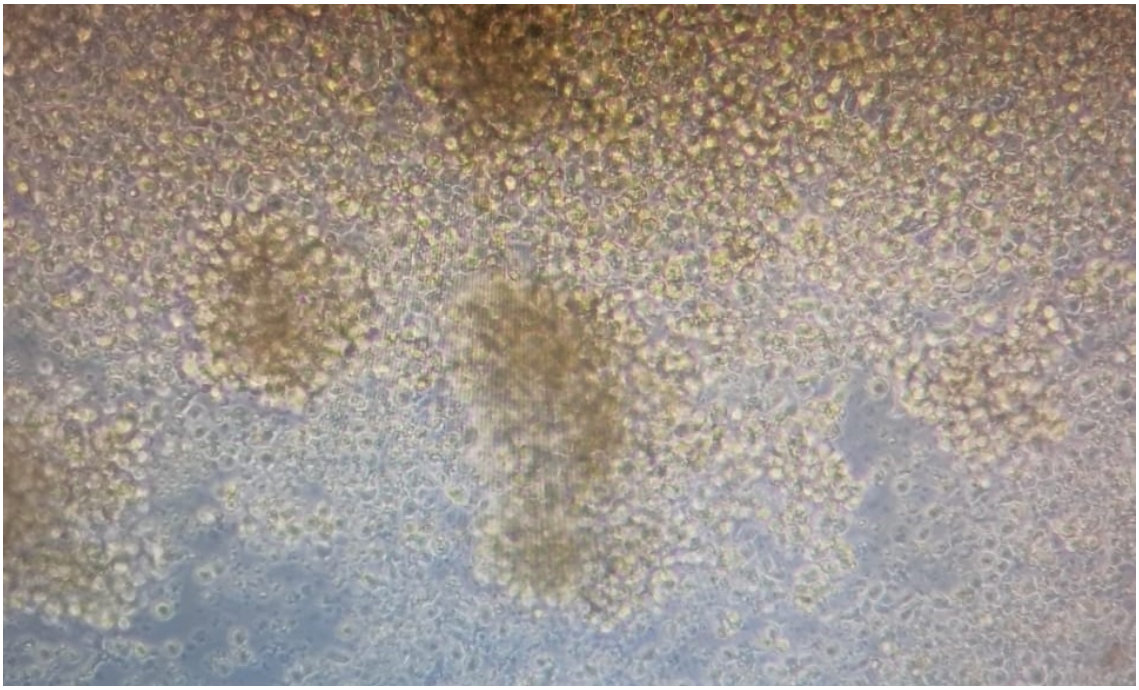
INSTITUTO POLITÉCNICO DE LISBOA

Instituto Superior de Engenharia de Lisboa
Escola Superior de Tecnologia da Saúde de Lisboa



Construction of a decision support system for the effectiveness of Immune Cell Therapies

Pedro Noronha de Castro Machado Teixeira

Trabalho Final de Mestrado para obtenção do grau de
Mestre em Engenharia Biomédica

Orientador

Nuno Alexandre Soares Domingues (ISEL)

Júri

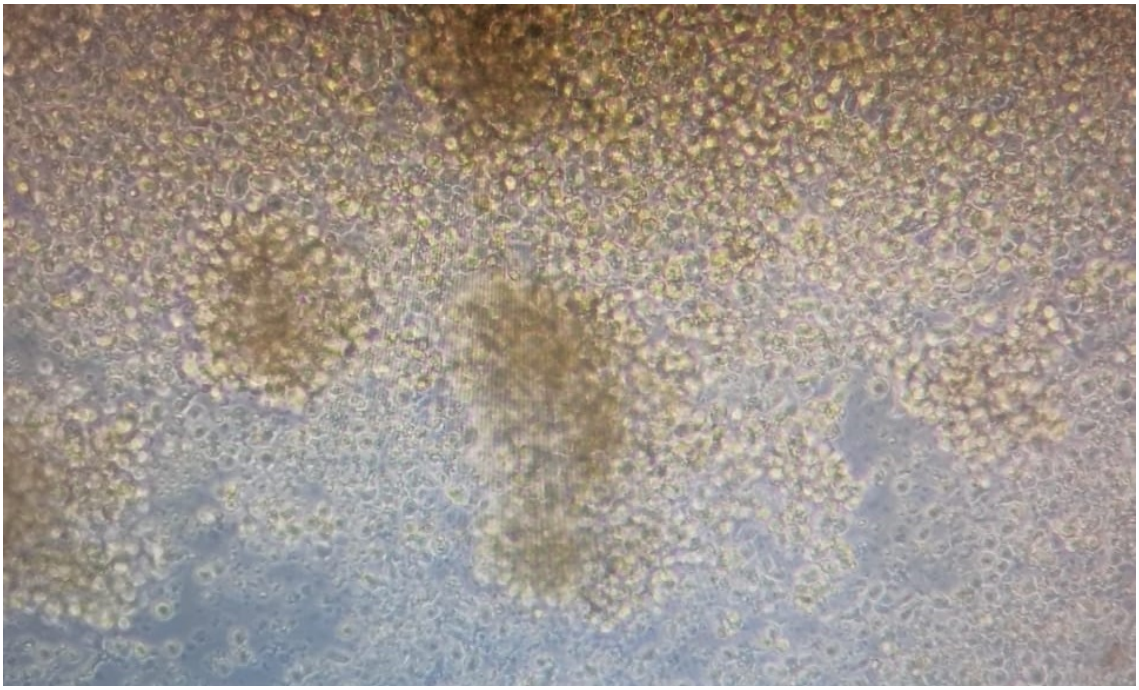
Presidente: Luís Miguel Minhalma

Vogais: Luca Gattinoni
Nuno Alexandre Soares Domingues

Setembro de 2024

INSTITUTO POLITÉCNICO DE LISBOA

Instituto Superior de Engenharia de Lisboa
Escola Superior de Tecnologia da Saúde de Lisboa



Construction of a decision support system for the effectiveness of Immune Cell Therapies

Pedro Noronha de Castro Machado Teixeira

Trabalho Final de Mestrado para obtenção do grau de
Mestre em Engenharia Biomédica

Orientador

Nuno Alexandre Soares Domingues (ISEL)

Júri

Presidente: Luís Miguel Minhalma

Vogais: Luca Gattinoni
Nuno Alexandre Soares Domingues

Setembro de 2024

Construction of a decision support system for the effectiveness of Immune Cell Therapies

This document was created with the (pdf/Xe/Lua)LaTeX processor and the NOVAthesis template (v7.1.18) [1].

Acknowledgements

The development of this final master's work was possible thanks to the support, help and encouragement of several people, whom I would like to thank.

I would like to thank my supervisor, Professor Nuno Domingues, for helping me to find the support I needed to verify and correct the proposed pipeline.

I really appreciate all the dedication and help you gave me in this work.

To the Immunosurgery Group, for their help in designing the experimental plan for this work and for their availability and helpfulness in clarifying any doubts.

To the Functional Immune Cell Modulation Group at the Leibniz Institute of Immunotherapy for their precise help in clarifying practical and theoretical doubts during the methodology, specifically within the *in vitro* experiments and in reviewing the writing of the monography.

In particular, I would like to thank João Correia, a great companion in this final phase and with whom I shared many hours during the development of this work.

The last word goes to my family and friends, who have always given me the necessary support and motivation.

To my co-workers who always gave me the support and motivation I needed to realise this adventure, even in the most difficult times.

Finally, special thanks to my mother for her unconditional support in this and every phase of my life.

I dedicate this work to her.

Thank you all very much

Abstract

The ability to integrate immune responses from tumour infiltrating lymphocytes, TILs, whole blood assays (WBA) from flow cytometry data with the secretome through cytokine quantification using ELISA, makes it possible to optimise immune cell therapies. A centralised repository allows clinicians and researchers to unravel the complexities of immune responses triggered by various treatments in different patient groups. It promotes understanding of how immune cell therapies orchestrate their therapeutical effect.

The aggregation of immunoreaction profiles paves the way for the identification of biomarkers that can anticipate the success of immune therapy interventions. In an era of personalised medicine, where healthcare providers can tailor immunotherapies according to the patient's unique immunological landscape, therefore an organised storage of this data is required. In addition, longitudinal monitoring of immune responses provides information on the durability and evolution of therapeutic efficacy, revealing potential resistance mechanisms that may develop over time.

As a proof of concept, two main assays using peripheral blood and tumour-infiltrating lymphocytes from patients with solid tumours, have been developed and optimised. Firstly, by placing immune cells from these two sources with synthetic peptides that are the same as the peptide sequences expressed in the solid tumor or patient-specific tumors, we can reveal functionality through the production of interferon gamma, IFNγ obtained through an ELISA assay and through the staining of intracellular cytokine staining carried out using flow cytometry. Secondly, evaluate the immune phenotype of TILs pre and post expansion and cytokine modulation and PBMCs phenotype to observe any changes. Subsequently, we performed statistical comparison tests to find any correlation between the phenotype of the immune cells and the detection of synthetic peptides via positive interferon-gamma production to reveal the potential effector functionality of the

immune cells.

Essentially, the integration of TIL, WBA data with ELISA and flow cytometry techniques in a decision support system not only speeds up the refinement of immune cells that have potential, but also constitutes a promising advance in the treatment of patients, because it leverages the advancement of personalised medicine. The aim of this study is to help develop accurate and effective analyses of ELISA and flow cytometry assays with the potential to reveal treatment options, which offers renewed hope for patients battling cancer and other immune-mediated diseases. This study found a relationship between the Survivin mix (1-33), Survivin (97-111) and Mesothelin MPF-11 and the selected T cells, which suggests a degree of correlation. Thus, there is a potential functionality of the T cells selected from these epithelial tumour patient samples in response to a list of 30 test peptides.

Keywords: Adoptive Cell Therapy, Tumor Infiltrating Lymphocytes, Python, Tumor Associated Antigens,

Resumo

A capacidade de integrar respostas imunológicas de linfócitos infiltradores de tumores, TILs, ensaios de sangue total (WBA) provenientes de dados de citometria de fluxo com o secretoma por quantificação de citocinas através da técnica de ELISA permite otimizar as terapias com células imunitárias. Um repositório centralizado torna possível aos investigadores e clínicos desvendar as complexidades das respostas imunitárias desencadeadas por vários tratamentos em diversos grupos de doentes. Promove a compreensão da forma como as terapias com células imunitárias orquestram os seus efeitos terapêuticos.

A agregação de perfis de imunorreação abre caminho à identificação de biomarcadores preditivos que podem antecipar o sucesso das intervenções de terapia imunitária. Numa era de medicina personalizada, em que os prestadores de cuidados de saúde podem adaptar as imunoterapias de acordo com a paisagem imunológica única de cada doente, é necessário um armazenamento organizado destes dados. Além disso, o acompanhamento longitudinal das respostas imunitárias oferece informações sobre a durabilidade e evolução da eficácia terapêutica, revelando potenciais mecanismos de resistência que se podem desenvolver ao longo do tempo.

Como prova de conceito, foram desenvolvidos e optimizados dois ensaios principais utilizando sangue periférico e linfócitos infiltrantes de tumores de doentes com tumores sólidos. Em primeiro lugar, ao colocar células imunitárias destas duas fontes em contacto com peptídeos sintéticos que são iguais às sequências peptídicas expressas no tumor sólido ou em tumores específicos do paciente, podemos revelar a funcionalidade através da produção de interferão gama, IFNγ obtido através de um ensaio ELISA e através da coloração de citocinas intracelulares efectuada por citometria de fluxo. Em segundo lugar, avaliámos o fenótipo imunitário das TILs antes e depois da expansão por modulação de citocinas e

o fenótipo das PBMCs, para observar quaisquer alterações. Posteriormente, realizámos testes de comparação estatística para encontrar qualquer correlação entre o fenótipo das células imunitárias e a deteccao dos peptídeos sintéticos com produção positiva de interferão-gama, para revelar a potencial funcionalidade efectora das células imunitárias.

Essencialmente, a integração de dados de TIL e WBA com técnicas ELISA e de citometria de fluxo num sistema de apoio à decisão não só acelera o refinamento das células imunitárias com potencial, como também constitui um avanço promissor na identificação de células imunitárias com potencial, mas também constitui um avanço promissor no tratamento dos doentes, porque alavanca o avanço da medicina personalizada. Este estudo tem como objetivo ajudar a desenvolver um armazenamento de análises precisas e eficazes dos ensaios ELISA e de citometria de fluxo com potencial de revelar opções de tratamento, que oferece uma esperança renovada para doentes que lutam contra o cancro e outras doenças imuno-mediadas.

Este estudo encontrou uma relação entre a Survivin mix (1-33), a Survivin (97-111) e a Mesotelina MPF-11 e as células T seleccionadas, o que sugere um grau de correlação. Assim, existe uma potencial funcionalidade das células T selecionadas destas amostras de pacientes com tumor epitelial em resposta a uma lista de 30 péptidos sujeitos a ensaio.

Palavras-chave: Terapia Celular Adotiva, Linfócitos infiltradores de tumores, Python, Antigénios Associados a Tumores

CONTENTS

List of Figures

List of Tables

Acronyms

ACT	Adoptive Cell Therapy
APC	antigen presenting cells
CAR	Chimeric Antigen Receptor
CD	Cluster of Differentiation
CTLA-4	cytotoxic T lymphocyte-associated antigen 4
ELISA	Enzyme-Linked Immunosorbent Assay
EMA	European Medicines Agency
ER	endoplasmic reticulum
FDA	Food and Drug Administration
GUI	graphical user interface
Gy	Gray
HIV-1	Human Immunodeficiency Virus type 1
HLA	Human Leukocyte Antigen
ICS	Intracellular Cytokine Staining
IFNγ	interpheron gamma
IL-15	Interleukin-15
IL-2	Interleukin-2
IL-21	Interleukin-21
IL-7	Interleukin-7
LAG-3	lymphocyte activation gene 3
LAMP-I	lysosomal-associated membrane protein-I
MFI	mean fluorescence intensity
MHC	Major Histocompatibility Complex
MHC-I	Major Histocompatibility Complex I
MM	Multiple Myeloma
NCI	National Cancer Institute
OD	Optical Density

PBMC Peripheral blood mononuclear cells
PCR Polymerase Chain Reaction
PD-1 programmed cell death 1
PHA Phytohaemagglutinin
RNA-seq RNA sequencing
SNP single-nucleotide polymorphism
TAA tumor associated antigens
TAP transporter associated with antigen processing
TCR T-cell receptor
TIL Tumor-Infiltrating Lymphocytes
TIM-3 T-cell immunoglobulin mucin domain 3
TNFα Tumor Necrosis Factor-Alpha
WBA Whole Blood Assay
WGS whole genome sequencing
WHO World Health Organization
WT wild type
WXS whole exome sequencing

1

Introduction

1.1 Cancer and Immune Cell Therapy

Cancer is the second leading cause of death worldwide. The World Health Organization (WHO) reported that in 2018, approximately 18.1 million people were diagnosed with cancer, resulting in 9.6 million deaths [2].

Unfortunately, the outlook for cancer is bleak, with 29-37 million new cases predicted by 2040[3].

Despite the development of numerous anti-cancer therapies, tumor cells tend to persist and can lead to local recurrence or metastasis due to their high adaptability and ability to develop mechanisms to evade immune surveillance.

Therefore, Immune Cell therapy is an emerging field in medicine that holds great promise in the treatment of cancer and autoimmune diseases.

This innovative approach involves using living cells to target and eliminate cancer cells or regulate the immune system in autoimmune diseases, providing a new avenue for therapeutic intervention.

Unlike traditional treatments such as surgery, chemotherapy, and radiation therapy that directly target cancer cells or suppress the immune system, cell therapy takes advantage of the body's own immune system and natural defenses to fight against the disease.

By harnessing the power of the immune system, cell therapy has the potential to offer long-lasting and durable responses in patients with cancer and autoimmune diseases.

However, these therapies always need *in vitro* and *in vivo* validation before they can be confirmed for preclinical testing.[4]

One of the most important validations is the immune cell phenotype and functional assessment of these cells when in contact with tumor antigens or tumor tissues.

1.2 Main Problem

The European Medicines Agency (EMA) recommendations for the manufacture and validation of cellular products used for adoptive cell therapy require validation in the preclinical phase by validating cell phenotype and functional properties in accordance with the relevant guidelines for advanced therapy medicinal products.[5]

Current evidence, as outlined in sections below, suggests that a specific phenotype of a given T-cell may have a positive or negative prognostic effect and the ability to control the tumor and thus become a durable cell therapy treatment.

However, these therapies are being developed based on acquired scientific knowledge and T-cells, as demonstrated above, have maturation and differentiation phases, so they are not static doses, but cells with a degree of complexity and types of plasticity that are still unknown in a tumor environment.

However, the construction of a cytometry panel that well characterises the differentiation and maturation phases of T cells, as well as homing markers, may be essential for a longitudinal clinical study to assess the beneficial or adverse effect and extract scientific evidence. In addition to cytometry panels, measuring interpheron gamma (IFNγ) levels when a cell product is in contact with common tumor peptides or neopeptides provides data on the type of functionality of the product.

To date, and in a small laboratory context, there is no specifically designed programmed pipeline that can combine cytometry and Enzyme-Linked Immunosorbent Assay (ELISA) assays and draw conclusions.

A solution that could do this would have added value because it would reduce the time needed to analyse assays and draw conclusions without relying on an advanced statistical comparison programme. It would reduce the cost of software subscriptions with annual licences and provide comparisons between immunocellular phenotype and functionality.

1.3 Main Goals

Given the problem outlined above, the general aim of the study is to design a laboratory assay to obtain characterisation of the immune cell phenotype and the mode of expression of functionality. After the practical implementation, the work will involve the development of Python scripts that can extract the data obtained in the laboratory for subsequent statistical analysis, also using other Python scripts.

In addition, the aim is to provide the laboratory with an easy-to-use tool that can be used for subsequent tests involving comparisons with phenotyping tests and functionality tests in the context of cellular immunotherapies. The discussion of this work is based on the importance of the markers selected and the type of functional response recorded, as well as observations at the level of T cells in the peripheral circulation and tumor-infiltrating T cells.

The different objectives are as follows:

- (**1**) - Design a practical laboratory assay to obtain the immune cell phenotype and the type of functional response in terms of IFNγ production.
- (**2**) - Design two scripts for data analysis and extraction:
  - **2.1** - one to read ELISA plate Optical Density values and analyse via four parameter logistics curve and report cytokine concentration values in pg/mL;
  - **2.2** - one to extract cytometry data previously analysed using commercial software - FlowJo^{TM};
- (**3**) - Design and implement three scripts for data analysis with statistical packages:
  - **3.1** - one that compares the relative frequencies of the cell phenotype obtained from the cytometry assay with the relative frequency of IFNγ obtained from the ELISA assay;
  - **3.2** - one which obtains the correlation between cytotoxic T cells, regulatory T cells and gamma delta T cells from the cytometry assay obtained with the relative frequency of IFNγ obtained in the ELISA assay;
  - **3.3** - one, which checks whether there is a significant difference between the Th1 and Th2 populations of T cells from blood and T cells from a tumor.
- (**4**) - Present the conclusions drawn, major limitations of the conducted study and final considerations suggesting future developments.

1.4 Research Question

Is there a correlation between the type of immune cell phenotype and the amount of IFNγ produced by immune cells, specifically CD4, CD8 and gamma delta T-cells when encountering tumor antigens?

This work explores this question by analyzing the immunogenicity of epitopes, via IFNγ production, and immune cell phenotypes to explore potential use of immune cells in the context of adoptive cell therapy.

1.5 Dissertation Structure

This thesis is divided into five chapters to answer the following questions.

Chapter 1 - Introduction

This chapter briefly introduces the subject of the thesis. It also presents the problem, main goals, the research question and thesis structure.

Chapter 2 - Literature Review and Theoretical Background

The aim of Chapter 2 is to explore the topic of the dissertation in greater depth by providing a brief introduction to immunology, the types of adaptive cell therapies and finally a focus on tumor antigens. The experimental methods used are also presented, as well as the references in the literature on the positive and negative effects of the different types of immune cell phenotypes in the tumor context, especially in the context of TIL with scientific evidence, culminating in the problems of the topic. Lastly, to reinforce the need for a pipeline tool to improve the validation of immune cell therapies.

Chapter 3 - Materials and Methods

Chapter 3 presents the study design, identifying the variables collected and the statistical approach used in the scripts developed. It also briefly explains It also briefly explains the experimental study protocol created to obtain the data.

Chapter 4 - Results and discussion

Chapter 4 begins by characterising the samples obtained, taking into account the type of functional response and the cellular phenotypes that show the most clinical evidence. It then presents an analysis of the impact on modelling of TIL immune cells versus PBMCs to assess whether there are differences between peripheral immune cells and immune cells located close to the tumor. This was followed by an evaluation of the correlation between variations in cell phenotype means. The results will be discussed.

Chapter 5 - Conclusion and Future work

Chapter 5 aims to respond to the aims of the thesis presented in Chapter 2, to suggest considerations for future research.

2

Literature Review and Theoretical Background

This literature review introduces the topic of cell therapy, the treatment of liquid and solid tumors by T cells that mediate specific anti-tumor immune responses, and the type of differentiation and maturation that is associated with the ability to control the tumor and thus lead to a more positive prognosis.

Finally, the way in which tumor antigens are classified and the way in which the functionality of T cells is monitored using functional assays will be reviewed more explicitly, expression of interferon gamma IFNγ or other cytokines that can reveal the type of modelling that T cells undergo. These approaches allow the improvement of the type of immunotherapy to be used.

The concepts here presented will allow the comprehension of the type of sample collection and experimental analysis and to design the type of analysis for a pipeline that will serve to functionally validate T cells in contact with antigens.

2.1 Immune System and Adaptive Immunity

The immune system is a complex defense mechanism that protects the body against external factors such as toxins, pathogenic organisms (including bacteria, fungi, parasites, viruses), and cancer cells.

