mod2	true	string	Phospho (S) (Variable)	false	true,false
variable_mod3	true	string	Phospho (T) (Variable)	false	true,false
variable_mod4	true	string	Phospho (Y) (Variable)	false	true,false
pred_charge	2:3	string	Precursor charge	false	
activ_method	HCD	string	Activation Method	false	HCD:CID

Additional software/data

- OpenMS 2.3
- Percolator 3.1.1

Source code & report issues
`https://github.com/qbicsoftware/ligandomics-ID-workflow-2_1`

Ligandomics Identification Co-Processing

Version: 1.0
Author: Leon Bichmann, Christopher Mohr

Description
Performs MHC ligandomics data processing. The workflow handles co-processing of biological/technical replicates and uses Comet[428] and Percolator[429,430] in order to identify peptides. The FDR is calculated using Percolator, based on a competitive target decoy approach using reversed decoy sequences and merged identifications of all replicate runs if available. MapAlignerIdentification and FeatureFinderIdentification[460] are used for peptide identification.

Input
- mzML/mzML.gz(s) with mass spectrometer output data
- FASTA with human reference proteome or individualized proteome

Parameters

Parameter	Value	Type	Description	Required	Restrictions
pmt	5.0	double	Precursor Mass Tolerance [ppm]	true	0.0:100.0
fmt	0.02	double	Fragment Mass Tolerance [Da]	true	0.0:1.0
fbo	0.0	double	Fragment Bin Offset	true	0.0:1.0
dmr	800:5000	string	Digest Mass Range	true	-
fdr	0.05	double	False Discovery Rate [%]	true	0.0:1.0
noh	1	int	Number of ranks	true	0:100
centroided	true	boolean	Centroided input data	false	true, false
ms_levels	1	int	MS Levels	false	1:2

Output

- featureXML with extracted features
- idXML with identifications
- CSV(s) with identified peptides

Additional software/data

- OpenMS 2.2
- Percolator 3.1.1

Source code & report issues

`https://github.com/qbicsoftware/ligandomics-ID-workflow-copro`

Ligandomics Identification Co-Processing

Version: 2.1
Author: Leon Bichmann, Christopher Mohr

Description

Performs MHC ligandomics data processing. The workflow handles co-processing of biological/technical replicates and uses Comet[428] and Percolator[429,430] in order to identify peptides. The FDR is calculated using Percolator, based on a competitive target decoy approach using reversed decoy sequences and merged identifications of all replicate runs if available. MapAlignerIdentification and FeatureFinderIdentification[460] are used for peptide identification.

Input

- mzML/mzML.gz(s) with mass spectrometer output data
- FASTA with human reference proteome or individualized proteome

Output

- featureXML with extracted features
- idXML with identifications
- CSV(s) with identified peptides

Additional software/data

- OpenMS 2.3
- Percolator 3.1.1

Parameters

Parameter	Value	Type	Description	Required	Restrictions
pmt	5.0	double	Precursor Mass Tolerance [ppm]	true	0.0:100.0
fmt	0.02	double	Fragment Mass Tolerance [Da]	true	0.0:1.0
fbo	0.0	double	Fragment Bin Offset	true	0.0:1.0
dmr	800:5000	string	Digest Mass Range	true	-
fdr	0.05	double	False Discovery Rate [%]	true	0.0:1.0
noh	1	int	Number of ranks	true	0:100
centroided	true	boolean	Centroided input data	false	true, false
ms_levels	1	int	MS Levels	false	1:2
maxmod	5	int	Maximum number of modifications	false	0:10
fixed_mod1	false	string	Carbamidomethyl (C) (Fixed)	false	true,false
variable_mod1	true	string	Oxidation (M) (Variable)	false	true,false
variable_mod2	true	string	Phospho (S) (Variable)	false	true,false
variable_mod3	true	string	Phospho (T) (Variable)	false	true,false
variable_mod4	true	string	Phospho (Y) (Variable)	false	true,false
pred_charge	2:3	string	Precursor charge	false	
activ_method	HCD	string	Activation Method	false	HCD:CID

Source code & report issues

`github.com/qbicsoftware/ligandomics-ID-workflow-copro_2_1`

Ligandomics Quality Control

Version: 1.0

Author: Mathias Walzer, Christopher Mohr

Description

Performs quality control for MHC ligandomics data, which will be reported as a qcML file.

Input

- mzML/mzML.gz(s) with mass spectrometer output data
- TXT (.alleles) with MHC alleles (one per line)

Output

- qcML(s) with quality measures and plots

Parameters

Parameter	Value	Type	Description	Required	Restrictions
MHC class	I	string	Predictions for MHC class	false	I,II
centroided	true	boolean	Centroided input data	false	true,false

Additional software/data

- OpenMS 2.0-44ed56b
- NetMHCpan 3.0
- NetMHCIIpan 3.1
- Comet 2015024
- R 3.2.2
- R scripts to generate figures
- FASTA with human protein sequences and decoys

Source code & report issues

`https://github.com/qbicsoftware/ligandomics-QC-workflow`

Merge NGS data

Version: 1.0
Author: Christopher Mohr

Description

Merges forward/reverse reads of NGS data in fastq format (e.g. of different lanes or replicates).