It is composed of two major subgroups: the innate immune system and the adaptive immune system. These two components have their own advantages and disadvantages but complement and help each other.[6] The innate immune system is the body's first line of defense against pathogens. It includes physical barriers (such as skin and mucous membranes), chemical barriers (such as antimicrobial peptides), and cellular components (such as neutrophils, macrophages, and dendritic cells).[7]

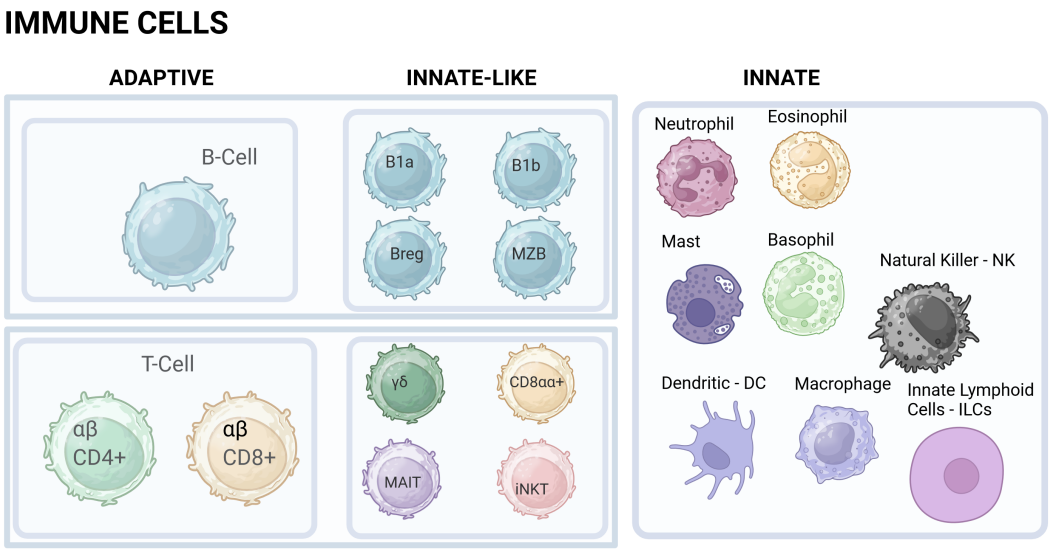


Figure 2.1: Different Immune Cells present in Human Body - classification

The adaptive immune system, on the other hand, is more specialized and involves the activation of B-cells and T-cells to produce specific antibodies and cellular responses against pathogens.[8]

In addition to these two major subgroups, there are also non-canonical immune cell subsets known as innate-like immune cells. These cells display properties belonging to both the innate and adaptive immune systems, as it can bee seen in figure 2.1. [9] The immune system must be able to discriminate between beneficial and detrimental organisms, prevent excessive reactions against external factors, differentiate between pathogenic organisms and host cells or tissues, and initiate and terminate responses when a pathogen is detected and cleared.[10]

This intricate balance is maintained through the interplay between the innate and adaptive immune systems, as well as the involvement of various immune cell types, such as helper Th, $CD4^+$ and cytotoxic, $CD8^+$ T-cells, B-cells, neutrophils, and innate lymphoid cells (ILCs).[11] Recent studies have shown that the immune system plays a role in tissue homeostasis, regeneration, and repair, in addition to protecting against pathogens.[12]

There is evidence of the enrollment of the immune system in patient prognosis.

For example, an immune response may be associated with the prevention and treatment of transfusion-related acute lung injury (TRALI) through immunomodulatory strategies to inhibit excessive immune system activation[13]. And in Multiple Myeloma (MM) cells impede the establishment of a potent immune response against MM.

The efficacy of immunotherapies in treating MM patients has been validated through the discovery of the graft-versus-myeloma effect observed in allogeneic bone marrow transplantation situations.[14]

The immune system has a dual role as protector and cause of disease. In the case of bacterial infections, the immune system should be targeted therapeutically to inhibit over-reactivity.[15]

Understanding the complex interactions between the innate and adaptive immune systems, as well as the role of non-canonical immune cell subsets, is crucial for developing novel therapeutic interventions for various immune-related diseases and conditions.[16]

Research on the immune system's response to environmental factors, such as dietary components and microbes, can also provide insights into how these factors shape the mucosal immune system and influence host susceptibility to disease.[17]

2.2 Adaptive Immune System

The adaptive immune system consists of fewer immune cells than the innate immune system. Nonetheless, it possesses unique characteristics that differentiate it from the innate system, such as specificity and memory. The key components of the adaptive immune response are T-lymphocytes which can be further categorized into helper Th, $CD4^+$ T-cells and cytotoxic $CD8^+$ T-cells, as well as B-cells involved in the humoral immune response.[18]

Like all immune cells, T and B cells belong to the haematopoietic lineage. In the case of T-Cells, haematopoietic stem cells are released from the bone marrow and migrate to the thymus where, they differentiate into thymocytes.[19]

There, they express randomly rearranged antigen receptors, also known as T cell receptors (TCRs)[20]. TCRs are heterodimers comprised by an alpha chain and beta chain specific for a small number of potencial antigens. T and B cells, along with other immune cells, are derived from the haematopoietic lineage. In the context of T-cells, haematopoietic stem cells are released from the bone marrow and travel to the thymus where they undergo differentiation into thymocytes.[19]

During this process, these developing T-cells express antigen receptors called T cell receptors which have undergone random rearrangements. The structure of a TCR consists of two chains - an alpha chain and a beta chain - that each have specificity for a limited number of potential antigens.[19]

Thymocytes require weak signaling triggered by moderate affinity interactions between their TCRs and self peptide major histocompatibility complex complexes. Those thymocytes that express TCRs unable to bind with self-peptide MHCs undergo apoptosis, in a process referred to as positive selection. Conversely, thymocytes expressing high-affinity receptors that react. Thymocytes undergo a critical process called positive selection, where weak signaling is necessary as a result of moderate affinity interactions between their TCR and self-peptide- MHC (major histocompatibility complex) complexes.[19]

Thymocytes that possess TCRs are incapable of binding to self-peptide MHCs, are therefore eliminated through apoptosis. On the other hand, thymocytes expressing receptors that exhibit high affinity for self- antigens go through negative selection, which leads to their deletion. Negative selection serves as a protective mechanism against autoimmune reactions targeting host tissues by promoting the development of self-tolerance. Following this rigorous screening process, only the surviving thymocytes have the opportunity to differentiate into naive T- cells.[19]

Following their maturation in the thymus, T-cells are subsequently released into the circulatory system. The diverse array of unique T-cell receptor molecules expressed by these circulating T- cells collectively make up what is known as the T cell repertoire. These receptors on T-cells specifically recognize and bind to specific antigens that are presented by antigen presenting cells (APC). This binding interaction occurs within a specialized structure called the immunological synapse, where close contact between the cell membranes of both the T-cell and APC allows for optimal engagement of TCRs and other co-receptors with their respective ligands.[21]

Upon successful formation of an immune synapse, activation is triggered within the bound T-cell population, leading them to differentiate into effector cells capable of exerting immune functions. Following activation, these activated T-cells undergo clonal expansion wherein they proliferate to generate new daughter cells carrying identical copies of their original distinctive set of antigen- specific T-cells. [19, 21]

After successfully battling an infection, a significant portion of effector T-cells undergo programmed cell death known as apoptosis. However, a subset of these

effector cells transform into long-lasting memory T-cells. The remarkable characteristic of these memory T-cells is their ability to orchestrate swift immune responses upon encountering the specific pathogen they recognize. As a result, even years or decades after the initial exposure, re-infection by the same pathogen can be efficiently cleared due to this acquired immunological memory. This represents a notable advantage offered by adaptive immunity compared to innate immunity since it enables the production and maintenance of long-term protection in response to pathogens and tumor.[19, 21]

2.2.1 The importance of T-cell differentiation and maturation

The differentiation and maturation of T-cells play a crucial role in effectively fighting tumors and pathogens. During the process of T-cell differentiation, precursor cells in the thymus undergo a series of developmental stages and acquire specific cell surface markers and functional properties that enable them to recognize and respond to antigens.

Subsequently, mature T-cells migrate to peripheral tissues where they encounter antigens presented by antigen-presenting cells such as dendritic cells. Upon activation, differentiated T cells undergo further maturation, which involves the acquisition of effector functions and the ability to carry out specific immune response.

The differentiation is ensured by four major subsets in Human T-cells, according to two cell surface markers, the common lymphocyte antigen CD45RA and the chemokine receptor CCR7.[22] CCR7 molecule is a homing receptor for secondary lymphoid organs, for T-cells homing via high endothelial venules (HEV).[23, 24] Therefore, precursor and central memory T-cells (TCM) are $CCR7^{+}$ and are characterized by repeated circulating into lymph nodes to encounter antigen presented by APC. On the other hand, effector-memory and effector T-cells down-regulate the CCR7 which allows them to migrate into peripheral nonlymphoid tissue. The CD45RA marker is expressed by precursor T-cells and again in terminally differentiated T-cells.[22, 25] Accounting for senescence marker the CD57 on the surface of T-cells appears to be related to T-cell senescence and reduced proliferation. [26] As for apoptosis marker within Human T-cells, CD95, also known as APO-1 or Fas is an apoptosis-inducing protein expressed on the surface of effector lymphocytes as well as T-cells.[27]

Also important are the inhibitory molecules such as programmed cell death 1

(PD-1), cytotoxic T lymphocyte-associated antigen 4 (CTLA-4), T-cell immunoglobulin mucin domain 3 (TIM-3) and lymphocyte activation gene 3 (LAG-3) are expressed on the surface of T-cells, to control an excessive immune response and avoid over tissue damage. Expression of these inhibitory receptors, collectively known as immune checkpoints, is elevated in T-cells from patients with different conditions to reduce a productive immune response. Blockade of immune checkpoints using monoclonal antibodies (mAbs), engineered to specifically target inhibit their immune-supressive activity has formed the basis of modern-day clinical immunotherapy, largely with success of anti-PD-1, and anti-CTLA-4 and anti-PD-L1 (PD-1 ligand) mAb.

A summary of the functional and proliferative capacity of T-cells is provided in figure 2.2, where it is plain to see that as long as T-cells undergo a differentiation process they tend to lose stemness state and memory which also affects the proliferation capacity.

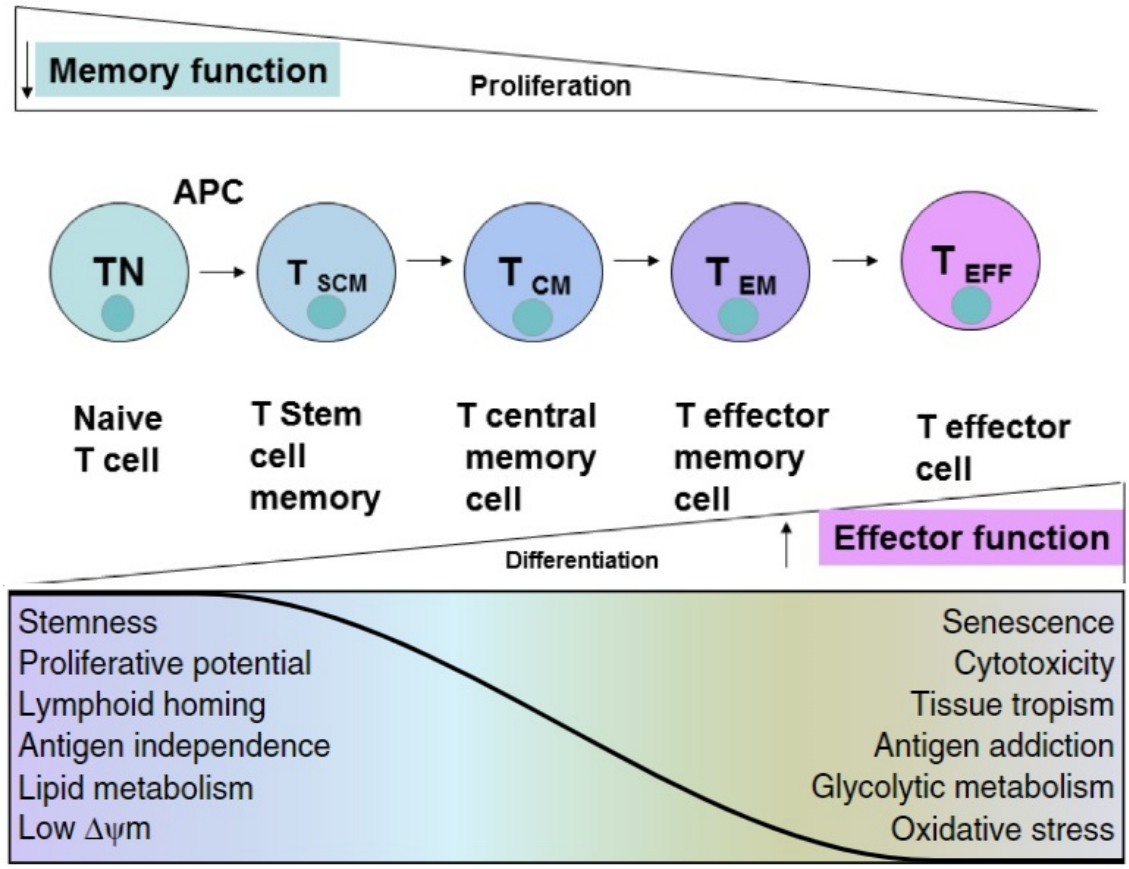


Figure 2.2: T-cell differentiation and maturation [25, 28]

2.3 Cancer and the Immune System

It took more than a hundred years of discussion to firmly establish that the immune system is capable of monitoring, detecting, responding to, and sometimes eliminate cancer.

This understanding evolved from William Coley's controversial experiments in the early days, rare accounts of spontaneous tumor regressions,[29] observations of increased cancer rates in immune-suppressed patients with acquired immunodeficiency syndrome and transplant recipients,[30] to positive associations between high levels of Tumor-Infiltrating Lymphocytes (TIL)s, and patient outcomes.[31, 32]

The suppressive role of the immune system against tumors is now widely acknowledged and utilized in various innovative therapeutic approaches, gradually establishing cancer immunotherapy as the fourth pillar of cancer treatment.[33]

2.3.1 Mechanisms of Tumor Development

Despite their diversity as a group of diseases, cancers share certain characteristics, that drive and enable their malignant transformation. In their original publication, Hannahan and Weinberg identified the original six hallmarks of cancer [34], which have since become widely known to provide a conceptual framework for understanding cancer development, as described in Subfigures **??**.

These are:

1. sustaining proliferative signaling,
2. evading growth suppressors
3. resisting cell death
4. enabling replicative potential
5. inducing angiogenesis
6. activating invasion and metastasis

No one single hallmark is sufficient to cause cancer itself and malignant transformation occurs as a multistep, evolutionary process involving sequential acquisition of most or all of the hallmarks.

A decade later, new insights from the broad and multidisciplinary scene of cancer research prompted a revisit of the original hallmarks. As a result, two emerging hallmarks and two enabling characteristics were added:

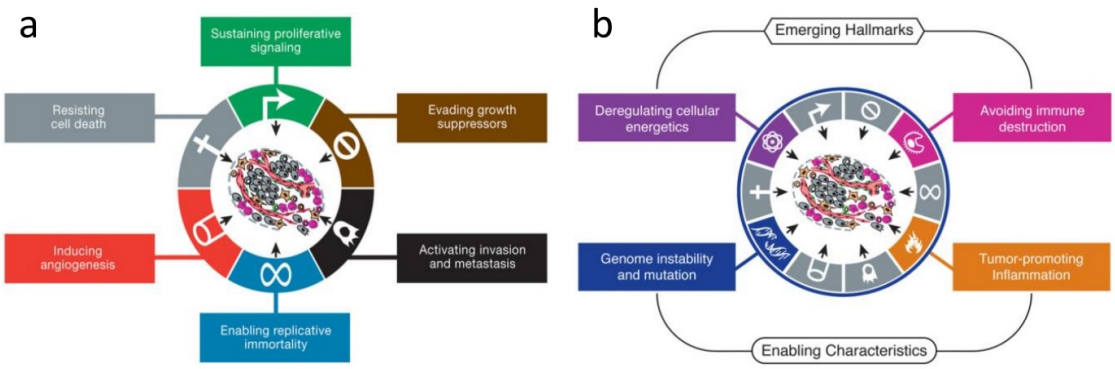


Figure 2.3: Hallmarks of cancer in (a) and Emerging hallmarks in (b) [25, 34]

- deregulation of cellular energetics
- genomic instability
- avoiding immune destruction
- tumor-promoting inflammation

As depicted in figure 2.3[35, 36]. The latter three are particularly relevant to the scope of the thesis presented here. Genomic instability is a key feature of premalignant and malignant cells, essentially enabling all the other features.

tumor cells accumulate mutations, some of which confer a selective advantage over normal cells, thereby facilitating the emergence of specific malignant clones.

Gaining function mutations in tumor genes and loss of function mutations in tumor suppressor genes contribute to the process of malignant transformation. At the same time, the increase in mutations, ironically, represents a cancer Achilles heel, making the tumor vulnerable to immune cells' detection. In the context of Human Leukocyte Antigen (HLA) class I molecules, inadvertent mutations in the protein genome result in the processing and presentation of new altered peptides, known as neopeptides, on the cell surface. Consequently, tumor cells are classified as malignant and may be detected and destroyed by the immune system.

Although still in an early stage of recognition in 2011, the inclusion of the prevention of immune destruction, as an emerging mark, was a major step towards the recognition of the immune system in the control and sculpture of cancers. Since then, the great clinical successes of various forms of cancer immunotherapy have eliminated all remaining scepticism about the importance of the immune system in controlling cancer and its potential in cancer treatment. Although the

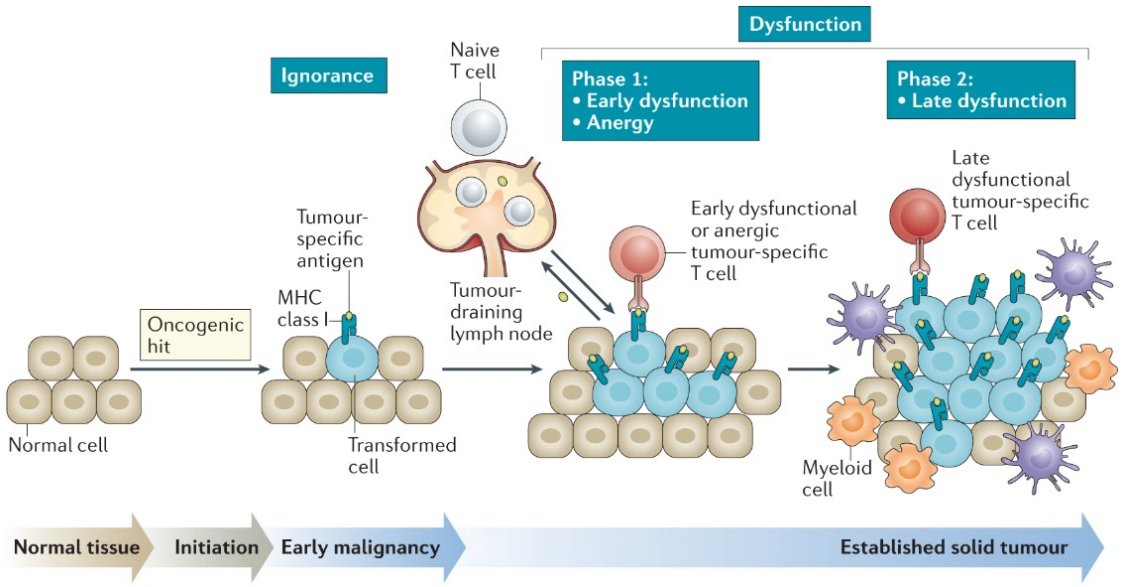


Figure 2.4: Cancer-specific differentiation of $CD8^+$ T-cells: a two-phase differentiation program. [37]

original features essentially represent the failure of the cell's intrinsic tumor control mechanism, the immune system is a cell's external cancer regulator. At the same time, the immune system's dichotomy role in the interaction with cancer is strongly highlighted by the possibility of tumor-induced inflammation as depicted in figure 2.4. This link was conceptualized from the observation that tumor formation often coincides with chronic inflammation sites.[38]

Thus, the presence of intratumoral immune cells can paradoxically also promote tumor formation and progression by releasing inflammatory molecules. These are, for example, growth, survival, proangiogenic factors, enzymes that modify the extracellular matrix, and inductive signals that lead to the transition of the endothelial mesenchymal region.[36]

However, tumor-promoting and tumor-antagonizing cells represent different subgroups of cells, so the role of the immune system in cancer is not surprising at all, but it is a consequence of the complex and heterogeneous system of the immune system in health and disease.

2.3.2 Cancer Immunoediting

The "Cancer Immunosurveillance" hypothesis was proposed already in the 1950's by Burnet [39] and by Foley [40], but was largely neglected, when experiments failed to demonstrate differences in tumor susceptibility between immunocompetent and

immunodeficient groups of mice, with the mouse models available at the time.

In 2001, improved mouse models assisted the discovery that not only tumor quantity, but also quality was affected by the immune system.[41]

When injected into recipient mice, tumors grown in immunodeficient donor mice were more immunogenic than tumors grown in immunocompetent donor mice.

These results provided the first hints that the immune system not only controls, but also shapes tumors through an editing process, and paved the way for a second generation version of the original ”Cancer Immunosurveillance” theory, namely the ”Cancer Immunoediting” theory shown in figure 2.5.[42]

The theory describes the interactions of a growing tumor with the immune system as three phases:

- elimination
- equilibrium
- escape

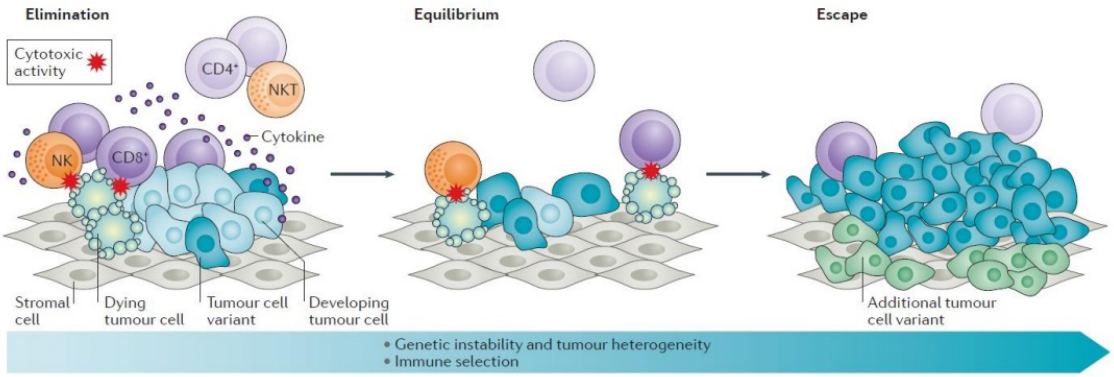


Figure 2.5: The cancer Immunoediting theory [43]

In the elimination phase, the effector cells and molecules of the innate and adaptive immune defence detect and destroy neoplastic cells before becoming clinically obvious tumors. If the mechanisms that operate during the elimination phase are insufficient, rare tumor cell variants can survive and enter the equilibrium phase. In the phase of balance, there is a balance, where the adaptive immune system controls tumor growth on the one hand, but on the other, forms the tumor immunegenicity through a constant selective pressure. Finally, this selection can result in the expansion of tumor cells that are no longer recognized or destroyed by the immune system.

At this point the final escape phase is reached; a clinically visible tumor displaying largely uncontrolled growth. Not all tumors progress through all phases. Some tumors are eradicated early during elimination, others remain at equilibrium throughout the life of the patient and yet other, more aggressive tumor types enter directly into equilibrium or escape phases. The three phases should be considered non-static and interconnected entities, reflecting the Darwinian evolutionary process occurring between the subclones of tumor cells.

The directional progress from one phase to another can be impacted by external factors, such as environmental stress and deterioration of the immune system with aging, but also through therapeutic intervention. The scope of immune therapy is obviously to reverse tumor growth and push in the direction of elimination, and indeed this has been demonstrated feasible in patients obtaining partial or complete responses from checkpoint inhibitor therapy or adoptive cell transfer.[44–46]

However, immune therapeutic treatment can also lead to effective destruction of subclones of tumor cells, creating a selective advantage and outgrowth of other subclones that escape destruction through immune escape mechanisms. Examples of these mechanisms are antigen and HLA class I and II downregulation or loss, defects in the antigen processing machinery, secretion of immunosuppressive effector molecules and recruitment of immunosuppressive cells [47–52]. In this way the immune system not only surveils, but also edits tumors.

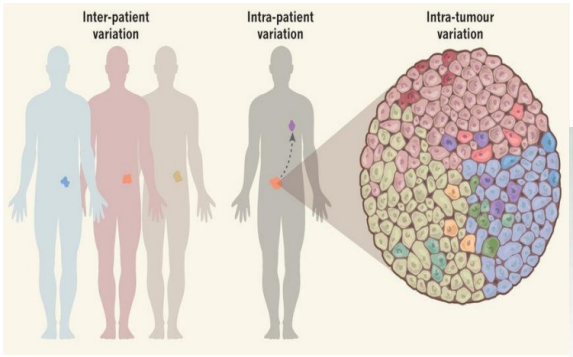
The different effects of immune therapy on immunoediting can even be observed within the same patient, where distinct tumors can shrink, grow or remain unaffected by therapy.[53] Such mixed responses likely reflect the influence of tumor heterogeneity as well as the local tumor microenvironments.

2.3.3 Tumor Heterogeneity

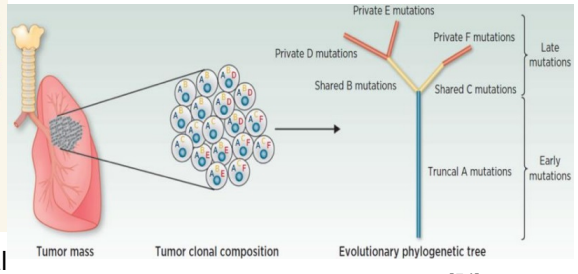
Tumor heterogeneity is a key challenge in the treatment of cancer, immunotherapy being no exemption. Heterogeneity exist at all levels:

- inter-patient
- intra-patient
- intra-tumor

This highlights the fact, that no two cancers, or even two tumors or two tumor lesions, are the same.[54]



(a) Tumor heterogeneity occurs at several levels[55]



(b) Clonal analyses of tumors[54]

Figure 2.6: Tumor Heterogeneity

Mutations can be either clonal (present in all tumor cells) or subclonal (present in subsets of cells).

The clonality of mutations within tumors can be estimated with bioinformatics tools, and analyses across multiple tumor biopsies can hence reveal the spatial composition and heterogeneity of a tumor's mutations.

After such mapping of clonal and subclonal mutations, phylogenetic trees can be constructed, where the trunk represents clonal mutations occurring early in tumorigenesis in common ancestral cancer cells and the branches represent subclonal mutations occurring in descendants later in tumorigenesis.[55]

A logical implication of the immense heterogeneity of tumors, which can be seen in the figure 2.6, is that therapies should be aimed at targeting multiple tumor neoantigens at once (term that will be introduced below), and hence it is a major focus of current strategies to select and combine the most relevant combination of neoepitopes to target individually in each patient. This opens the pathway towards personalized therapy, which will depend on crossing of multiple data.

2.4 T-cells in Cancer

T-cells are the main effector cells in the recognition and killing of cancer cells by the immune system. Two distinct types are defined by their co-receptors cluster of differentiation Cluster of Differentiation (CD)4 and CD8. The main function of CD4 cells, or T helper cells, is to orchestrate and regulate B cell and CD8 T-cell functions, whereas CD8 T cells act as cytotoxic T cells, with the capacity to directly kill infected or malignant cells [56]. Although CD4 T cells are also believed to play an important role in sculpting the immune recognition of tumors. [57–59] Furthermore

for non alpha beta T-cells like NK and $\gamma\ \delta$ T-cells may be future important players in tumor recognition and in sculpting the tumor microenvironment.[60, 61]

2.5 Adoptive Cell Therapy

Adoptive T cell transfer Adoptive Cell Therapy (ACT) is one of the most promising immunotherapy approaches in cancer treatment. There are currently four well-developed ACT techniques, including autologous tumor-infiltrating lymphocyte TIL therapy T-cell receptor engineered T-cell therapy (TCR-T), and chimeric antigen receptor T-cell therapy CAR-T).[46]

While TIL relies on the isolation and *in vitro* expansion of T cells derived from tumor or peripheral blood, respectively, TCR-T and CAR-T therapies use genetic modification of T lymphocytes to endow them with tumor antigen specificity.[62]

As an example, the patient's T lymphocytes are purified and stimulated with APC pulsed with immunogenic epitopes derived from antigens and tumors expressed by the tumor. These cells are stimulated in the presence of stimulatory cytokines, such as:

- Interleukin-2 (IL-2)
- Interleukin-7 (IL-7)
- Interleukin-15 (IL-15)
- Interleukin-21 (IL-21)

These lymphocytes expand significantly and are then injected into the patient.

2.5.1 Tumor Infiltrating Lymphocytes - TIL

For a T-cell to carry out its tumor-killing function, it must first travel to the solid tumor and enter it. Some types of cancer are characterised by highly immunogenic (or hot) tumors with large T-cell infiltrates, while others are characterised by non-immunogenic (or cold) tumors with few or no infiltrating T-cells.

Tumor infiltrating lymphocytes TILs have been shown to infiltrate solid tumors associated with a survival benefit in patients with cancer.[63–66] These data suggested that TILs are effective at delaying tumor progression, and arguably the most advanced and fully tested adoptive cellular strategy for the 'personalized' treatment of cancer.

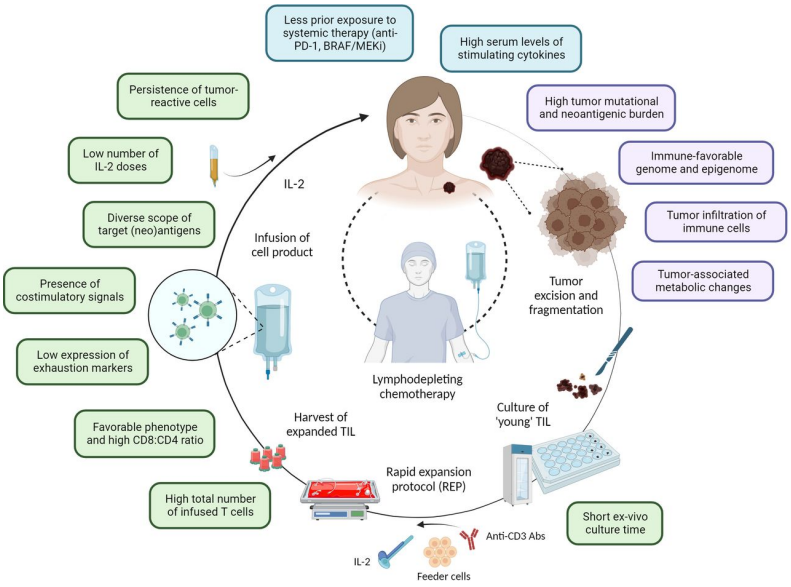


Figure 2.7: TIL manufacturing scheme [72]

TIL therapy was the first immune cell therapy shown to be practical to grow and to possess robust antitumor activity in a 1988 paper[67] in which TIL mediated regression of established poorly immunogenic tumors in murine tumor models along with high-dose IL-2.[68–70] This work was then followed up at the National Cancer Institute (NCI) by the establishment of practical techniques for expanding TIL from patients with metastatic melanoma by using IL-2, followed by a large-scale ”rapid expansion protocol” (REP) by using an anti-CD3 antibody and IL-2.[71] In figure 2.7 it is depicted the overall manufacturing strategy for TILs.

TIL were employed for adoptive transfer first in small pilot studies for patients with melanoma and shown to induce regression in 11 of 20 patients, a 55% response rate, mostly partial regressions, in patients that were extensively pretreated; in some cases, the responses were long lived. Median survivals were 11 to 12 months, with modest evidence of long-term survival. However, the long lead time before adoptive transfer required for patients to allow cells to propagate, resulted in a highly selected patient population that also received high-dose IL-2. There was evidence that the adoptively transferred TIL could traffic to tumor deposits.[68, 69]

Further studies showed that effective anti-(melanoma) tumor responses are mainly cytolytic, predominantly CD8 T cells that could lyse autologous tumor cells and produce IFNγ, suggesting that both immune effector mechanisms are biologically and clinically relevant.[68–70]

The clinical benefit of the adoptive transfer of TIL seemed to correlate with their recognition of the patient’s tumor and their ability to produce cytokines which led

to the compilation of requirements of TIL propagation, as stated in Jeffrey Weben, 'White paper' for the implementation of active cellular therapy of cancer.[73]

Several studies have shown associations between high numbers of intratumoral $CD8^+$ T cells or high CD8:CD4 regulatory T cell ratios and patient prognosis. In some cases, $CD8^+$ TILs have even found to be a better prognostic indicator than pathological criteria and more predictive of disease progression than oncogene expression.[32, 74–76]

Notably, some of the most immunogenic tumors, and therefore the most studied in the context of oncoimmunology and immunotherapy, are melanoma and lung cancer. Cancers, which have a high tumor mutation burden due to the carcinogenic effects of ultraviolet (UV) radiation exposure and tobacco smoking. (UV) exposure and tobacco smoking.[77]

However, the mere presence of T-cells is not sufficient for anti-tumor reactivity. The proliferative capacity, memory phenotype and the location within the tumor all influence the correlation with patient prognosis. [78, 79]

Interestingly, some tumors initially grow (pseudoprogress) following immune therapy, before they start to shrink. This phenomenon which almost led to the checkpoint inhibitor ipilimumab (anti-CTLA-4) not passing first round of clinical trials, is caused by an influx of T-cells to the tumor, again providing further evidence of the importance of TILs for patient outcome.[80]

An interesting and promising area of research is oncolytic virotherapy, which can cause specific lysis of tumor cells, followed by an influx of T-cells and thereby convert cold to hot tumors.[50]

2.5.2 Chimeric Antigen Receptor - CAR

Genetic modification of lymphocytes was first proposed over 30 years ago, and there have been many developments in the field since then that have facilitated their clinical utility. CAR-T approaches have been highly successful in treating certain haematopoietic cancers, particularly those directed against the B cell- specific surface protein CD19.[81]

CAR-T therapies use single chain variable fragment (scFv) chimeric antigen receptors, CAR, that can directly recognise tumor cell surface antigens without the requirement of Major Histocompatibility Complex (MHC) restriction, which is one of the main limitations of TCR-directed immunotherapies.[46, 62]

These CAR constructs can also incorporate the intracellular domains of several different co-stimulatory molecules, which can facilitate T cells' activation, proliferation, and effector functions. However, while CAR-T therapies are highly

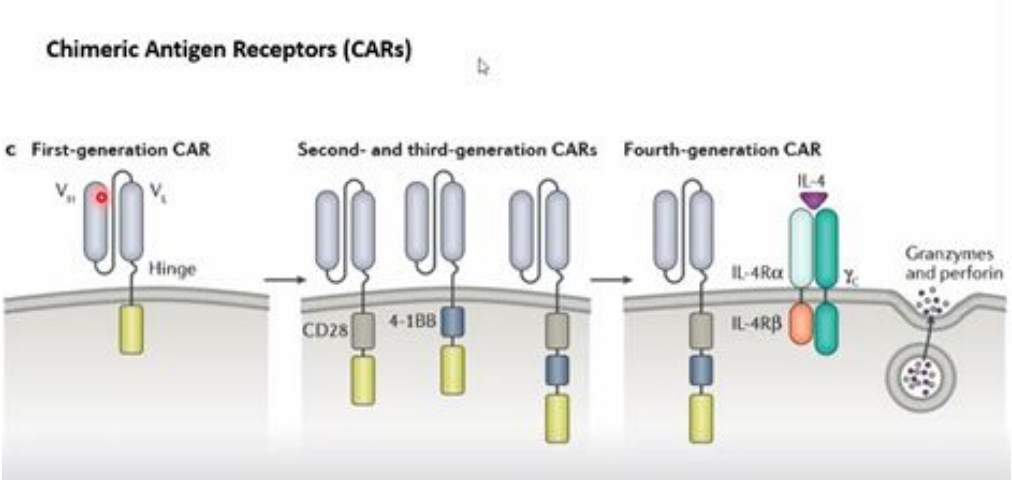


Figure 2.8: Four Generations of the structure of a chimeric antigen receptor CAR, consisting of ligand binding domains, linkers, hinge regions, transmembrane domains and intracellular signaling/activation domains. Able to recognize surface antigens other in a non-MHC manner.[81]

promising, they are limited to targeting tumor cell surface proteins that usually do not show a high degree of tumor specificity and have demonstrated little clinical efficacy in the setting of solid tumors.[82]

The different CAR constructs up to date are shown in figure 2.8.

2.5.3 T Cell Receptor T - TCR

TCR-T cell approaches can overcome targeting tumor cell surface proteins with low tumor specificity and have produced compelling clinical data in solid cancers. While CARs use an antibody-like structure for antigen recognition, TCR-T cells utilize a full TCR complex, which includes a heterodimer consisting of TCR alpha- and beta-chains, with T-cell signaling upon antigen recognition conveyed naturally by clusters of CD3 chains.[83]

Despite the success of TCR-T therapy in solid cancers, there are still challenges to overcome. These challenges include TCR product manufacturing, patient selection, and preparation with lymphodepletion.[84]

With expression of the complete TCR complex, engineered TCR-T cells can recognize polypeptide fragments expressed both within the tumor cell and at the cell surface, thus offering a much broader range of target antigens. Historically, TCR-T targets have included:

- carcinoembryonic antigen (CEA) for colorectal cancer[85]
- glycoprotein gp100 (PMEL) for melanoma[85]

- melanoma antigen recognized by T-cells 1 (MART-1) for melanoma[85]
- melanoma-associated antigen 3 (MAGE-A3) for melanoma/multiple myeloma[85]
- New York esophageal squamous cell carcinoma 1 (NY-ESO-1) for melanoma synovial cell sarcoma[85]

More recently, TCR-T cells have also targeted neoantigens generated by somatic mutations in tumor DNA, which are more tumor-specific but are also less shared by cancer patients.[86] The identification of tumor-specific antigens is one of the major challenges.

For example, in one study from 2016, a patient had 249 nonsynonymous mutations in a melanoma, of which 126 were predicted to bind to HLA- A*0201.[87]

By screening these neoantigens against the patient's own TILs, resulted in a response of 2 of the 126 putative neoantigens. However, T-Cell response was generated to a much larger proportion of these peptides in *in vitro* cultures of PBMCs from healthy donors, less then 2%.

This study points that the mutation needs to be in an exon, to be translated into a protein, it must bind to MHC molecules, it must be recognized by a TCR and it must be associated with the right immune response, capable to induce T-cell activation, stimulated cytotoxic functions and imprint migration to the tumor.[87]

These engineering approaches can successfully redirect the specificity of T-cells to selectively target tumor-associated antigens.

However, several significant limitations of these approaches remain to be overcome. One of the key goals and objectives for future research, includes the personalized identification of therapeutic tumor-specific TCRs and potential modifications to improve TCR signalling and effector function.[88]

The specificity of T cells can be reoriented by inserting genes encoding alpha-beta T Cell Receptors (TCRs) or Chimeric Antigen Receptors (CARs). CARs, developed in the late 1980s, can identify non-MHC-restricted cell surface components and can be efficiently introduced into T cells using viral vectors. TCRs, composed of one alpha and one beta chain, recognize antigens processed and presented by the patient's own MHC molecules.

CARs, on the other hand, are artificial receptors that identify antigens presented on the tumor cell surface without needing to be MHC-restricted.

A major scope now relies on the importance of selecting the right T-cell subpopulation and antigenic targets for the inserted TCRs or CARs. This assumption is based that early-stage T-cells, such as naïve or central memory cells, show

improved antitumor responses.For example, $CD8^+$ T-cells' differentiation state is inversely related to their proliferation and persistence capacity.

Overall, adoptive cell therapy approaches such as TIL therapy and CAR T cells are showing great promise in the field of cancer immunotherapy. These therapies have the potential to revolutionize cancer treatment by providing targeted and potent strategies for eradicating tumor cells.

In figure 2.9 shown on the next page, it summarizes all the genetic modified T-cell therapies applied in clinical context but focused on using T-cells in stem cell memory differentiation state since it has been seen in sveral previous studies to enhance a better prognosis, since T-cells are still in an initial proliferation state and can undergo maturation phase.

In summary it discusses two therapeutic approaches using Tscm-cell-based interventions for treating human diseases. On one side, Tscm (T memory stem cells) can be disrupted to manage diseases driven by these cells, such as autoimmunity, T-cell leukemia, and T-cell tropic infections.

These cells can be harnessed to enhance T cell-based immunotherapies against cancer and infectious diseases. This involves using naive-like/precursor T cells from patients or donors to generate and expand Tscm cells *in vitro*, with or without genetic modifications.

Such modifications may include inserting tumor or virus-specific genes for CAR or TCR, editing TCR genes, transferring suicide genes for safety controls, or editing the CCR5 gene particularly in Human Immunodeficiency Virus type 1 (HIV-1) settings.

Additionally, virus-specific Tscm cells can be expanded using existing antigen-specific TCRs through specialized textitin vitro protocols that promote Tscm cell generation, involving interactions with antigen-presenting cells APC.[25]

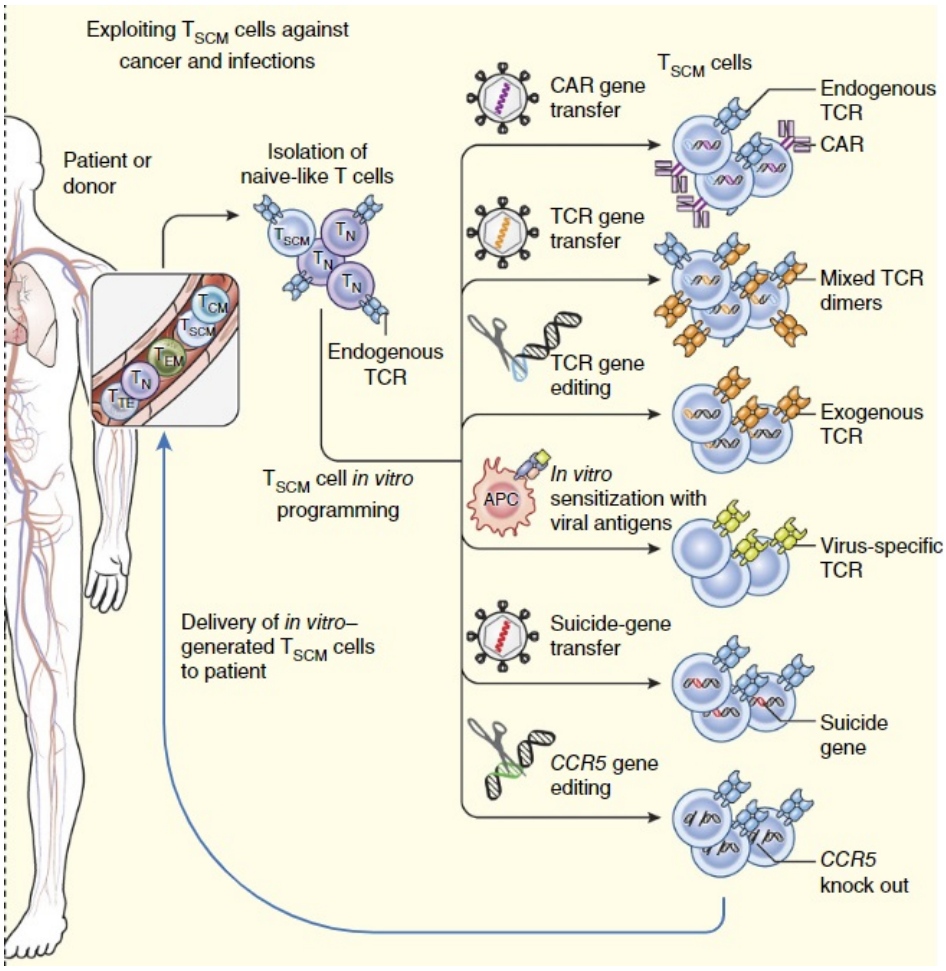


Figure 2.9: CAR and TCR based cell immune cell therapeutic stratergies for Human diseases with T-cells in Tscm phenotype.[25]

2.5.4 How do CD8$^+$ T-cells recognize cancer?

Through a peptide presented by the MHC, the cytotoxic or CD8$^+$ T-cell recognizes the cancer cell, as depicted in figure 2.10 in here, this recognition process is divided into three molecules:

- TCR - T-cell receptor
- Peptide
- Major Histocompatibility Complex I (MHC-I), major histocompatibility molecule complex class I or human leukocyte antigen HLA

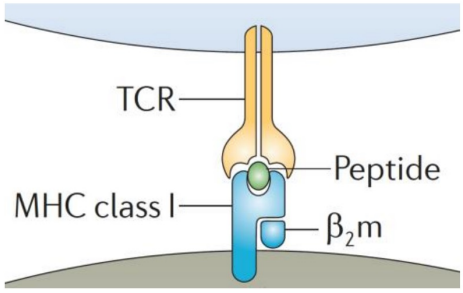
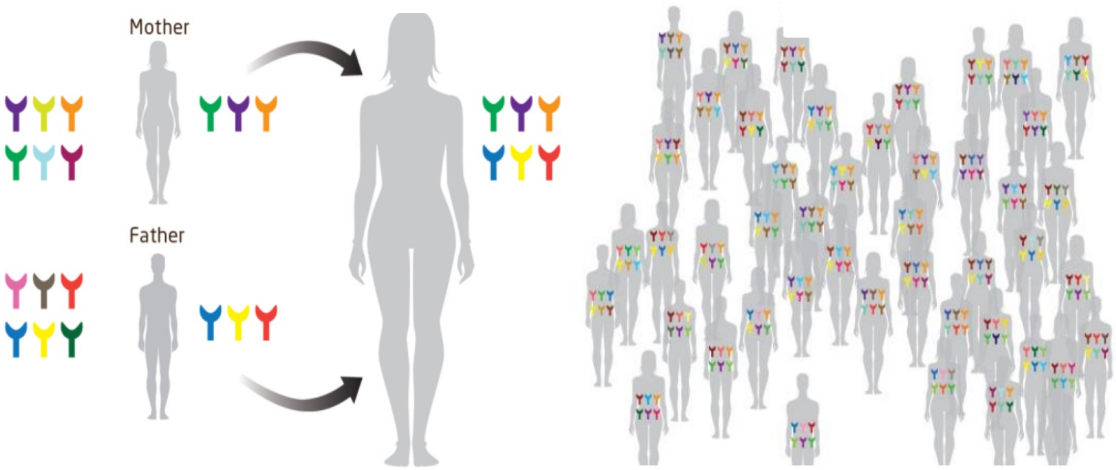


Figure 2.10: The MHC-peptide-TCR interaction[89]

However, the three-part molecular interaction is complex and diverse. Firstly, the HLA locus is both polygenic (constituting HLA-A, -B and -C) and highly polymorphic, with more than 10,000 alleles identified to date.[90]

Each individual can therefore present up to 6, (3 maternally + 3 paternally derived) different HLA molecules at a time and homozygosity at the HLA locus is rare, as shown in figure 2.11 on the next page. Every HLA molecule possesses different peptide binding preferences, with the result that they present different peptide repertoires to the TCR. The combined polygenism and polymorphisms of HLA molecules probably evolved as a protective mechanism for the entire population. Even if a pathogen evolves to avoid the immune detection of the peptide-HLA (pHLA) complexes present in a single individual, chances are that the population as a whole is likely to be protected.[91]

Although most likely the result of an evolutionary arms race with pathogens, HLA polygenism and polymorphism also positively affect the chances that a malignantly transformed cell will be detected by an encountering T-cell. It is therefore not surprising, that HLA loss at one or more loci is a known immune escape mechanism, used by tumors to hide from T- cells.[92]



(a) Maternally and Paternally inherited HLA molecules

(b) Human population as a whole with more than 10.000 different HLA alleles

Figure 2.11: HLA in Population, generated in BioRender

Secondly, proteins are built from 20 naturally occurring amino acids. Given this, the amount of theoretical peptides that can bind to HLA class I molecules (8-14 amino acid length restriction) arises to more than 1018.[93]

This number is however affected in both directions. It may be underestimated, as post-translational modifications, affecting T-cell recognition of the original peptide sequence, increase the number of possible HLA ligands.[**coulie2014tumor**, 94]

Besides that, the human proteome as the source of peptides, as would be the case for cancer, the number of possible peptides is substantially lower. Moreover, far from all endogenous peptides end up as HLA bound, presented peptides at the surface.