Input

- FASTQ/FASTQ.GZ(s) with forward and reverse reads respectively

Output

- FASTQ(s) with merged forward and reverse reads respectively

Source code & report issues
`https://github.com/qbicsoftware/merge-NGSdata-workflow`

NGS Quality Control

Version: 1.0
Author: Adrian Seyboldt, David Wojnar, Christopher Mohr

Description
Performs quality control for fastq files using FastQC.

Input
- FASTQ/FASTQ.gz with NGS reads

Output
- ZIP with FastQC output
- HTML with FastQC quality report

Additional software/data
- Snakemake
- FastQC 0.11.4

Source code & report issues
`https://github.com/qbicsoftware/ngsqc`

NGS Read Alignment

Version: 1.0
Author: Marc Sturm, Adrian Seyboldt, Stefan Czemmel, Christopher Mohr

Description
Performs NGS read alignment against a specified genome reference using BWA.

Input
- FASTQ/FASTQ.gz(s) with NGS reads
- BWA indexed reference genome

Output
- FastQC output

- BAM(s) with mapped and unmapped reads
- BAI(s) with BAM index

Additional software/data

- Snakemake
- SAMtools 1.2
- BWA 0.7.10
- FastQC 0.11.4
- Picard 1.140
- NGS-bits 0.1

Source code & report issues

`https://github.com/qbicsoftware/mapping`

Protein Quantification

Version: 1.0
Author: David Wojnar, Adrian Seyboldt, Christopher Mohr

Description

Performs protein quantification using MaxQuant[426], which is a quantitative proteomics software package designed for analyzing large-scale mass-spectrometric data sets. It supports all main labeling techniques like SILAC, Di-methyl, TMT and iTRAQ as well as label-free quantification.

Input

- RAW(s) with raw mass spectrometry data

Output

- MaxQuant output

Parameters

MaxQuant Parameters

Additional software/data

- MaxQuant 1.5.0.0
- mqrun

Source code & report issues
`https://github.com/qbicsoftware/mqrun`

RNA-Seq Data Analysis

Version: 1.1
Author: Marc Sturm, Adrian Seyboldt, Stefan Czemmel, Christopher Mohr

Description
Performs RNA-Seq data analysis using TopHat2[207], for gapped-read mapping and HTSeq[227] to generate feature counts.

Input
- FASTQ/FASTQ.gz(s) with RNA reads
- BOWTIE indexed reference genome

Output
- FastQC output
- CSV with per sample feature counts
- TXT with merged feature counts (in case of replicates)

Parameters

Parameter	Value	Type	Description	Required	Restrictions
stranded	yes	string	Whether the data is from a strand-specific assay	false	yes, no, reverse
overlap_mode	union	string	Mode to handle reads overlapping more than one feature	false	union, intersection-strict, intersection–nonempty
gff_attribute	gene_id	string	GFF attribute to be used as feature ID	false	gene_id, transcript_id
feature_type	CDS	string	Feature type (3rd column in GFF file) to be used	false	exon, CDS
order	name	string	The alignments have to be sorted either by read name or by alignment position	false	name, pos

Additional software/data

- Snakemake
- FastQC 0.10
- TopHat 2.1.1
- HTSseq 0.6.1p2
- SAMtools 1.2

Source code & report issues

`https://github.com/qbicsoftware/rnamapping`

shRNA Counting

Version: 1.0

Author: Adrian Seyboldt , Christopher Mohr

Description

Performs counting of short hairpin (sh) RNA expression. Given a set of reads from shRNA sequencing, we count how often each reference shRNA sequence occurs at the expected position in the reads. Each read should contain a barcode at a specified position.

Parameters

Parameter	Value	Type	Description	Required	Restrictions
loop sequence		string	Loop sequence which should be in every read	true	
adapter		string	A list of possible adapters.	true	
oligo offset	-20	int	Offset for the oligo sequences relative to the loop location	true	
oligo length	19	int	Oligo length	false	
barcode offset	19	int	Offset for the barcode sequences relative to the loop location	true	
barcode length	3	int	Length of the barcode sequences	false	
adapter offsets		string	Offsets for the adapter sequences relative to the loop location.	true	
expected loop index	20	int	Expected loop index	true	

Input

- FASTQ/FASTQ.gz with RNA-Seq reads
- TSV with shRNA library sequences (and identifiers)
- TSV with barcode sequences (and identifiers)

Output

- idXML(s) with identified peptides
- xlsx with sequence counts (sorted and unsorted)
- HTML with a report including statistics and plots
- JSON with stats

Source code & report issues

`https://github.com/qbicsoftware/rnacount`

Somatic Variant Calling

Version: 1.0
Author: Christopher Mohr

Description

Performs somatic variant calling for tumor and normal tissue samples using Strelka[20]. Strelka is an analysis package designed to detect somatic SNVs and small indels from the aligned sequencing reads of matched tumor-normal samples.