The antigen processing and presentation pathway involves proteasomal cleavage, translocation to the endoplasmic reticulum endoplasmic reticulum (ER) via transporter associated with antigen processing (TAP), loading onto HLA molecules and transportation through the Golgi apparatus to the cell surface, like in figure 2.12 shown in the next page.

All of these steps confer peptide specificity, and thus it has been estimated that only 1% of peptide fragments from the human proteome are presented at the surface in context with HLA class I molecules.[95]

Some HLA's are more promiscuous than others, and HLA-C molecules are known to be more selective in peptide binding, resulting in a ten-fold lower pHLA expression at the cell surface.[96]

In cancer, T cells need to be able to recognize and react to the 1% peptide fragments from malignantly transformed proteins, with binding capacity for the

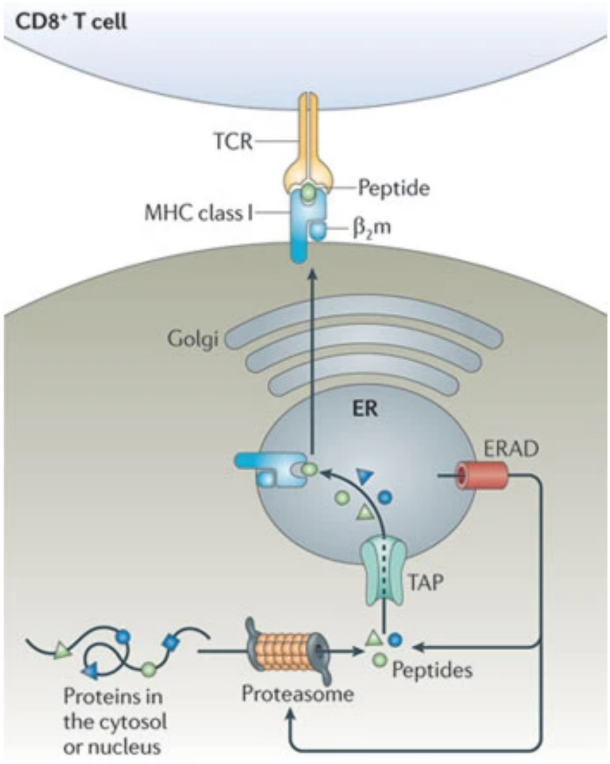


Figure 2.12: MHC class I antigen manner to process and present antigen [72]

HLA repertoire of the patient. Thus follows the third source of diversity, the TCR repertoire.

The diversity of the T cell receptor arises from somatic recombination during T cell maturation, where variable (V), joining (J), and in some cases, diversity (D) gene segments of the complementarity determining regions (CDRs) of the TCR locus are (nearly) randomly rearranged.

It was previously believed, that each clonal T cell could respond to only one target. However, it is estimated, that somatic recombination results in 107 TCRs per individual,[97] whereby only a fraction of the possible pHLA complexes would be detected, if one T-cell detected only one pHLA complex. Therefore, it is established, that TCRs are in fact highly cross-reactive, recognizing an estimated 106 different pHLA complexes.[98]

2.5.5 Central and Peripheral Tolerance Mechanisms applied by the Immune System

In addition to being able to recognize and respond to any possible pathogenic or malignant cell invasion, the second most difficult task of the immune system is to do so without being reactive to harmless foreign substances or body cells. This

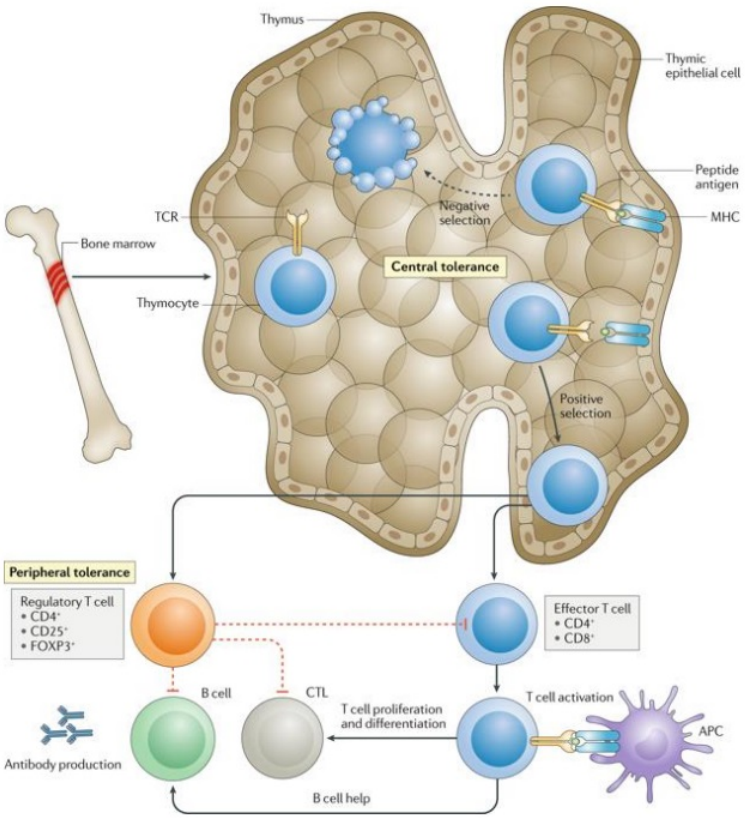


Figure 2.13: Central and Peripheral tolerance mechanism [99]

ability to distinguish between the self and the harmless self and the harmful self is actually the cornerstone of the success of the immune system.

And although allergies, autoimmune diseases, uncontrolled infectious diseases and cancer, demonstrate the accidental failure of the immune system to carry out its work, these are rare examples where things have gone wrong. The ability to distinguish between self and non-self is the result of the central and peripheral mechanisms of tolerance.

During maturation in the thymus, T-cells are presented with pHLA complexes as depicted here in figure 2.13. Positive selection ensures that only T-cells that can interact with host HLAs are kept, whereas those that interact too tightly with self-pHLA complexes are deleted through negative selection.

The process of central tolerance is believed to induce apoptosis in as much as 95% of all developing T-cells, a tremendous trade-off reflecting the importance of this control mechanism.[100]

Within the remaining 5% of T-cells, some autoreactive cells will succeed to escape the thymus. These are however kept in check by peripheral tolerance mechanisms, for example the lack of co-stimulatory signals, suppression by CD4

regulatory T cells and myeloid derived suppressor cells.[101]

The population of $CD4^+$ T-cells called regulatory T cells (Tregs) has been shown to suppress T cell–mediated host immune responses against self- and non-self- antigens. Tregs were first described by Sakaguchi and coworkers as a circulating subset of murine $CD4^+$ T cells expressing high levels of CD25 (the interleukin [IL]-2 receptor α chain), which upon adoptive transfers could prevent development of autoimmune disease.[102]

The number of administered doses of IL-2 was found to be positively associated with peripheral Treg reconstitution. These observations provide strong evidence that endogenous $CD4^+$ Tregs have a negative impact on cancer therapy, and suggest that strategies reducing Treg levels may provide clinical benefit to cancer patients; immune-suppressive drugs, particularly cyclophosphamide, reduces Tregs – a regimen that has been used in several clinical trials for 'conditioning' of patients receiving cellular therapy (clinical trials are registered at www.clinicaltrials.gov as NCT00001832, NCT00096382, NCT00335127, NCT00509496, and NCT00513604).[103]

Many of these natural peripheral tolerance mechanisms are 'hijacked' by the tumor, preventing effective elimination of tumor cells despite T-cell recognition.

2.5.6 Detection of Antigen Specific T-cells

T-cell detection methods can be divided into two groups, depending on the results of the analysis. Classic functional assays, such as enzyme-linked immunospot (ELISPOT) or ELISA, limiting dilution assay and intracellular cytokine staining (Intracellular Cytokine Staining (ICS)), can identify antigen-responsive T-cells, but do not provide information about the pMHC (peptide MHC) specificity is only obtained for functional T-cells and often with lower sensitivity. As an alternative, a number of methods have been developed that specifically MHC-restricted recognition of T-cells, but without any measure of functionality.

The detection of T-cell antigen specificity is complicated by the high diversity and complexity of the pMHC-TCR interaction, as described above.

A further challenge, particularly when interrogating the landscape of T-cells specific for a common cancer antigen, is the often low frequency of T-cell clones and their low TCR affinity for the binding pMHC complex. To address these issues, comprehensive analytical tools have been developed over the last two decades.[89]

In the context of these thesis, ELISA, ICS and phenotype assessment will be applied to build up a structured pipeline to evaluate T-cell functionality with ELISA

and ICS, and phenotype to classify immune cell type.

2.5.7 T-Cell Antigens

The first human melanoma T-cell antigen, melanoma-associated antigen (MAGE)-AI, was described nearly 30 years ago,[104] and hundreds of T-cell antigens in different cancer types have been described and presented in ”Cancer Antigenic Peptide Database”.[105] T-cell antigens in cancer can be grouped into five classes derived from the seven classes of Human Tumor Antigens in table D.1 in appendix D.

- **Overexpression** antigens are proteins, which are expressed in a range of normal tissues, but whose expression is significantly increased in cancerous cells. This leaves a window for tumor specific recognition, where the expression level on normal cells might not be sufficient to elicit a T cell response, whereas the increased expression in tumor cells may be.

- **Lineage-restricted differentiation** antigens are restricted to defined differentiated cell types and tumors that arise from these cells. Despite the expression in healthy tissues, many antigens from this group have been used in immune therapeutic trials, although sometimes causing on-target off-tumor autoimmunity side effects. [106]

- **Germline** antigens are a group of proteins, whose normal expression is restricted to testes, placenta and fetal development. Re-expression in cancer cells results from re-activation of genes that are normally silenced. Lack of HLA expression in testes and placenta confers an immune privileged nature on these tissues, meaning that in theory no autoimmune reactions are risked with therapeutic use of germline antigens.

- **Virally Induced** cancers, in this cancers, viral antigens are good candidates for immune therapeutic treatment due to their non-sef origin, hence independent from tolerance and autoimmunity [107]

- **Mutated molecules or translocation products**, which originate from somatic mutations in tumor tissue. Mutation antigens can arise from frequent shared mutations between patients, however oftenly they are private mutations, patient unique, or from even tumor lesions within a patient.

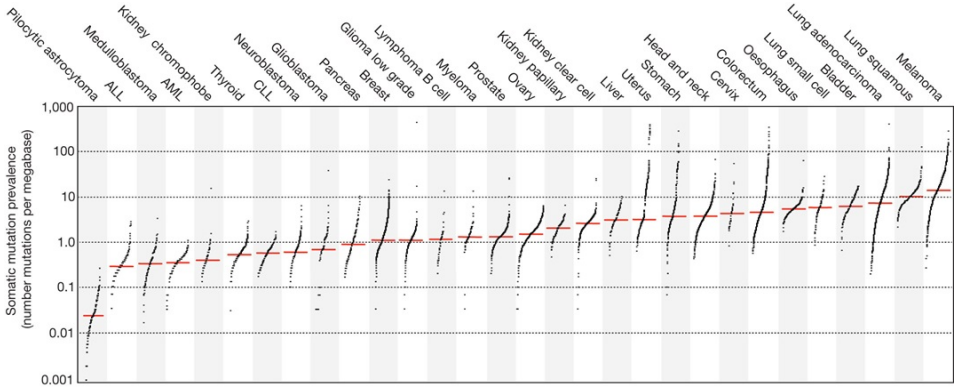


Figure 2.14: Somatic Mutations in Different Tumors [77]

In addition, T-cell antigens of relevance in cancer may also include cryptic epitopes, encoded by alternative open reading frames or spliced peptides and post-translationally modified peptides [**coulie2014tumor**].

The usefulness of an antigen in clinical implementation is defined by three parameters:

- tumor restriction
- antigen frequency
- antigen stability under immune selective pressure

An ideal target antigen would be expressed in all cancer patients, with sole expression in malignant cells only and with stable expression under immune selective pressure. However, only rare examples of antigens exist, like human papilloma virus (HPV) 16 E6/E7) that fulfill all requirements.[108]

The tumor restricted expression is important for safety concerns. Mutation and viral antigens are attractive targets in cancer, as they are uniquely expressed in the tumor tissue, which decreases the risk of on-target off-tumor autoimmune side effects in therapy. However, only around 9% of cancers are virally induced, and although shared somatic mutations, such as cyclin-dependent kinase 4 (CDK4) and B-RAF, appear, they are rare exceptions. The vast majority of somatic mutations is patient specific and therefore need to be identified and targeted individually in each patient. Moreover, the mutational load varies substantially between cancer types, with melanoma and lung cancers [77], as depicted in figure 2.14.

In cancers with fewer mutations, therefore, overexpression, differentiation and germline antigens could, in some cases, represent more realistic targets. In addition, the antigens in these classes are shared between patients and sometimes between cancer types, which means that therapeutic objectives are relevant in a

larger patient group and therefore turned into more cost-effective out-of-the-box therapies.

2.5.8 Neoepitopes/Neoantigens

The majority of mutation antigens, also known as neoantigens, are unique to individual patients, and technology did not support personalized screening strategies at the time, since shared non-mutated antigens were the focus of the 1990s/2000s within the scope of targeted immune therapeutic strategies.

However, with advances in next generation sequencing and lower sequencing costs, the neoantigens have gained renewed interest. The terms neopeptide and neoepitope are used to distinguish between neoantigen derived peptides predicted to bind to HLA's and the fraction of these that are recognized by T-cells.

An increasing number of studies point to an important role of neoantigens in cancer immunology and have gain interest in their clinical utility. It has been observed that the number of predicted neoantigens correlates with cytolytic activity in tumors, that the mutational burden correlates with TIL infiltration and with survival across a range of cancer types, and additionally, clinical responses from checkpoint inhibitor and adoptive cell therapies ACT have been associated with neoepitope specific T-cells [109]

2.5.8.1 Origin

Neopeptides can arise in two ways as shown in figure 2.15:[110], on the next page via two know mutation ways:

mutations in HLA binding (anchor) residues where an altered repertoire of peptides is presented;

mutations of residues in contact with the T-cell receptor leading to an altered recognition by the T-cell.

Whereas neopeptides in the first group can be identified by a shift in HLA binding affinity, the second group would not distinguish themselves as different from the wild-type peptide counterparts, when analyzed with a HLA binding prediction algorithm.

The two different categories of neopeptides are also expected to be differently impacted by tolerance mechanisms. Tolerization only occurs for presented peptides. Thus, T-cells recognizing neopeptides arising from improved binding to the HLA is expected to have surpassed central tolerance and for this group of neopeptides the self-similarity seems to only marginally affect their immunogenicity.

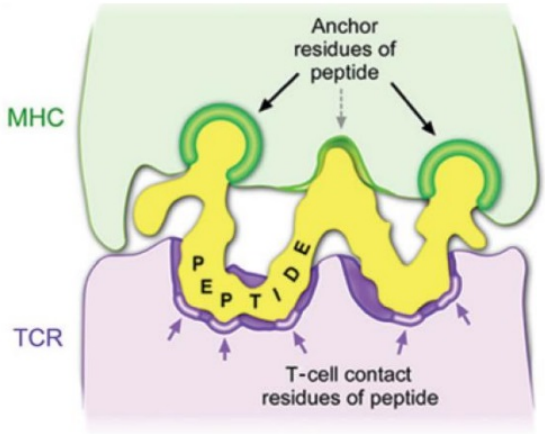


Figure 2.15: Mutations that give rise to neoepitopes [110]

In contrast, neopeptides arising from alterations in the TCR recognition area are conserved binders, capable of presentation in their wild-type form in non-cancerous tissue.[109, 110]

2.5.8.2 Mutation types

The most commons mutations are single nucleotide variations (SNVs) which has been tested in the scope of this thesis, but other important source of neoepitopes can arise from in-frame and frameshift insertions and deletions (indels). Frameshift derived neopeptides, which in nature are much different from the original wild-type peptides, can be speculated to represent a particularly immunogenic group. [111] All this mutations are shown in figure 2.16, on the next page on how a SNV can be generated.

2.5.8.3 Neoepitope Prediction

Immunogenic neoepitopes arise in only a fraction of mutations. A critical challenge in the development of neoepitope-based treatment strategies is the efficient and accurate prediction of putative neoepitopes. In figure 2.17 a scheme depicts the current strategy applied for neoepitope prediction, shown in the end of this subsection. Where DNA is extracted from a patients tumor and matched healthy tissue and subjected to either whole exome sequencing (WXS) or whole genome sequencing (WGS) to identify somatic mutations mapping specifically to the tumor. Incorrect classification of germline variants are avoided. The goal of this approach is to identify genetic variants that alter protein sequences. Since these variants can be responsible for both Mendelian and common polygenic diseases.

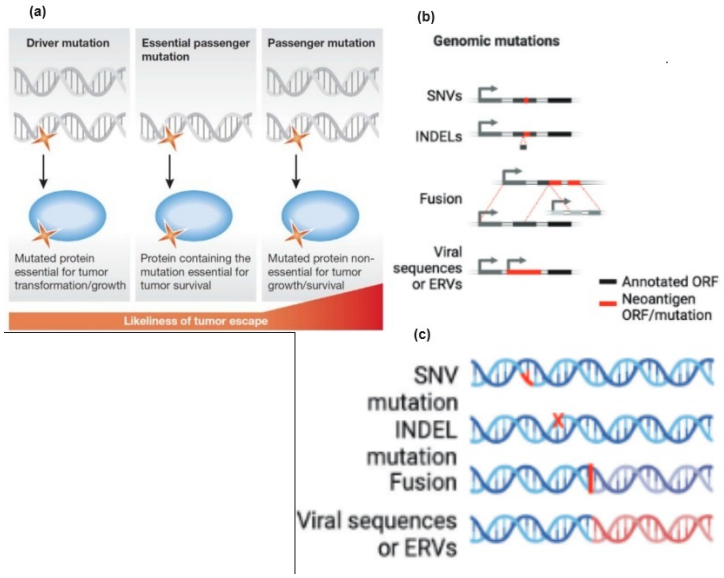


Figure 2.16: Common mutations that arise to neoantigens [111]

There are many factors that make exome sequencing superior to single gene analysis including the ability to identify mutations in genes that were not tested due to an atypical clinical presentation or the ability to identify clinical cases where mutations from different genes contribute to the different phenotypes in the same patient.

Furthermore, RNA sequencing (RNA-seq) of the tumor is applied to identify and select only expressed mutational products. It is used to analyze the continuously changing cellular transcriptome. Specifically, RNA-Seq facilitates the ability to look at alternative gene spliced transcripts, post-transcriptional modifications, gene fusion, mutations/single-nucleotide polymorphism (SNP) and changes in gene expression over time, or differences in gene expression in different groups or treatments.

Lastly, peptide sequences from the mutated, expressed proteins are run through HLA binding algorithms to prioritize putative neopeptides with potential to bind the HLA's of the patient (identified by WXS or WGS). Hereafter, the resulting library of patient-private putative neopeptides can be further experimentally tested *in vitro* T-cell detection assays.[112]

Another layer which will not be tested is that use of mass spectrometry based identification of endogenously processed and presented HLA-ligands. The combination with exome sequencing has shown increased chances of identifying immune-dominant neoantigens in mice, plus it can identify spliced peptides or post translationally modified peptide ligands, that are missed when predictions are based on the germline or mutated sequence.[113]

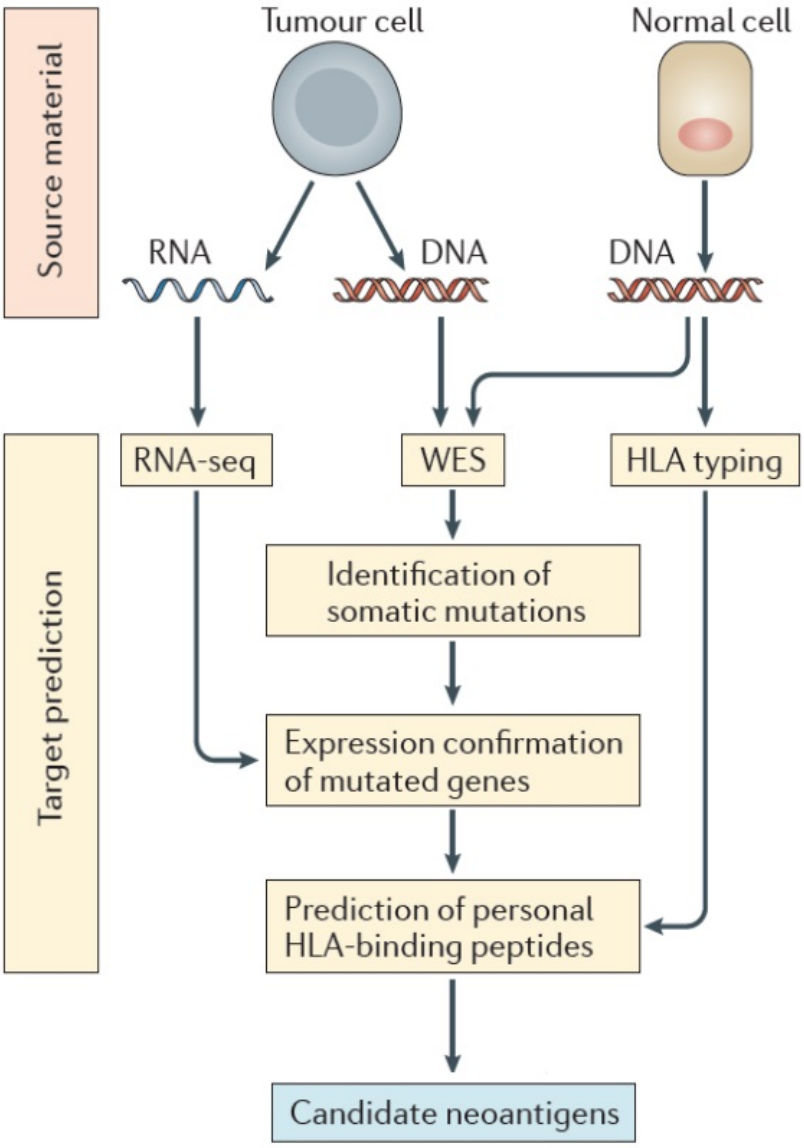


Figure 2.17: Neopeptide prediction pipeline [112]

2.5.9 Homing markers in the context of adoptive cell therapies for solid/epithelial tumors

A central feature of T-cells is their ability to enter tissues and organs, without this capacity, they may be directed against a pathogen, yet not able to enter into a target tissue and circulate in peripheral lymph nodes. This is true particularly for T-cells able to mediate tumor regression or, in contrast, promote disease progression, as in autoimmune disease.

For instance, the expression of tissue homing markers such as CCR9 and CD103 (both with affinity for gut and liver homing) on the surface of $CD8^+$ T cells provides this information, for example increased frequency of $CD8^+$ naive/precursor cells – also expressing $CCR9^+$[114] – associated with prolonged overall survival in patients with melanoma. Promising cancer immunotherapeutics depend on mobilization of cytotoxic T cells across tumor vascular barriers through mechanisms that are poorly understood.

In case of, $CD4^+$ T-cells they can be divide in several groups including helper T-cells and regulatory T-cells.

The T-helper subtype 1 is characterized by pro-inflammatory immune responses marked by production of IFNγ and Tumor Necrosis Factor-Alpha (TNFα)-α (directed against viral pathogens or tumors), whereas type 2 'tolerizing' $CD4^+$ T helper cells (Th2) produce anti- IL-4, IL-5 and IL-13 that favor antibody response, Th2-type cells are particularly important in immune responses against certain pathogens.[115, 116]

The Th17 cell subset, characterized by the production of the pro-inflammatory cytokine IL-17, can be beneficial at early stages of infection as well as in anti-cancer directed immune responses, yet over IL-17 at later disease in infections or cancer can be detrimental.[117–119]

The $CD4^+$ Th subsets are divided according not only to their cytokine and transcription factor expression profiles, but also based on their expression of certain chemokine/tissue homing markers (CXCR3, CCR4 and CCR6).[120]

$CD8^+$ memory stem T-cells (Tscm) are long-lived lymphocytes that persist in the host, without the need of constant antigen stimulation. Interestingly, these cells are multipotent and provide a potential reservoir for T-cell memory through life.[121]

Similar to CD4+ T-cells, CD8+ T-cells or TCR $\gamma\delta^+$ T-cells can also be categorized into Th1, Th2, or Th17-type immune cells.

The success of T cell-based cancer immunotherapies including adoptive T cell transfer, dendritic cell (DC) vaccines, and immune checkpoint inhibitors, hinges

on efficient trafficking of blood-borne cytotoxic $CD8^+$ T-cells across tumor vascular barriers a process guided by chemokines. And chemokine markers are used to characterize the T-cells Th1, Th2, Th17 and Treg as presented in Table C.1

Functional chemokine receptors including CXCR3, CCR2, and CCR5, are present on circulating tumor-reactive T-cells. Moreover, cognate chemokine ligands (CXCL9/10, CCL2, and CCL5) 'decorate' vessel walls.

Dipeptidylpeptidase 4 (DPP4, also known as CD26) limits lymphocyte migration to sites of inflammation and tumors. Inhibition of DPP4 enzymatic activity enhanced tumor rejection by preserving biologically active CXCL10 and increasing trafficking into the tumor by lymphocytes expressing the counter-receptor CXCR3.

These findings provide direct *in vivo* evidence for control of lymphocyte trafficking via CXCL10 cleavage and support the use of DPP4 inhibitors for stabilizing biologically active forms of chemokines as a strategy to enhance tumor immunotherapy.[122]

The CXCR3 chemokine receptor uniquely functions as the master-regulator of cytotoxic CD8(+) T-cell extravasation and tumor control despite the multiplicity of chemokines available in the tumor landscape.[123] CXCR3 engagement by CXCL9/10 triggers firm adhesion to vascular endothelium and subsequent migration into the underlying tumor parenchyma.

By contrast, CCR2 and CCR5 are dispensable for T-cell extravasation although they, together with CXCR3, could potentially regulate T-cell retention, proliferation, or survival in the intratumoral compartment.

Altogether, this CXCR3-dependent process allows $CD8^+$ T-cells to initiate apoptotic pathways leading to tumor cell destruction. This unique role for CXCR3 suggests that the efficacy of cancer immunotherapy could be enhanced by adjuvant treatments (like, chemotherapy, radiation, and dipeptidylpeptidase 4 (DPP4) inhibitors) that boost CXCL10 availability at vascular checkpoints.

The requirement for CXCR3 is not unique to adoptively transferred cells as it was also shared by the endogenous T-cell pool in a preclinical murine melanoma model: CXCR3 dominance is maintained for trafficking of adoptively transferred human $CD8^+$ effector T cells in human melanoma xenografts.[123]

These studies reveal a unique role for CXCR3 and its ligands in cancer immunotherapy that could not be predicted from genomic or proteomic profiling of the tumor microenvironment.

Also $CD103^+$ $CD8^+$ T cells, appear to have a role in adhering to epithelial surfaces in the gut.

CD103 marker, also known as integrin alpha E (ITGAE), forms a dimer with integrin beta 7 (ITGB7) to constitute a receptor which binds E-cadherin (CD324)

expressed on epithelial surfaces. E-cadherin is particularly important for intraepithelial immune cells penetration/localisation, which explains in part why TIL expressing CD103 might be useful in eliciting anti-tumor functions in the tissue compartments.

CD103 expression on T cells was strongly reflective of a survival advantage in patients with cancer (renal cell carcinoma [124], cervical cancer [125] and lung cancer [126], respectively). In another article, a higher ratio of $CD103^+$ intraepithelial lymphocytes to total $CD103^+$ lymphocytes in pancreatic ductal adenocarcinoma (PDAC) tissue based on immunohistochemical staining was indicative of better disease-free survival of the patients.[127]

Another study states that $CD103^+$ $CD8^+$ TILs are sequestered away from the tumor centre in PDAC to fibrous tissue compartments, in line with poor patient survival dynamics [128] further to yet another piece of evidence that these cells are cytolytic[129], is a shred of rather compelling evidence that $CD103^+$ $CD8^+$ T cells have a clinically relevant and possibly significant role in fighting epithelial cancers.

2.5.10 IFNγ production as a signature of tumor surveillance

IFNγ is the signature cytokine produced by Th1 cells ($CD4^+$) but is also derived from $CD8^+$ T-cells, $\gamma\delta$ T-cells, NKT-cells, and NK-cells. IFNγ exerts its biologic effects by interacting with an IFNγ receptor (IFNGR) 1 and IFNGR2 resulting in activation of Janus tyrosine kinase (JAK)- STAT signaling pathway which ultimately leads to phosphorylation of two STAT1 molecules followed by their dimerization and nuclear translocation IFNγ is a cytokine that is well recognized to play a central role in coordinating tumor immune responses.[130]

For instance, absence of IFN-γ receptors or STAT1-deficiency in preclinical studies have shown rapid tumor development and with higher frequency. [131, 132]

An unequivocal demonstration of cancer immune surveillance is that tumor suppressor function of the immune system is critically dependent on the actions of IFN-γ which along with lymphocytes collaborate to protect against development of carcinogen-induced sarcomas and spontaneous epithelial carcinomas. In tumor cells, IFN-γ upregulates the expression of the MHC class I antigen processing and presentation pathway, thereby enhancing tumor cell immunogenicity and facilitating tumor recognition and elimination.[133, 134] IL-12-induced tumor regression correlates with *in situ* activity of IFNγ produced by tumor-infiltrating cells and its secondary induction of anti-tumor pathways.[135]

IFNγ displays an essential role for tumor responsiveness by regulating the migration of T-cells into tumor tissue.[136]

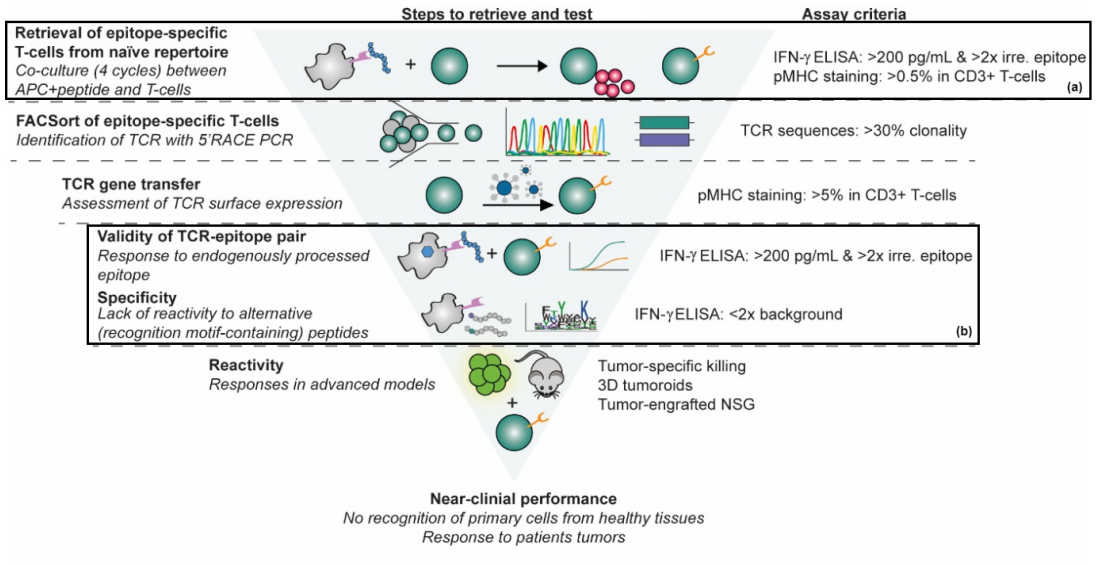


Figure 2.18: General Validation plan of TCR constructs [137]

Therefore, the production of IFNγ as a release criterium makes biological and clinically sense.

In the scope of this study, the selection of IFNγ seems to be a pivotal cytokine to select potential TCR constructs, as in here on figure 2.18 in section (a) and (b) IFNγ production obtained through ELISA serves as key parameter in the scope of TCR therapy functional validation, but it is also a key parameter within CAR and TIL therapy.

3

Materials and Methods

3.1 Introduction

The present study serves as a model of a normal study applied in the context of immuno-monitoring, taken place at the Leibniz Institute of Immunotherapy with patient samples. On the following sections it is explained in detail the *in vitro* and *in silico* experimental steps with material and methodology used.

All the materials/reagents and equiment/devices used for the *in vitro* assays are respectively in tables: table A.2 and table A.3 at appendix A.

All the algorithms used in the *in silico* assays used to extract results are in appendix B.

3.2 Population

The study population concerns five patients diagnosed with an epithelial tumor who underwent surgery and informed consent was obtained for the collection of pre-surgery blood and tumor tissue after surgery with the consent of the clinical pathology unit.

The patient data used in this study is encrypted and we did not have access to the clinical numbers of the respective patients.

The purpose of this study is to use changes in the pattern of immunocellular phenotype and type of effector response to build a future platform that will allow a close-to-personalised assessment of the pattern of immunocellular changes.

3.3 *In Vitro Assays*

The *in vitro* assays are derived from an experimental plan to obtain data on cellular immuno-markers, performed via flow cytometry through the phenotyping assessment of the immune cells. And from obtaining the concentration of IFNγ present in the co-culture supernatants of peripheral blood immune cells and tumor immune cells, TILs when in contact with synthetic peptides with the same sequence as tumor associated antigens and neoantigens (one amino-acid changed in red) as shown in table D.3 at appendix D.

The experimental plan, shown in figure 3.1, serves the purpose of studying the immune environment of tumors, focusing on the activity and characteristics of TILs, which are a critical component of the anti-tumor immune response within the tumor microenvironment. The process starts with tumor surgery, where a sample is taken and prepared for various analyses. The tumor sample undergoes whole exome sequencing (WES) to identify its genetic make-up and to synthesise peptides with private mutations for the specific patient. At the same time, portions of the tumor are cultured on day one with feeder cells and a stimulant anti-CD3-(OTK3) and IL-2 to promote the growth and activation of TILs.

At the same time, preoperative blood is collected to perform a Whole Blood Assay (WBA) with synthetic peptides, which serves as a baseline measure of the immune response. The synthetic peptide plate contains wild-type tumor-associated antigens and single nucleotide variants of tumor-associated antigens. Supernatants from this plate are then extracted on day seven.

Also on day seven, TILs are extracted and analysed by flow cytometry using T-cell/T-cell activation and T homing panels to characterise the lymphocytes and their migratory properties. In addition, these TILs are cultured with synthetic peptides for seven days to monitor their response to wild-type tumor-associated antigens and single nucleotide variants of tumor-associated antigens. After seven days, the supernatants are tested for IFNγ production via ELISA.

3.3.1 Primary culture conditions

The primary culture conditions concerns steps (a), (b) and (c) from the experimental plan in figure 3.1 on the next page.

All the TIL primary cell cultures were maintained at 37^{o}C in a humidified atmosphere containing 5% CO_2. The primary cultures of TILs were cultured under suspension conditions. TILs and pre-surgery-blood, Whole Blood Assay (WBA) co-cultured with synthetic peptides, according to the peptides list in table D.3 in

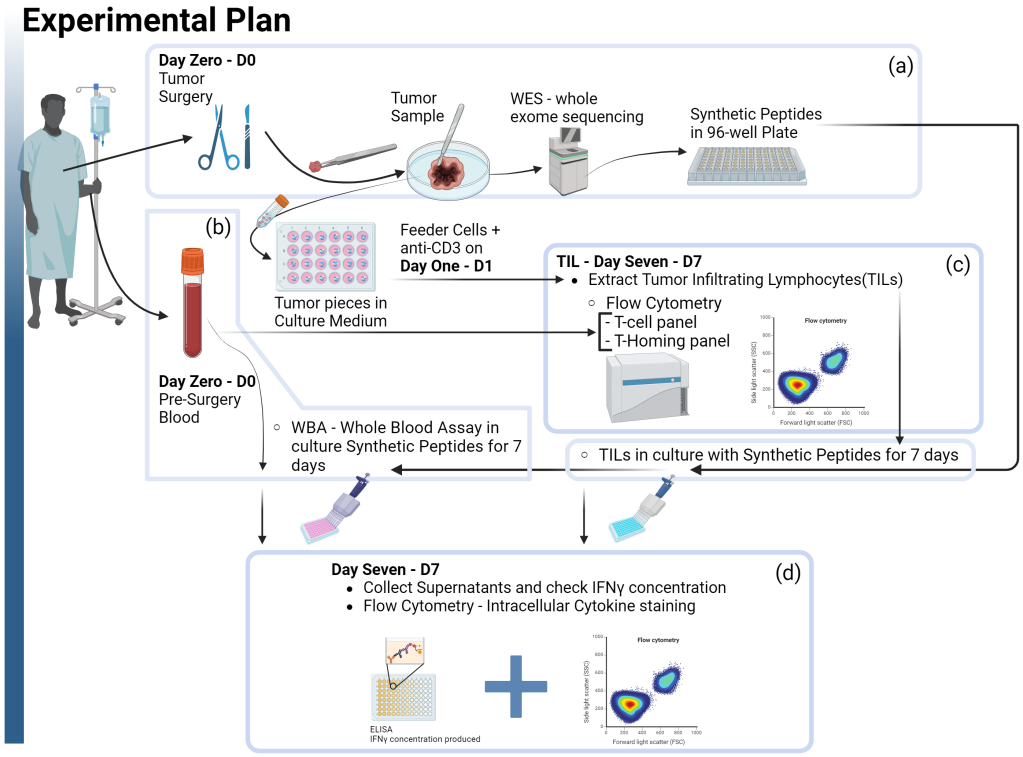


Figure 3.1: Experimental Plan - designed in BioRender^{TM}

appendix D, were cultured in polystyrene treated 24- or 96-well plates.

The TILs were expanded from tumor pieces using the optimal conditions for T-cell growth medium (medium AIM-V supplemented with interleukin 2, IL-2 (1000 IU/mL), IL-7 (1000 IU/mL), IL-15 (150 IU/mL) and IL-21(1 IU/mL) plus anti-human CD3 antibody at 30ng/mL and stimulation with allogeneic feeder cells (pool of Peripheral blood mononuclear cells (PBMC)s from three healthy donors) irradiated at 40 Gray (Gy). Moreover, after this expansion, meaning at the end of day seven, TIL from each patient was co-cultured with synthetic peptides.

TILs derived from tumor pieces cultured in medium from day zero to day seven, medium was refreshed every 3 to 4 days through half-media changes. Like in step (a) of the experimental plan, since day one until day seven of the culture period in which it starts to apply methods depicted in step (c) of the experimental plan. For the TILs co-culture assay, below step (c) in experimental plan, 24 hours after seeding the TILs with synthetic peptides, human serum was added to keep the TILs with better viability until day seven depicted in step (d) of the experimental plan.

Pre-surgery Peripheral blood from five selected patients was provided by the Pathology Unit and part of this blood was used to isolated PBMCs after separating whole venous blood over a Ficoll-Hypaque gradient. The PBMCs were then

washed twice with DPBS (Gibco) and stained using the cytometry staining for T-cell panel or T-cell activation panel and T-homing panel.

The remaining blood was used for the WBA, whole-blood assay to test the effector capacity against synthetic peptides which is explained below in IFNγ production via co-culture

section and supernatants are collected on day seven to be tested via ELISA and remaining cells are stained via ICS panel with or without CD107a marker.

3.3.2 Co-culture assays with synthetic peptides

For both assays, meaning TIL with synthetic peptides or WBA with synthetic peptides, wells with only TILs or prepared whole-blood were used as negative controls, whereas wells with TILs or prepared whole-blood in the presence of 5 pg/mL of phytohaemagglutinin (PHA) for maximal stimulation were used as positive controls and additional positive controls with the defined concentration as in table D.2. In all conditions, TILs or prepared whole-blood were incubated for 7 days, at 37°C in a humidified atmosphere containing 5% CO_2, TILs or whole-blood was seeded in pre-prepared 96-well plates round-bottom with 100μL T-cell growth medium with synthetic peptides at a concentration of 2μg/mL. The IFNγ data correspondent to the optimization of the effector versus synthetic peptide ratio to be used in the TIL synthetic peptides assay was adjusted to 1x10^4 effector cells against 1ug/mL of synthetic peptide and is reported after medium subtraction as IFNγ pg/mL/7 days.

On day zero of co-culture, for pre-surgery blood the remaining volume of whole-blood is resuspended with same volume of T-cell growth medium (AIMV (at appendix A, A.2) supplemented with IL-2 (1000 IU/mL),IL-7 (1000 IU/mL),IL-15 (150 IU/mL) and IL-21 (1 IU/mL)) and 100μL of this preparation is distributed over the pre-prepared synthetic peptide 96- well plate round bottom (at appendix A, A.2), while TILs after 7 days of expansion were washed with DPBS (at appendix A, A.2), resuspended in (AIMV (at appendix A, A.2) supplemented with IL-2 (1000IU/mL),IL-7 (1000IU/mL),IL-15 (150IU/mL) and IL-21 (1IU/mL)), counted and 100μL of this prepared solution is distributed into the pre-prepared synthetic peptide 96-well plate round bottom (at appendix A, table A.2). In all co-culture assays the synthetic peptides stay at a final concentration of 1ug/mL after distributing the prepared solutions of TILs or whole blood.

Supernatants were collected at day 7 from both co-culture assays and tested for the presence of IFNγ using the Enzyme-Linked Immunosorbent Assay, ELISA

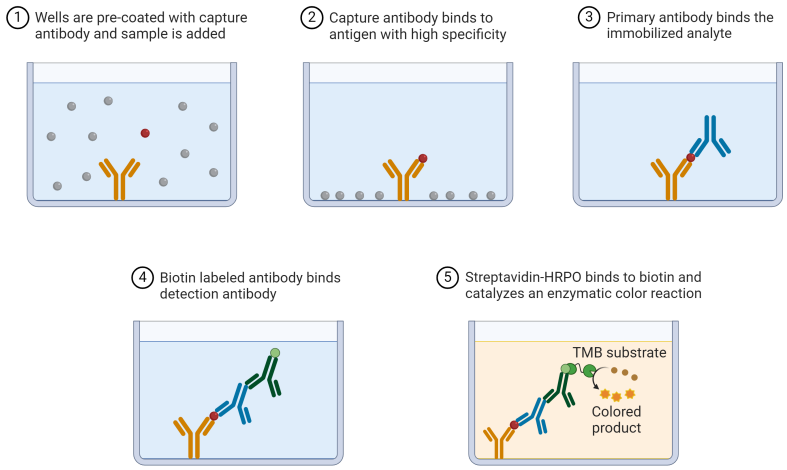


Figure 3.2: ELISA general step by step protocol

for IFNγ (Mabtech, Nacka Strand, Sweden) according to manufacturer instructions as shown in the figure 3.2. Biological duplicates were performed for all conditions. The ELISA assay returns fluorescent response which is measured by device SPARK 10M (at appendix A, table A.3) that gives optical density Optical Density (OD) values. These OD values are used in the *in silico* section.