Input

- BAM(s) with DNA reads
- FASTA with reference genome

Output

- VCF(s) with all somatic inDels/SNVs
- VCF(s) with filtered somatic inDels/SNVs

Parameters

Parameter	Value	Type	Description	Required	Restrictions
read_mapper	bwa	string	Read mapper used to generate bam files	true	bwa, eland, isaac

Additional software/data

- Strelka 1.0.14
- VCFlib 0.1
- SAMtools 1.3

Source code & report issues

`https://github.com/qbicsoftware/somatic-variantcalling-workflow`

Variant Annotation

Version: 1.0
Author: Christopher Mohr

Description

Performs annotation of genetic variants in VCF format using ANNOVAR[220].

Input

- VCF(s) with genomic variants

Output

- VCF(s) with annotated genomic variants

Parameters

Parameter	Value	Type	Description	Required	Restrictions
v	other	string	Variants called by	true	other, Strelka

Additional software/data

- ANNOVAR 2014–11–12

Source code & report issues

`https://github.com/qbicsoftware/variant-annotation-workflow`

Variant Annotation

Version: 2.0
Author: Christopher Mohr

Description

Performs annotation of genetic variants in VCF using SnpEff[221].

Input

- VCF/VCF.gz(s) with genomic variants

Output

- VCF(s) with filterd and unfiltered list of annotated genomic variants
- TSV(s) with filterd and unfiltered list of annotated genomic variants
- TXT with gene-based statistics
- HTML with SnpEff summary

Parameters

Parameter	Value	Type	Description	Required	Restrictions
reference	hg19	string	Reference genome	true	GRCh37.75, GRCh38.81, hg19, hg38, GRCm38.79, mm10

Additional software/data

- SnpEff 4.1k

Source code & report issues

`https://github.com/qbicsoftware/variant-annotation-workflow`

Variant Detection

Version: 1.0

Author: Marc Sturm, Adrian Seyboldt, Stefan Czemmel, Christopher Mohr

Description

Performs variant detection using FreeBayes[212], a bayesian genetic variant detector.

Input

- BAM(s) with DNA reads
- BAI(s) with index of BAM files
- BWA indexed reference genome

Output

- VCF(s) with detected variants

Additional software/data

- Snakemake
- SAMtools 1.3
- NGS-bits 0.1
- BWA 0.7.10
- Stampy 1.0.27
- Picard 1.140
- GATK 3.3
- FreeBayes 0.9
- VCFlib 0.1
- BCFtools 1.2
- VCFtools 0.1.13

Source code & report issues

`https://github.com/qbicsoftware/variantcalling`

Appendix G: Supporting Listings

```
{"workflow": {
   "description": "Description of workflow",
   "experimenttype": "Q_WF_NGS_HLATYPING",
   "filedirectory": "/path/to/workflow/optitype_v1_1_2016",
   "id": "optitype_v1_1_2016",
   "name": "OptiType",
   "nodes": [
      {
         "cmdparams": [
            ...
            "WORKFLOW-CTD"
         ],
         ...
         "inputports": [
            {
               "description": "a ctd parameter file that contains ctd enabled tools of the workflow.",
               "name": "WORKFLOW-CTD",
               "parameters": [
                  {
                     "default": 0.009,
                     "description": "Value for beta",
                     "name": "OptiType.1.b",
                     "range": [0, 0.1],
                     "required": true,
                     "type": "Float"
                  }
                  ...
               ],
               "portnumber": 6,
               "type": "CTD"
            },
            ...
            {
               "description": "CTD containing files to stage",
               "name": "IN-FILESTOSTAGE",
               "parameters": [
                  {
                     "default": "",
                     "description": "Bam file containing (mapped) DNA/RNA reads.",
                     "name": "InputFiles.1.bam",
                     "range": ["Q_NGS_MAPPING_DATA"],
                     "required": false,
                     "type": "File"
                  },
                  ...
               ],
               "portnumber": 0,
               "type": "FILESTOSTAGE"
            }
         ],
         ...
      }
   ],
   "sampletype": "Q_WF_NGS_HLATYPING",
   "version": "1.1"
}}
```

Listing 3: Simplified workflow configuration file. The JSON-based configuration file contains general information, such as a description, the name, and the identifier. For each workflow node, information about every parameter is given as defined in the corresponding CTD file.