Also cells from these incubated 96-well plates on day seven were harvested and washed to perform flow cytometry ICS on day seven step (d) from experimental plan. The staining protocol is explained below in Flow Cytometry Assays.

3.3.2.1 Flow Cytometry Assays

The performed flow cytometry assays follow a specific staining protocol which is explained below, also the figure 3.3 allows to visualize some of the terms used in the different protocols explained in the different flow cytometry assays performed. The overall scheme for flow cytometry experiment is shown in the next page.

T-cell subset panel or T-cell activation panel and T-homing panel

To monitor the phenotype of T-cells on day zero, and after their time in culture day seven, phenotypic analyses by flow cytometry were performed. For the T-cell panel or T-cell activation panel and T-homing panel after cells being washed with DPBS(at appendix A, table A.2) they are incubated for 15 minutes with LIVE/DEAD^{TM} corresponding to each panel and after incubation time, cells

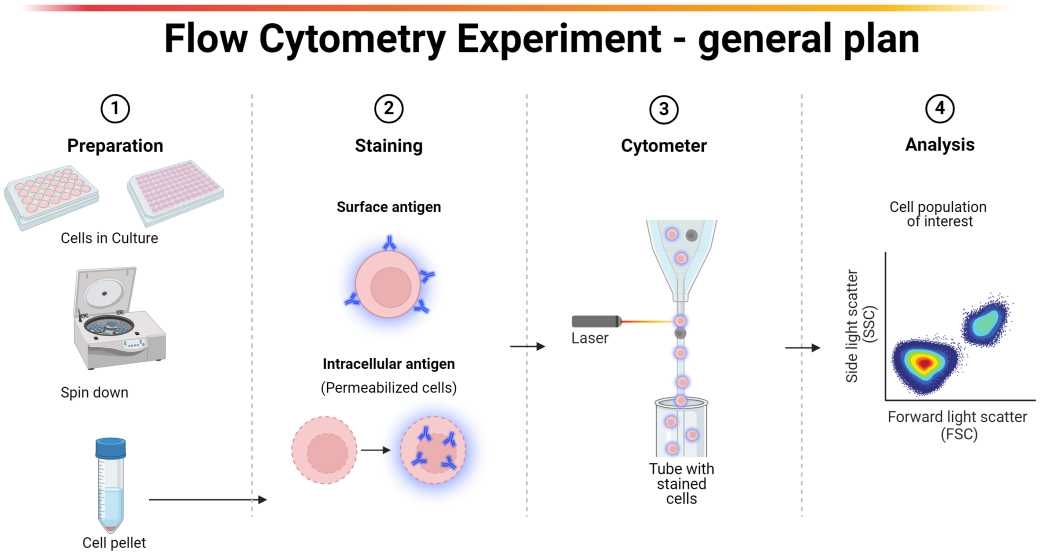


Figure 3.3: General staining plan for Flow Cytometry

are washed with DPBS again and surface/intermembrane markers that begin with CC or CX (meaning chemokines) or CD as in appendx E,table E.2 that account for a master mix corresponding to the assigned staining panel are added with the corresponding volumes, and samples are incubated for 20 minutes. After this time samples are washed again with DPBS (Gibco) with 2% heat inactivated FBS (at appendix A, table A.2) - hereafter referred to as FACS buffer, resuspended in 200μL of FACS buffer and analysed by flow cytometry (BC Cytoflex LX as in at appendix A, table A.3). Data analysis, as to gating or gauging the T-cells with positive fluorescence, was performed using FlowJo software version 10, to determine different CD or chemokine markers frequency and mean fluorescence intensity (MFI) by employing the gating strategies shown for T-cell subsets panel gating strategy in appendix Ein figure E.3, for T-cell activation panel in figure E.2 and for T-homing panel in figure E.4.

ICS-Intracellular Cytokine Staining

T-cells either TILs or PBMC present cytolytic granules that contain different types of cytolytic proteins, such as granzyme and perforin, which are released after activation of these cells to induce target cell death and release cytokines as

welll. Vesicles that contain these cytolytic proteins/cytokines present lining their membranes the lysosomal-associated membrane protein-I (LAMP-I) or CD107a, whose expression can be evaluated to assess the cytotoxic ability of an effector cell during/after its stimulation. The ICS with CD107a degranulation assay was used to determine T-cell functionality after their *in vitro* co-culture with synthetic peptides. CD107a expression was assessed after T-cells stimulation with the different synthetic peptides. Effector:target cells ratios was the same in all conditions. Briefly, T-cells extracted from TIL co-culture or whole-blood co-culture after 7 days were washed with DPBS (at appendix A, table A.2), resuspended in medium supplemented with IL-2 (1000 IU/mL), IL-7 (1000 IU/mL), IL-15 (150 IU/mL), IL- 21 (1 IU/mL), counted and manually distributed into the respective FACS tubes. Cells were co-cultured for 1 h at 37°C, in a humidified atmosphere containing 5% $C0_2$ and in the presence of the anti-CD107a PE antibody (at appendix E, table E.1). After 1 h, protein transport inhibitor containing monensin (BD GolgiStop™, BD Biosciences) was added to the medium and incubated for 5 h to inhibit trans-Golgi apparatus function. Tubes with only effector cells were used as negative controls, whereas tubes with effector cells in the presence of 1 pg/mL of phorbol-12-myristate-13-acetate (PMA; 25 ng/mL; Sigma-Aldrich) for maximal stimulation were used as positive controls. Negative and positive controls were incubated simultaneously. And a tube with effector cells and all peptides from peptides list (at appendix D, table D.3) at a concentration of 1ug/mL. After incubation, cells from all conditions were stained with the LIVE/DEAD™ and surface CD4 (Thelper), CD8 (cytotoxic) and TCR$\gamma\delta$ with the corresponding volumes described in table E.1 in appendix E.After the incubation period cells were washed with DPBS (A.2) and fixed with fix-perm buffer (BD Cytofix/Cytoperm™) incubated 20 min at room temperature then spin-down and stained with with the master mix (with fix-perm buffer) containing the remaning antibodies against different interleukins TNFα and IFNγ E.1 with an incubation period of 20 minutes. After the staining, cells were washed with FACS with perm buffer, resuspended in 200μL of FACS with perm buffer and analysed by flow cytometry (BC Cytoflex LX). Data analysis was performed using FlowJo software version 10. To determine the CD107a$^+$ and cytokine frequency and MFI, from the CD4, CD8 and TCRgd (gama-delta) T-cell frequencies by employing the gating strategies shown for ICS panel gating strategy at appendix E in figure E.1.

3.4 *In Silico* Assays

To perform data analysis of the obtained OD values from ELISA and to retrieve the analyzed relative frequencies and MFI from the T-cell populations and cytokine released relative frequency performed on flow cytometry analysis there was the demand to develop an *in silico* pipeline to obtain the values of interest to then perform the statistical analysis essential for the validation of effective and non-effective T-cell phenotypes.

The scheme in the figure 3.4 below, outlines a data analysis pipeline that incorporates statistical analysis, data visualisation and immunological assays.

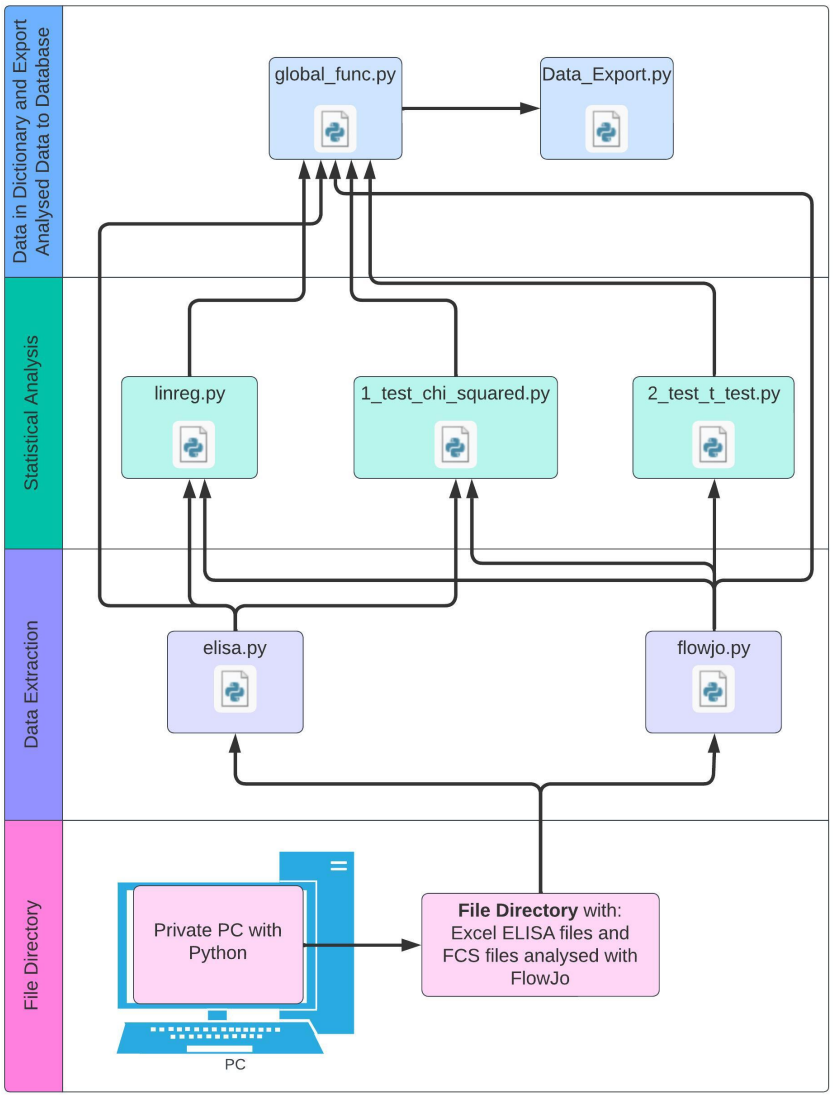


Figure 3.4: Python Scripts Pipeline - designed in Lucidchart™

The inclusion of a Python script indicates that the analysis workflow is semi-automated through scripting, which can be reproducible and efficient for other longitudinal studies.

The process starts with a dataset organised by patient and sorted by date to enable longitudinal studies.

Data pre-processing includes steps such as calculating means, performing contingency tables for chi-squared comparisons, which are critical to ensuring data quality. Statistical analysis is performed using linear regression models to identify, for example, correlations within the data.

Firstly, the user needs a private PC and to organise the data within the specific folders as shown in figure 3.5, hereafter, called file directory.

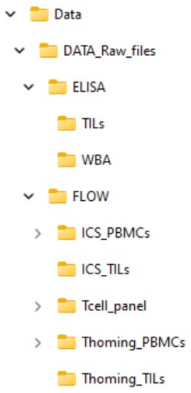
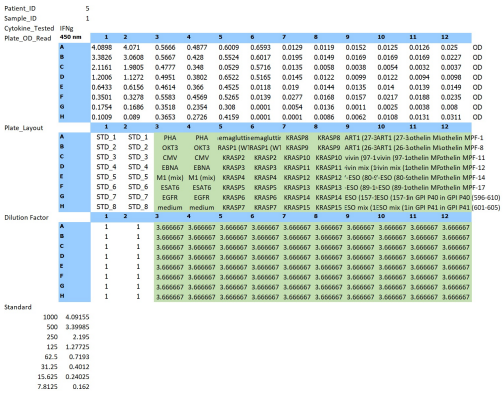


Figure 3.5: Organized File Directory structure to run the Python scripts



Patient_ID	5												
Sample_ID	1												
Cytokine_Tested	IFNg												
Plate_OD_Read	450 nm	1	2	3	4	5	6	7	8	9	10	11	12	
	A	4.0898	4.071	0.5666	0.4877	0.6009	0.6593	0.0129	0.0119	0.0152	0.0125	0.0126	0.025	OD
	B	3.3826	3.0608	0.5667	0.428	0.5524	0.6017	0.0195	0.0149	0.0169	0.0169	0.0169	0.0227	OD
	C	2.1161	1.9805	0.4777	0.348	0.0529	0.5716	0.0135	0.0058	0.0038	0.0054	0.0032	0.0037	OD
	D	1.2006	1.1272	0.4951	0.3802	0.6522	0.5165	0.0145	0.0122	0.0099	0.0122	0.0094	0.0098	OD
	E	0.6433	0.6156	0.4614	0.366	0.4525	0.0118	0.019	0.0144	0.0135	0.014	0.0139	0.0149	OD
	F	0.3501	0.3278	0.5583	0.4569	0.5265	0.0139	0.0277	0.0168	0.0157	0.0217	0.0188	0.0235	OD
	G	0.1754	0.1686	0.3518	0.2354	0.308	0.0001	0.0054	0.0136	0.0011	0.0025	0.0038	0.008	OD
	H	0.1009	0.089	0.3653	0.2726	0.4159	0.0001	0.0001	0.0086	0.0062	0.0108	0.0131	0.0311	OD
Plate_Layout		1	2	3	4	5	6	7	8	9	10	11	12	
	A	STD_1	STD_1	PHA	PHA	emagluttiinemaglutti	KRASP8	KRASP8	KRASP8	ART1 (27-3ART1 (27-1othelin Miothelin MPF-1				
	B	STD_2	STD_2	OKT3	OKT3	RASP1 (W1RASP1 (W1	KRASP9	KRASP9	KRASP9	ART1 (26-3ART1 (26-1othelin Miothelin MPF-8				
	C	STD_3	STD_3	CMV	CMV	KRASP2	KRASP2	KRASP10	KRASP10	vivin (97-1vivin (97-1othelin MPothelin MPF-11				
	D	STD_4	STD_4	EBNA	EBNA	KRASP3	KRASP3	KRASP11	KRASP11	vin mix (1vivin mix (1othelin MPothelin MPF-12				
	E	STD_5	STD_5	M1 (mix)	M1 (mix)	KRASP4	KRASP4	KRASP12	KRASP12	'ESO (80-1'-ESO (80-1othelin MPothelin MPF-14				
	F	STD_6	STD_6	ESAT6	ESAT6	KRASP5	KRASP5	KRASP13	KRASP13	ESO (89-1-ESO (89-1othelin MPothelin MPF-17				
	G	STD_7	STD_7	EGFR	EGFR	KRASP6	KRASP6	KRASP14	KRASP14	ESO (157-IESO (157-1in GPI P40 in GPI P40 (596-610)				
	H	STD_8	STD_8	medium	medium	KRASP7	KRASP7	KRASP15	KRASP15	ESO mix (1ESO mix (1in GPI P41 in GPI P41 (601-605)				
Dilution Factor		1	2	3	4	5	6	7	8	9	10	11	12	
	A	1	1	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	
	B	1	1	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	
	C	1	1	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	
	D	1	1	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	
	E	1	1	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	
	F	1	1	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	
	G	1	1	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	
	H	1	1	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	3.666667	
Standard														
	1000	4.09155												
	500	3.39985												
	250	2.195												
	125	1.27725												
	62.5	0.7193												
	31.25	0.4012												
	15.625	0.24025												
	7.8125	0.162												

Figure 3.6: ELISA excel raw data with patient ID and cytokine, 96-well plate layout, OD readings, dilution factor and standard concentration average

For ELISA, the optical density readings from the plates must be entered into an Excel file, as shown in figure 3.6, with the corresponding plate layout and to indicate the cytokine to be read in the ELISA, plus the dilution factor used for

each sample. To note that this dilution factor was previously optimized in the *in vivo* assays.

And for flow cytometry data, it requires the fcs files and the FlowJo workspace used for flow cytometry data analysis previously performed by a qualified technician or researcher.

All the Python scripts presented here require the packages listed in table B.1 in this document in appendix A.

The two Python scripts, $elisa.py$ and $flowjo.py$, are then used to extract the data and retrieve it into specific dictionaries to perform the analysis and save it in csv file format.

The data is retrieved by a python script called $global\_func.py$ that takes the data generated by $elisa.py$ and $flowjo.py$ and puts it together in a large data structure and offers functions for other scripts to navigate this data structure and easily fetch the data they need.

After that, the python script, graph_pct_graphs.py can report the bar graphs for T-cell panel, a combination of results from T-cell subset and T-cell activation panel results with average and standard deviation, from PBMC and TIL. And it also reports bar graphs for ICS and T-homing for PBMC and TIL as well.

Only then can statistical testing, an essential part of the workflow, be performed to validate whether the results extracted from flow cytometry and ELISA or flow cytometry alone provide significant evidence. All the statistical analysis modules were developed according to the general statistical scheme in figure 3.7 used to identify the correct statistical test, on the next page.

For example, when immune cell modulation occurs, the $2\_test\_t\_test.py$ for Th1 and Th2 reports independence between these cell subsets.

The $1\_test\_chi\_squared.py$ evaluates whether the percentage of T cell subsets is significantly different from the percentage of peptide interferon gamma IFNγ production to determine whether a T-cell subset may be more or less sensitive to that specific peptide.

Finally, linreg.py, which evaluates whether or not helper T cells CD4$^+$, cytotoxic T cells CD8$^+$ or gamma-delta, TCR$\gamma\delta$ T-cells are highly correlated with synthetic peptide interferon gamma production.

After all the statistical analysis modules the data generated needs to be accessed via a $data\_export.py$ which will communicate with a database that would be developed in the future in case of need.

A well-organised data structure is essential for managing complex datasets and analysing results to be stored in a database for a longitudinal study.

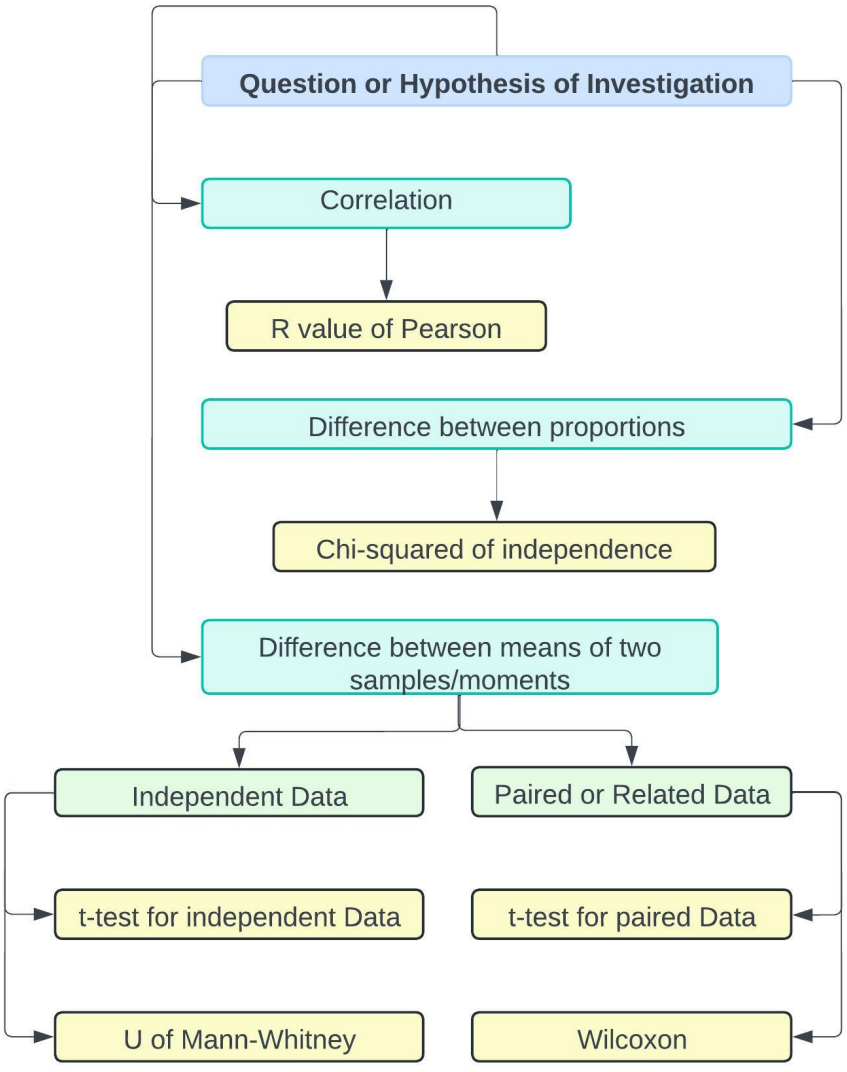


Figure 3.7: Choose and Identify the Statistical Test to be used

Below is explained in detail each python script/module developed for the proposed pipeline. For further references see the url: `https://github.com/NoronhaC/Pipeline$_$Immuno$_$MSc$_$Thesis` to get access to the developed scripts or see the appendix B in this monography.

3.4.1 *elisa.py*

The herein script/module which is located in this document in appendix B at B.1, labelled as ELISA.py, was designed for analyzing and processing data from ELISA tests related to patient samples, using spreadsheets as inputs like the one in figure 3.6 shown in the previous pages. The script includes numerous functional components, ranging from reading data from Excel files, performing curve fitting for concentration determination, to generating graphical representations and handling file operations.

Initially, the script imports necessary libraries and modules such as os, scipy.optimize for curve fitting, numpy, matplotlib for plotting, xlrd for reading Excel files, csv for file operations, and itertools.compress for filtering data based on conditions. Key functionalities include defining a function read_plate to extract patient and sample data along with cytokine information and optical density (OD) readings from given spreadsheets. This data extraction is thorough, organizing extracted values into structured dictionaries for further manipulation.

In addition, the script defines a logistic regression function log4pl to fit a four-parameter logistic model, essential for converting raw OD readings into calibrated concentrations. The equation to develop this model is shown below in equation 3.1, with a, as the lower asymptote and d, as the upper asymptote, these are the lower and upper limit of detection. The calibrated curve should report a function of this type shown in figure 3.8 displayed below.

$$y = d + \frac{a-d}{1+\left(\frac{x}{c}\right)^b} \tag{3.1}$$

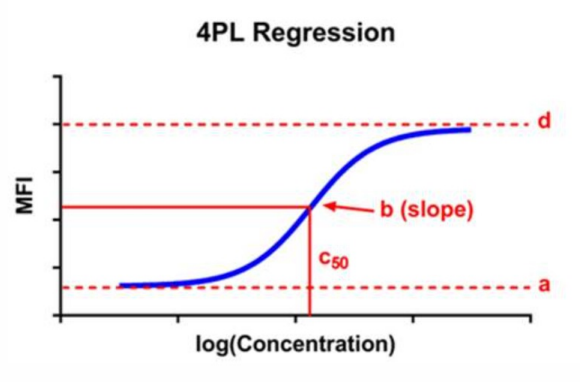


Figure 3.8: 4PL - Four Parameter Logistics Curve

There are also utility functions to convert values into percentages and calculate changes which will be used for chi-squared tests. The execution part of the script is robust, beginning with directory checks and creation, followed by iterative processing of files and sheets within specified directories. It performs data filtering, curve fitting, and plots both standard curves and concentration charts. It saves these graphical outputs and tabulated data into organized directories and files, differentiating between control substances, viral antigens, and peptides.

3.4.2 $flowjo.py$

This script/module which is located in this document in appendix B at B.1, labelled as flowjo.py, was designed for processing flow cytometry data, specifically analyzing data from FLOW workspace (.wsp) files retrived for the analysis using FlowJo 10 official software. The script uses various Python libraries including flowkit, a powerful Python module handling .fcs files, gating strategies, and generating reports on cell populations within samples. For handling cytometry data, graphviz for creating visual representations of data hierarchies, and os and shutil for file and directory management.

The script begins with a series of function definitions tailored to handle different aspects of flow cytometry data analysis. For the data extraction functions such as get_fluoro_labels and extract_data retrieve specific data from samples, including fluorescent event labels and mean fluorescence intensity MFI data.

For file handling, the functions like get_wsp_files recursively search directories for workspace files, which are essential for the analysis. As for the data Processing, the script includes detailed logic for processing data based on predefined categories such as ICS, Tcell, and T-homing, each corresponding to different types of immune cell analyses.

Finally the script uses graphviz, to generate diagrams to visually represent the hierarchy and relationships between different gates used in the cytometry analysis. If a gate was not executed, the graph would still be exported and described as n.a. for not available. The main execution block of the script sets up the environment by cleaning up and preparing directories, then iterates over each workspace file.

For each file, it determines the type of analysis PBMC or TIL and the prefix (analysis category), then calls the analyze function. This function performs a comprehensive analysis for each sample in the workspace, extracting and processing data like gate percentages and MFI comparisons, and generating graphical outputs and CSV files for each sample. The script is structured to handle a wide range of sample data, making it highly adaptable for different datasets within the domain of flow cytometry.

Furthermore it could be explored in the future with gating.ml to perform automatic gating without the need of FlowJo™10 official software.

3.4.3 $global\_func.py$

This module was designed to manage and analyze data from ELISA and flow cytometry experiments by reading from CSV files generated from the two previous scripts and structuring the data into a complex nested dictionary for easy access during analysis. It employs custom functions for reading CSV data, filtering by specific conditions, and extracting data associated with particular gates like, CD4, CD8 within the flow cytometry data.

The script begins by defining functions: read_csv reads CSV files and converts the content into dictionaries, handling special cases where the file contains irregular delimiters or is associated with specific assay types (like 'mfi' and not 'ICS'). get_data and get_gate_data are utility functions that fetch data based on given parameters like assay prefix (ELISA or FLOW), data type (mfi,as to the MFI or gate percentage, CD %), and experimental groups (like T-cell, ICS, or T-homing).

The main body of the script sets up a data structure to hold all the experimental data categorized under ELISA and FLOW experiments. It then populates this structure with data read from various CSV files located under specified directory paths, handling different data types and assay conditions (PBMC vs. TIL). This structure supports varied and extensive experimental setups, making the script versatile for different laboratory data management tasks related to immunological assays.

This entire script/module is in appendix B at B.1.

3.4.4 $graph\_pct\_graphs.py$

It generates the visualizations for data from the different selected T-cell subsets. Through defining a set of functions to compute averages and standard deviation of percentage data for specific markers (gates) on cells in different experimental groups, such as TILs and PBMC.

The main functions get_comp_data and get_group_avg retrieves and processes data for specified gates and groups using external functions from the global_func module. The script handles three types of T-cell data: ICS, Tcell, and T-homing, defined by different lists of gates. For each type, besides calculating averages and standard deviation of the gate percentages it stores the results.

The generated bar plots for each type of data, allows the user to compare results between PBMC and TIL.

This entire script/module is in appendix B at B.1.

3.4.5 $linreg.py$

This script/module located in this document in appendix, B at B.1, labelled as linreg.py, conducts linear regression analyses to examine correlation between flow cytometry gate percentages in here from $CD4^+$, $CD8^+$ and TCR$\gamma\delta$ and ELISA, over TIL and WBA. However, it can be changed to use data across different experimental conditions with PBMC, TIL and WBA.

It utilizes the Python libraries such as scipy.stats for statistical calculations and matplotlib.pyplot for plotting, along with custom functions imported from global_func explained previously.

The script defines a couple of utility functions to process data: get_avg calculates the average mean fluorescence intensity MFI of specific peptides from given data, and perform _ linear_regression runs a linear regression analysis between random uniform data points (generated within a specified range) and actual ELISA data, percentage increase in relation to the negative control of cells alone with medium. The results, including the regression coefficients and their 95% confidence intervals plus the pearson correlation coefficient, calculated using the equation 3.2 shown below. Where r stands for pearson correlation coefficient and x_i is x variable sample and x_i is x variable sample, where $\overline{x}$ is the mean of the x-variable and $\overline{y}$ is the mean of y-variable.

These processed values are printed, saved as a CSV file, and plotted graphically. Throughout its execution, the script interacts with data fetched through custom functions, performing and visualizing linear regression analyses for various immunological markers such as CD4, CD8, and GD from both Peripheral Blood

Mononuclear Cells (PBMCs) and Tumor-Infiltrating Lymphocytes (TILs), under specific analytical conditions labeled 'WBA' and 'TIL'. The script manages data directories and handles file operations extensively, ensuring organized storage of results.

$$r = \frac{\sum (x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum (x_i - \overline{x})^2 \sum (y_i - \overline{y})^2}} \tag{3.2}$$

3.4.6 $1\_test\_chi\_squared.py$

It conducts a chi-squared contingency tests on pairs of data lists, primarily comparing gate percentages from flow cytometry data against ELISA results, across different sample types and experimental setups. It leverages the scipy.stats module for statistical calculations and tkinter for simple GUI interactions to gather user input for the significance level (alpha) used in the tests.

The script is structured around several key functions. The print_comp_data function outputs pairs of data lists to the console, while perform_chi2_test handles the statistical analysis, performing chi-squared tests to assess the independence of categorical variables represented in the data lists. It uses the equation 3.3 to calculate the χ_c^2, the Chi Squared test value obtained and the $\sum$ being the sum of a fraction of O_i observed scores and expected scores being E_i.

$$\chi_c^2 = \sum \frac{(O_i - E_i)^2}{E_i} \tag{3.3}$$

This function also manages the results by logging them into CSV files depending on whether the relationships are statistically significant or not, based on the alpha value provided by the user. Files are created and managed in a specific directory structure to organize the results of dependent and independent outcomes.

Execution begins by setting the alpha level via a tkinter dialog box, ensuring user interaction for critical input, the level of significance for the test α. After setting up the necessary file structure and ensuring that previous results are cleared, the script fetches data using custom functions (get_data and get_gate_data) from a predefined structure or database. The data is then used in pairwise comparisons between different sets of experimental conditions, such as comparing PBMC and TIL within the 'T-homing' flow cytometry protocol or comparing gate percentages from T-cell analysis in PBMCs against ELISA data measuring optical density (OD) in WBA assays. It is organized to facilitate multiple comparisons, efficiently outputting data descriptions, conducting the chi-squared statistical test, and systematically saving and presenting the results.

Appendix B at B.1, contains the entire script/module.

3.4.7 $2\_test\_t\_test.py$

The present script focuses on analyzing and comparing immune response data by performing statistical t-tests on flow cytometry results from two types of samples: Peripheral Blood Mononuclear Cells, PBMC and Tumor-Infiltrating Lymphocytes, TIL. It utilizes the Python libraries scipy.stats for statistical analysis and tkinter for user input, alongside custom functions to fetch and process data.

The script starts by asking the user to input a significance level (alpha, α) through a tkinter dialog box, later used to determine the significance of the statistical tests. It then retrieves specific gate data (like Th1 and Th2 gates) from both PBMC and TIL, calculating averages for specific markers within these gates using the get_average function. Markers for Th1 gates include CXCR3 and CCR6, while Th2 gates focuses on CCR4 and CCR6.

The script then proceeds to evaluate the normality of these datasets using the Anderson-Darling test. If the data follows a normal distribution, it performs a paired t-test, using the equation 3.4. The observed mean of first sample $\overline{x_1}$ and observed mean of second sample $\overline{x_2}$ and standard deviation of first sample s_1^2 and standard deviation of second sample s_2^2 and sample size of first sample, n_1 and sample size second sample, n_2 as in the equation.

$$t = \frac{(\overline{x_1} - \overline{x_2})}{\sqrt{\frac{s_1^2}{n_1} + \frac{s_2^2}{n_2}}} \tag{3.4}$$

Otherwise, it applies a Wilcoxon signed-rank test to assess differences in the markers between PBMC and TIL for both Th1 and Th2. The outputs include t-statistics, p-values, and decision statements on whether the null hypothesis (no difference between PBMC and TIL) can be rejected or accepted based on the alpha level provided. This structured approach allows for a rigorous analysis of immunological data, aiming to understand differences in immune responses between peripheral blood and tumor environments. Appendix B at B.1, contains the script/module.

3.4.8 $data\_export.py$

Although still incomplete and not fully integrated the goal of this script was to interact with a MySQL database using a graphical user interface (GUI) created with

tkinter and ttkbootstrap. It facilitates data entry and querying within a biomedical research context, focusing on data management tasks associated with ELISA results, patient samples, and other related clinical data.

The core functionality includes connecting to a MySQL database and executing SQL queries to fetch or manipulate data. The query_database function performs SQL queries, specifically handling SELECT statements by fetching all results from the database cursor. The GUI component is built using tkinter and styled with ttkbootstrap, which provides themed widgets for a more visually appealing interface.

The script sets up a dynamic form based on selected database tables, such as 'elisa', 'patient', 'sample', and various others.

Users can select a table from a dropdown menu, which triggers the build_menu function to create a data entry form tailored to the specific columns of the chosen table. The form fields are dynamically adjusted according to the table structure defined in table_cols dictionary, and user inputs are gathered using dropdowns or text entries according to the form_items dictionary, which specifies the required data type and possible values for each field.

Interaction starts with the main window where users pick a database table, and upon selection, the relevant form is generated for data entry. This interface is part of a larger system designed to streamline data entry and querying processes in biomedical research, making it easier to manage complex datasets related to various types of clinical analyses. In the Appendix B at B.1, it is possible to consult this script/module.

4

Discussion and Results

The presentation of results is accompanied by a discussion. Firstly there is the presentation of ELISA results from TIL and WBA.

Secondly the results from correlation between CD4-Thelper, CD8-cytotoxic and TCR$\gamma\delta$ T-cells relative frequencies from ICS, IFNγ release from flow cytometry against peptides, versus IFNγ concentration results from ELISA.

Thirdly the results from the Chi-squared assays, discussed with level of significance of 1 and 5%. Finally we present the t-test comparison of Th1 from PBMC versus TIL, and Th2 from PBMC versus TIL results and a desicion based on the p-value.

the results from flow cytometry assays, as a mean from main populations with interesting results and statistical comparison test with results that correlate with the study.

4.1 ELISA results from TILs and WBA

In these assays, the variable under study is the average concentration of interferon gamma IFNγ. The results per patient are shown below of the synthetic peptide recognition assay, displaying the immune response of the patient and eliciting the function of immune cells in the peripheral blood versus the enriched TILs.

As previously explained in the Methods section the WBA was performed with blood collected in pre-surgery setting.

The IFNγ concentration reported here is without the medium control concentration, meaning that values presented here relate to a positive response above medium control.

The differences in IFNγ production in response to the same stimuli suggest that these immune-monitoring functional assays need to be carefully calibrated and standardized to ensure consistent results across different tests and time points.

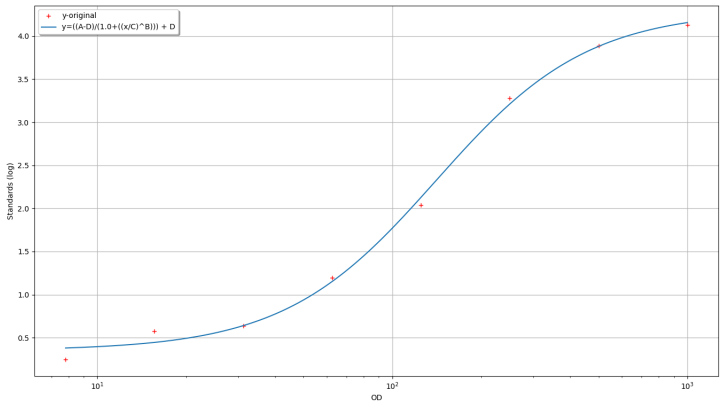


Figure 4.1: 4PL - four parameter logistics curve calibration from ELISA samples

However, some patients, like patient three and four reveal higher functionality of TILs when compared to immune cells from whole blood. This result is congruent with the literature reviewed in the literature review chapter as to TILs being capable of recognizing a broader range of tumor associated antigens and neoantigens with single nucleotide variants used in this experiment as synthetic peptides.

All the ELISA results were extracted through the use of 4 parameter logistic curve, all reported a coefficient of determination (r^2) above 0.95, proving that all the ELISA in here have a almost perfect interpolated curve, has it can be seen in figure 4.1, presented on the top of this page.

ELISA is a cost-effective option compared to IsoPlexis^{TM} [138] or multiplex analysis platforms, offering a budget-friendly means of cytokine detection.

However, its primary limitation lies in its ability to measure only one cytokine at a time, unlike more comprehensive, but expensive like multiplex technologies.

4.1.1 Patient 1

From the 2 graphs in the presented figure 4.2, we can see an immune response, TILs against synthetic peptide assay are the response of two peptides, KRASP3 and KRASP1 (wild type (WT)), showing a similar level of IFNγ production, around 0.456-0.460 pg/mL.

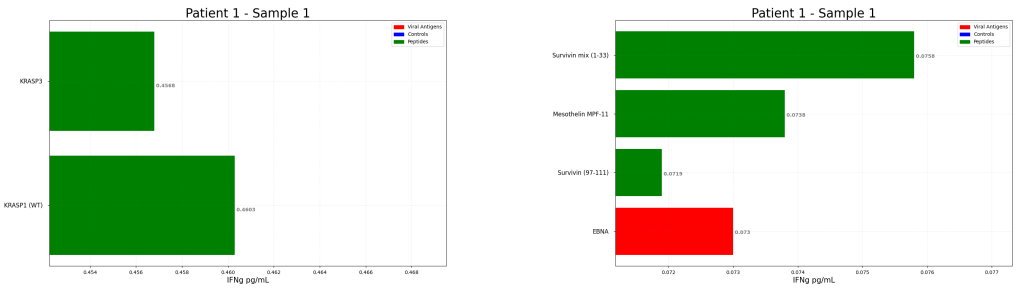
Both peptides elicit a similar immune response as indicated by the almost identical IFNγ levels, which suggests these peptides may be similarly recognized by the patient's immune system present in tumor infiltrating lymphocytes.

The WBA expresses a response for the control (EBNA), a standard viral antigen used to gauge the immune response's baseline, and peptides Survivin mix (1-33) elicits the highest response at approximately 0.0758 pg/mL, Mesothelin MPF-11, and Survivin (97-111) show a moderate response. Here, the peptides elicit stronger immune responses than the control, as shown by higher IFNγ concentration levels.

Comparing the two graphs, we see a wider range of IFNγ concentrations in the second graph. This suggests a more variable immune response to the different peptides presented in TILs against synthetic peptides.

The control (EBNA) in the second graph (b) has a lower IFNγ concentration than the peptides, suggesting that the peptides are likely to be more relevant targets for the immune response in this particular patient.

The peripheral blood against the synthetic peptides, from the WBA shown in the second graph reveals a very narrow range of IFNγ concentration, whereas the first graph shows both higher values and a broader range. This could indicate that TILs are more immunogenic or that they represent a broader repertoire of antigens.



(a) Patient 1 - TIL - IFNγ concentration

(b) Patient 1 - WBA - IFNγ concentration

Figure 4.2: Patient 1 - ELISA results from IFNγ production against synthetic peptides

4.1.2 Patient 2

At the bottom of the page the are results for this patient in the graphs, in figure 4.3.

The WBA displays the highest concentration of IFNγ is in response to the 'NY-ESO mix (1-43)', whereas the WBA, 'Mesothelin MPF-8' and 'Mesothelin MPF-1' induced the highest IFNγ production. This difference may point to the specific immunogenicity of these peptides in the context of the patient's immune system.

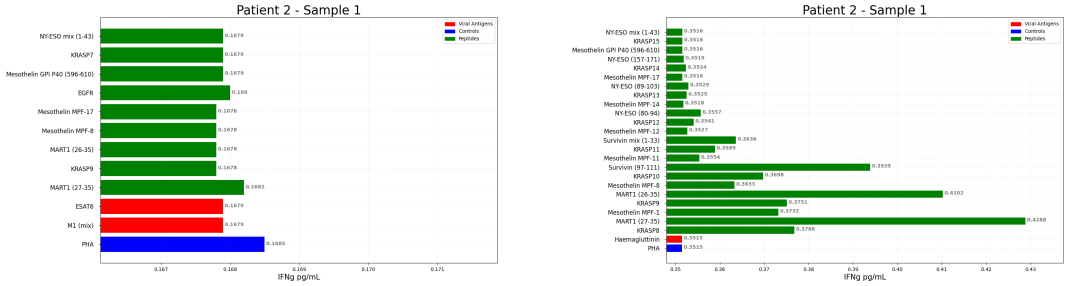
The second graph (b), as to the WBA, displays a notable immune response to 'Survivin mix (1-33)', 'Survivin (97-111)', and several 'Mesothelin MPF' and 'NY-ESO' peptides which are not present in the first graph the TIL assay.

As an overall the range of IFNγ Concentration: The range of IFNγ concentration in the second graph, as WBA recognition, is broader (approximately 0.35 to 0.43 pg/mL) compared to the first graph, as to TIL recognition (approximately 0.167 to 0.171 pg/mL). This suggests that viral antigens, peptides, and controls in the second graph elicited a wider range of immune responses as measured by IFNγ production.

By comparing the two graphs, specific antigens such as 'Mesothelin GPI P40 (596-610)' and 'NY-ESO mix (1-43)' can be seen in both graphs. They elicit different levels of IFNγ production in the two graphs.

As to the controls, they are consistent, PHA and M1 mix in the first graph, Haemagglutinin and PHA in the second, show a consistent low-level IFNγ production in both graphs. The consistency of controls responses suggests that the experimental setup is reliable.

As a conclusion, the level of IFNγ production is higher in WBA, suggesting that the patient's immune system was more activated.



(a) Patient 2 - TIL - IFNγ concentration

(b) Patient 2 - WBA - IFNγ concentration

Figure 4.3: Patient 2 - ELISA results from IFNγ production against synthetic peptides

4.1.3 Patient 3

This patient results are shown in the graphs presented in the figure 4.4 at the bottom of the page.

The range of IFNγ concentrations in the second graph(b), the WBA, approximately 0.36 to 0.43 pg/mL which is significantly higher than the first graph, the TIL, of approximately 0.11 to 0.15 pg/mL.

This suggests that the WBA detected a stronger immune response or had a higher sensitivity.

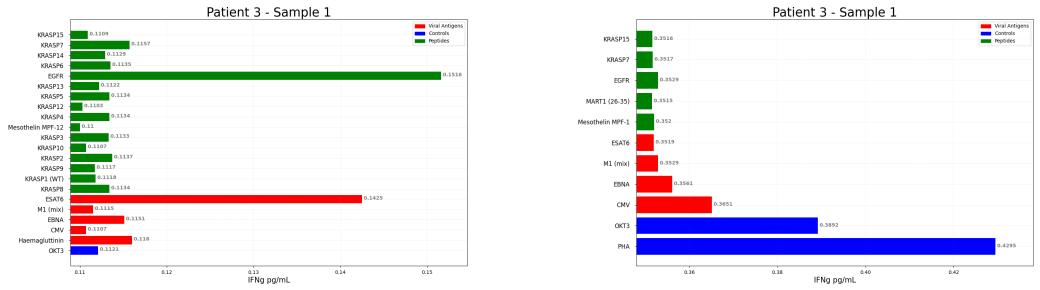
In the TIL, the highest IFNγ production is in response to the 'EGFR', whereas in the second graph, the 'KRAS' peptides seem to elicit the strongest response.The controls (PHA, OKT3, M1 mix, Haemagglutinin, EBNA, and CMV) show varied responses between the two graphs. For instance, PHA shows a higher IFNγ response in the WBA compared to the first.

This may suggest differences in the functionality of the controls between the two assays or variability in the patient's response.

Antigens such as 'KRAS' peptides are present in both graphs, with 'KRAS' peptides showing higher IFNγ levels in the second graph, suggesting these peptides may be more immunogenic.

There is a trend toward stronger immune responses to certain 'KRAS' peptides, and the WBA captures a broader range of these responses.

The TIL against peptide recognition shows a more uniform response across different peptides, with a narrower concentration range. In contrast, the WBA which shows a much wider range of responses, with 'KRAS' peptides showing the highest levels of IFNγ, which could indicate a robust immune reaction to these particular peptides.



(a) Patient 3 - TIL - IFNγ concentration

(b) Patient 3 - WBA - IFNγ concentration

Figure 4.4: Patient 3 - ELISA results from IFNγ production against synthetic peptides

4.1.4 Patient 4

The results for this patient are depicted in the graphs on the figure 4.5 below. In the the first graph (a),TILs displays a narrower range of IFNγ concentrations, approximately 0.24 to 0.30 pg/mL, with the highest responses to the control (PHA) and viral antigens (CMV and EBNA). The WBA shows a range of approximately 0.188 to 0.195 pg/mL with the highest responses to peptides (specifically Mesothelin GPI P41 (601-605) and NY-ESO mix (1-43)).

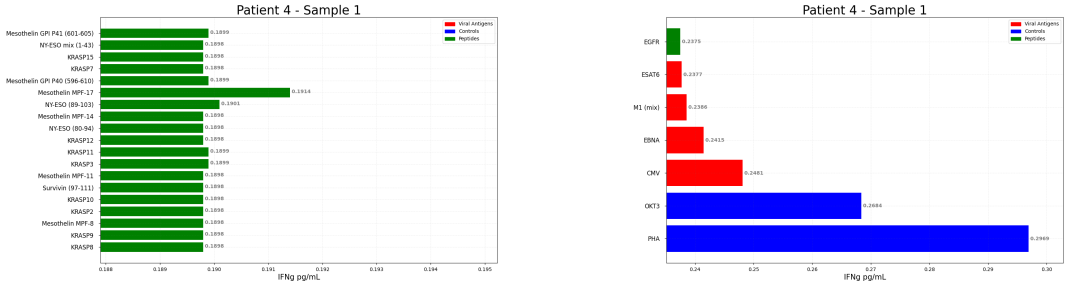
In the TILs, the immune response to peptides is minimal when compared to the response to viral antigens and controls. In contrast, the WBA shows a more robust immune response to peptides, with relatively little variation among the different peptides tested.

The PHA, show a significantly higher IFNγ production in the TILs, which serves as a positive control to indicate the functionality of the assay. The WBA does not report these controls.

The level of IFNγ in the TIL has the highest response to PHA, a common mitogen used to stimulate T-cell activity in these types of assays. In the WBA, the peptides related to 'Mesothelin GPI P41 (601-605)' and 'NY-ESO mix (1-43)' evoke the highest responses.

The TILs response may indicate a general state of immune activation, as evidenced by the higher IFNγ levels in response to viral antigens and controls. This could be indicative of an immune system that is already primed or reactive to these stimuli.

The differences between the two graphs (a) and (b) may be due to the differences in patient's immune system reacts to these stimuli from peripheral blood versus lymphocytes near the tumor.



(a) Patient 4 - TIL - IFNγ concentration

(b) Patient 4 - WBA - IFNγ concentration

Figure 4.5: Patient 4 - ELISA results from IFNγ production against synthetic peptides

4.1.5 Patient 5

The results for this patient are shown in graphs presented in the figure 4.6 below.

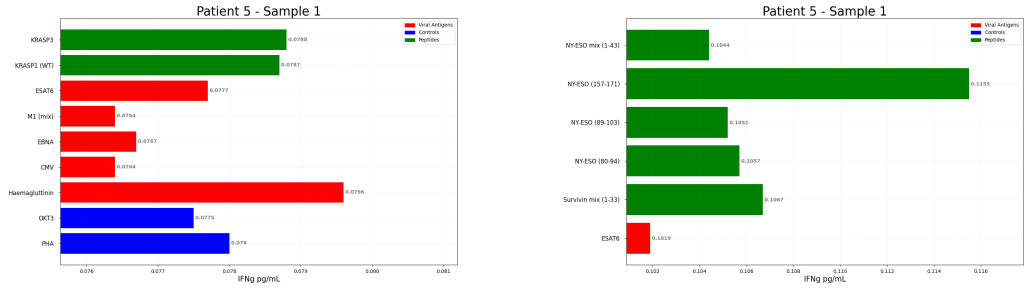
In this patient the WBA show a higher range of IFNγ concentrations (approximately 0.102 to 0.116 pg/mL) than the TIL (approximately 0.076 to 0.080 pg/mL). This could indicate that the immune response to the peptides tested in the WBA was stronger or more varied. The TIL indicates that 'KRASP3' and 'KRASP1 (WT)' peptides elicited the highest IFNγ concentrations, which suggests a significant immune response to these peptides.

In contrast, the WBA shows the 'NY-ESO mix (1-43)' and other 'NY-ESO' peptides producing the highest IFNγ levels, indicating a stronger immune response to these peptides compared to 'ESA6', which has a lower response.

The control responses in the TIL (PHA, OKT3, Haemagglutinin, CMV, EBNA) show lower IFNγ production compared to the peptides, while in the WBA 'ESA6', a viral antigen, elicited the least IFNγ production among all the tested substances. The consistency of the control responses within each graph suggests that the assay conditions were controlled effectively.

The graphs (a) and (b) show that different antigens and peptides can elicit fluctuating levels of IFNγ production, reflecting the patient's immune response.

The peptides 'KRASP3' and 'KRASP1 (WT)' peptides in the TILs, and the 'NY-ESO mix (1-43)' in WBA, elicited the strongest immune responses as measured by IFNγ production. This suggests that these particular peptides may be the most immunogenic for this patient or that induced higher IFNγ production under *in vitro* testing.



(a) Patient 5 - TIL - IFNγ concentration

(b) Patient 5 - WBA - IFNγ concentration

Figure 4.6: Patient 5 - ELISA results from IFNγ production against synthetic peptides

4.2 T-cell panel results from Flow Cytometry

The results presented here are from the T-cell activation panel and T-cell subset panel from the five patients sample are in figure 4.7.

The percentages of precursor and central memory (TCM) $CD8^+$ T cells are relatively similar in PBMCs and TILs, with wide variability between samples and within the population. The effector T-cells (TEFF) from PBMCs present a similar percentage to the TILs as to the effector memory (TEM) compared to effector T-cells (TEFF), TEM are present in higher percentages in PBMCs and TILs. This could indicate a more activated or differentiated state in the bloodstream as TEM are moderately higher than TILs.

Regarding tissue residency and activation, $CD8^+$ $CD103^+$ T-cells are markedly higher in TILs, which fits with CD103 association with tissue-resident memory T-cells. Similarly, $CD8^+$ T-cells expressing activation markers ($CD95^+$, $CD39^+$, $CD69^+$) are also present in greater percentages in TILs. Reporting in a clinical setting the CD39 CD69 T-cells profile may represent a good prognosis if they are $CD39^-$, $CD69^-$ or a bad prognosis in the case of being $CD39^+$, $CD69^+$, since there is association with non-responders to TIL therapy according to the Rosenberg et al.[139]

The CD95 marker is more present in PBMCs as compared to the tumor microenvironment TILs revealing the exhaustion of TILs as compared to PBMCs, which fits with T-cell fitness of PBMCs versus TILs.

Overall, this data suggests a differential state of activation and memory $CD95^+$ status of $CD8^+$ T-cells when comparing the systemic circulation PBMCs to the tumor microenvironment TILs. There is a pronounced activated and tissue-resident phenotype $CD103^+$ within the TIL.

If a gate was not included in the flowjo workspace then it is not included in the difference T-cell panel results, demonstrating that the experimental was not performed correctly or that the FlowJo™ workspace analysis excluded the gate on the extracted and presented population in the graph as n.a. for not available.

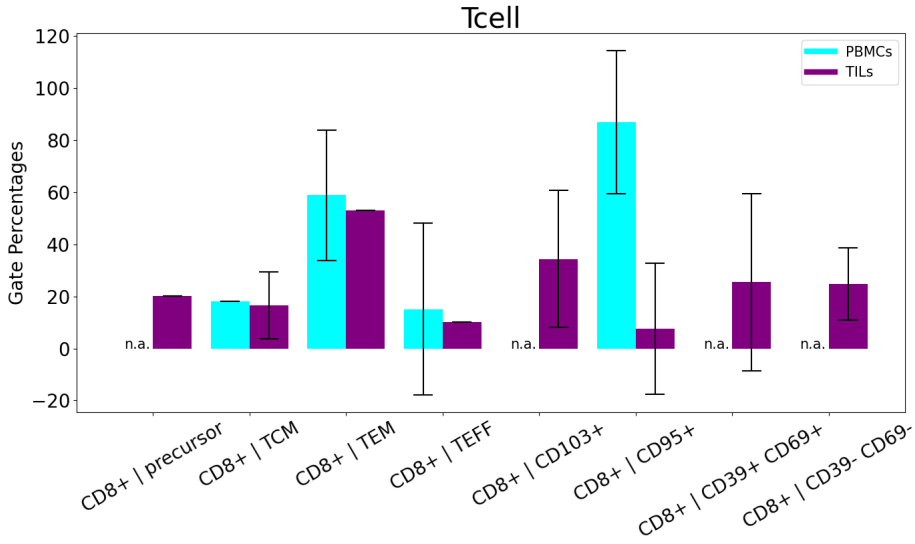


Figure 4.7: T-cell subset and activation panel results

4.3 Intracellular Cytokine Staining results

The Intracellular cytokine staining are shown in figure 4.8 at the bottom of this page shows a comparison of the percentages of different T cell subsets between PBMCs and TILs. This bar graph presents the percentage of $CD4^+$ and $CD8^+$ T-cells producing IFNγ as detected by Intracellular Cytokine Staining (ICS) within PBMC and TIL.

Concerning $CD4^+$, both PBMC and TIL show a similar low percentage of $CD4^+$ T-cells producing IFNγ, with TIL having slightly less variability as indicated by shorter error bars. The presence of IFNγ-producing $CD4^+$ T-cells might suggests a Th1-type immune response.

Regarding $CD8^+$ a higher percentage of $CD8^+$ T-cells in PBMC are producing IFNγ compared to $CD8^+$ T-cells in TIL. The broader error bars for PBMC indicate more variability in the IFNγ production within this group. $CD8^+$ T-cells are cytotoxic and their IFNγ production is crucial for the direct killing of tumor cells, so this result correlates with the literature review.

The graph suggests that systemic $CD8^+$ T-cells in PBMC have a higher propensity to produce IFNγ compared to those localized within the tumor microenvironment IFNγ, which could indicate suppression or exhaustion of TIL in the tumor milieu or perhaps different stages of activation.

For TCR$\gamma\delta$ T-cells IFNγ positive indicate that TILs have a more robust response compared to PBMCs. The larger variability in TIL could suggest heterogeneous responses within the TIL population or differences in the tumor microenvironment affecting IFNγ production.

Note that the error bars indicate the standard deviation within the sampled population for each group, showing how the response varies among the T-cells.

Shorter error bars for TIL could suggest a more homogeneous response within the tumor environment, while longer bars for PBMC could indicate a more diverse immune status in the blood.

The current graph does not show the CD107a degranulation marker, which could provide an additional indication of T-cell degranulation upon antigen encounter. In this experimental setting, the synthetic peptide would resemble the tumour antigen.

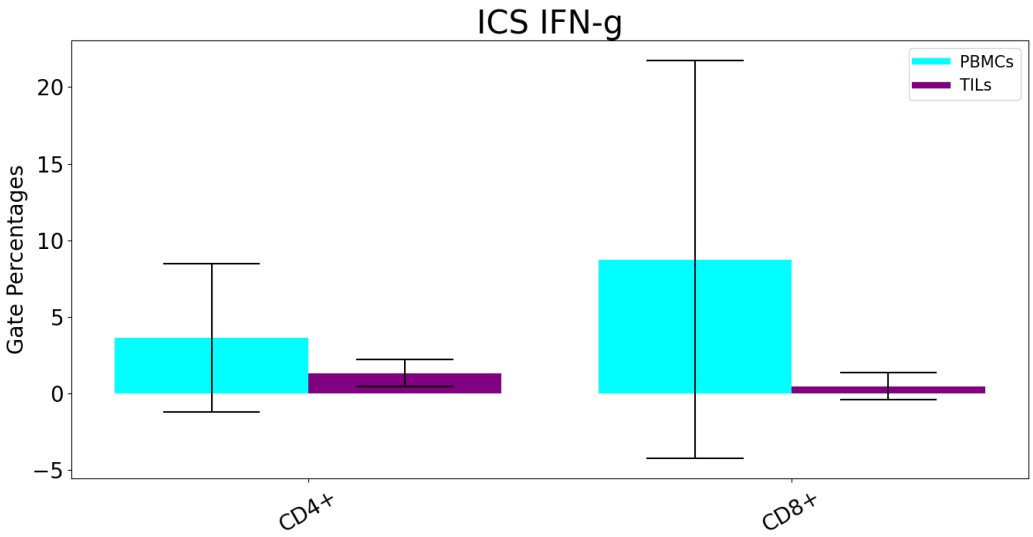


Figure 4.8: Intracellular Cytokine Staining results for $CD4^+$, $CD8^+$ and $TCR\gamma\delta^+$

4.4 Chi-squared results comparing flow cytometry results against ELISA results

A comparison between T-cell responses as measured by ELISA from Whole Blood Assay WBA and Intracellular Cytokine Staining ICS of PBMC, results that were presented previously.

Specifically, the immune responses of various T-cell subsets, including CD8, CD4, and TCR$\gamma\delta$ T-cells, are evaluated for their production of IFNγ in reaction to synthetic peptides.

Revealing with a level of significance of 1% and 5% that Survivin mix (1-33), Survivin (97-111) and Mesothelin MPF-11 reveals a level of dependency with the different gated T-cells. Suggesting a level of T-cell functionality from this group of patients against these peptides out from 30 peptides used in the experimental plan in the co-culture assay.

4.5 Linear Regression Results

This data analysis was designed to assess the release of interferon-gamma IFNγ in response to synthetic peptides, as a measure of immune reactivity. Using intracellular cytokine staining (ICS) flow cytometry method, we evaluated the relative frequency of IFNγ from CD4, CD8, and TCD$\gamma\delta$ T-cells versus the detection of IFNγ after a 7-day culture detected via ELISA, comparing the relative frequency of these T-cell populations in peripheral blood mononuclear cells (PBMCs) and tumor-infiltrating lymphocytes (TILs) against the IFNγ released when in contact with a range of synthetic peptides.

The differences in IFNγ levels released in reaction with these peptides provide insights into the specificity and potency of the immune response in the context of both systemic and tumor-localized immune environments. This approach could offers valuable information for immunological assessment of potential T-cell phenotype weighting more or less in the IFNγ production.

The results here are an average from all the patient samples, since all had epithelial cancer.

It does not show linear correlation because of 'bulk' immune cells and not sorted or because it is mechanistically impossible to correlate the IFNγ present in the supernatant from co-culture with synthetic peptides with ICS, it remains therfore unclear.

This may be partly due to the fact that thawing the supernatants from the co-culture assay to perform the ELISA may degrade the IFNγ cytokine, or it may be necessary to discard outliers in the presented graphs.

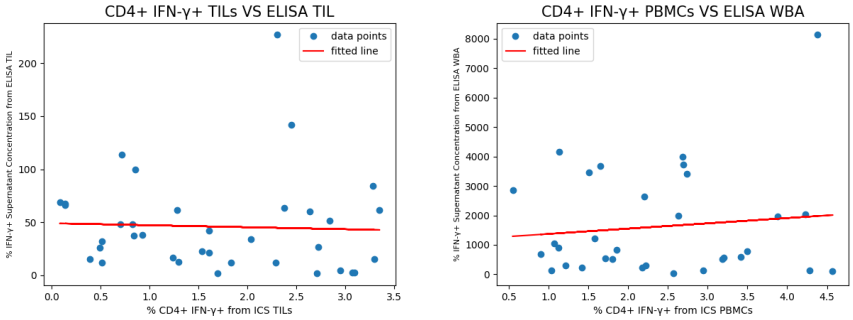
The values of the ELISA from PBMCs are higher due to the presence of both CD4, CD8 T cells and NK cells resulting in higher concentrations of IFNγ present in the supernatant.

It may be interesting to improve such assay and analysis by separating per synthetic peptide cytokine response to reach a clear analysis.

4.5.1 CD4$^+$ versus IFNγ^+ production from ELISA

TIL and WBA

The linear correlation is low. The coefficient of determination is 0.013 for TIL and 0.006 for PBMC derived from whole blood in pre-surgery. This can be seen in the figure 4.9 below, with the two presented graphs.



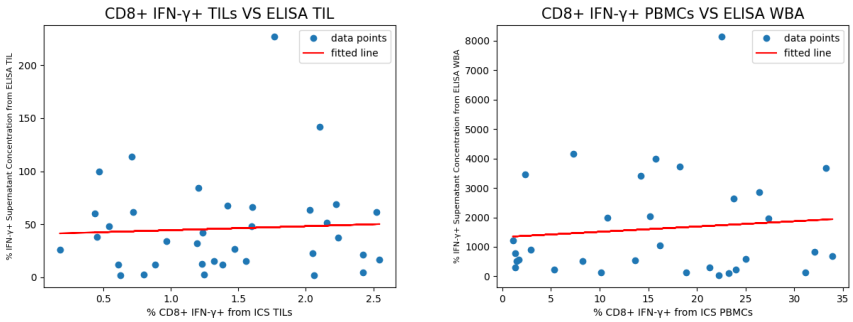
(a) ICS CD4 TIL IFNγ^+ (b) ICS CD4 PBMC IFNγ^+

Figure 4.9: ICS CD4 IFNγ^+ versus ELISA results from IFNγ production against synthetic peptides

4.5.2 CD8$^+$ versus IFNγ^+ production from ELISA

TIL and WBA

The coefficient of determination is 0.001 for TIL and 0.008 for PBMC derived from whole blood in pre-surgery. This can be seen in the figure 4.10 below with the two generated graphs.



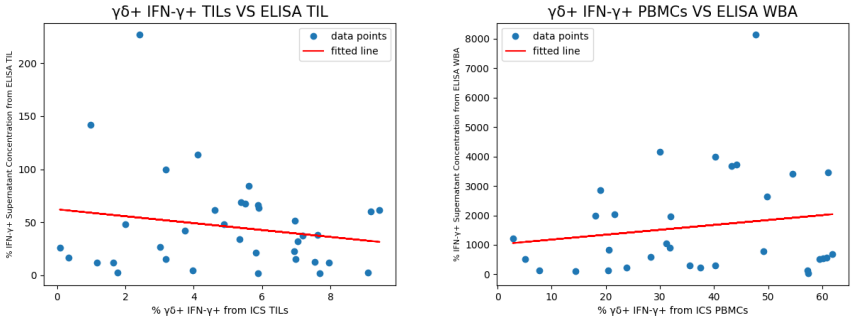
(a) ICS CD8 TIL IFNγ^+ (b) ICS CD8 PBMC IFNγ^+

Figure 4.10: ICS CD8 IFNγ^+ versus ELISA results from IFNγ production against synthetic peptides

4.5.3 TCR$\gamma\delta^+$ versus IFNγ^+ production from ELISA

TIL and WBA

In here the linear correlation reveals a low degree. The coefficient of determination is 0.116 for TIL and 0.023 for PBMC derived from whole blood in pre-surgery. This can be seen in the figure 4.11 below with the two graphs.



(a) ICS TCR$\gamma\delta^+$ TIL IFNγ^+ (b) ICS TCR$\gamma\delta^+$ PBMC IFNγ^+

Figure 4.11: ICS TCR$\gamma\delta^+$ IFNγ^+ versus ELISA results from IFN production against synthetic peptides

As a final conclusion, the linear regression here presented is not a good correlation, this can be explained to the fact that these are non-sorted T-cells, meaning that with this analysis performed on non-sorted cells it is difficult to elicit any great correlation between T-cell phenotype versus IFNγ production from ELISA.

However, it can be used and explored for comparison between T-cell phenotype of sorted cells from the same phenotype against peptides of potential interest, so this script can be applied in that context.

Furthermore, this module can be improved to perform the correlation between IFNγ produced during intracellular cytokine staining versus the synthetic peptides co-culture assay that reports IFNγ as well.

4.6 T-homing Assay Results

T-homing bar graph in the figure 4.12 below, shows a comparison of the percentages of different T cell subsets between peripheral blood mononuclear cells PBMCs and tumor-infiltrating lymphocytes TILs. It measures the gate percentages for four different T-cell types: $CD4^+$ Th1, $CD4^+$ Th2, $CD4^+$ Th17, and $CD8^+$ T-scm.

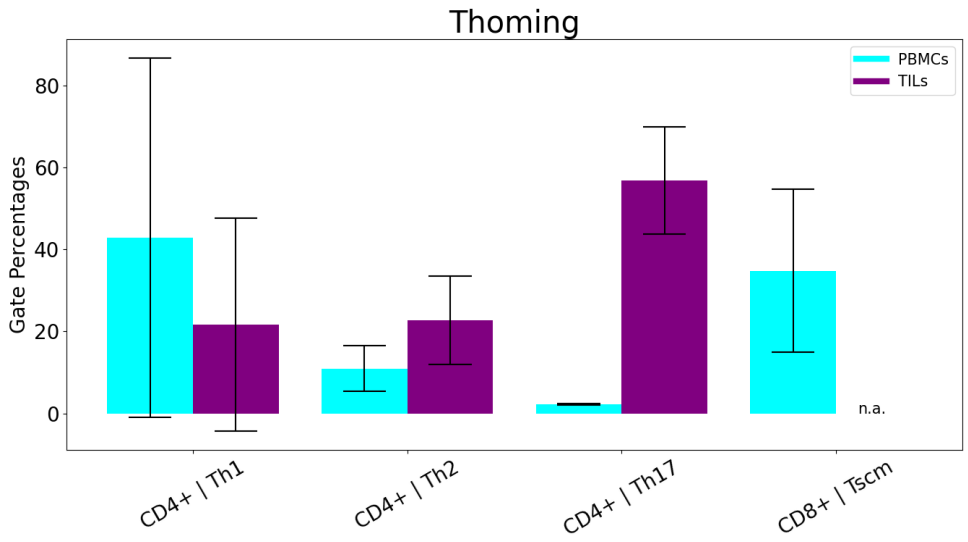


Figure 4.12: T-homing results, average from 5 patient sample

The percentage of $CD4^+$ Th1 cells is higher in PBMCs than in TILs. PBMCs have a lower percentage of $CD4^+$ Th2 cells compared to TILs, but the difference is not as pronounced as with $CD4^+$ Th1 cells. The percentage of $CD4^+$ Th17 cells is significantly higher in TILs than in PBMCs. $CD8^+$ T-scm cells are present in PBMCs only since there was a mistake while running the *in vitro* assay for T-homing on TILs. Overall, this suggests that the composition of T-cell subsets in the tumor environment is different from that in the peripheral blood, which is expected since TILs are selected for their ability to recognize and respond to tumor antigens. The higher percentage of $CD4^+$ Th17 cells among TILs might indicate a role for this subset in the immune response against the tumor, while the lower percentages of $CD4^+$ Th1 and $CD8^+$ T-scm cells in TILs could suggest a reduced presence or recruitment of these cells to the tumor site or a consumption of these cells within the tumor microenvironment. These relationships of dependency are presented below.

4.6.1 Chi-squared test results for T-homing

From this analysis it is possible to extract some relations of dependency based of the calculated p-value. The results presented below appeared due to the obtained p-value being lower than the significance level of 1% and 5%.

By comparing gate relative frequencies from flow cytometry assays of T-homing from TILs against PBMCs we can verify the relationships between T-cell phenotypes as stated in the table 4.1 below.

Table 4.1: Chi-squared, T-homing PBMC versus TILs

PBMCs	versus	TILs
Th2 - $CD4^+$ $CCR6^-$ $CCR4^+$	vs	Th17 - $CD4^+$ $CCR6^+$ $CCR4^+$ $CXCR3^-$
Th2 - $CD4^+$ $CCR6^-$ $CCR4^+$	vs	Th1 - $CD4^+$ $CCR4^-$ $CXCR3^+$ $CXCR3^+$
Th1 - $CD4^+$ $CCR4^-$ $CXCR3^+$	vs	Th17 - $CD4^+$ $CCR6^+$ $CCR4^+$ $CXCR3^-$
Th1 - $CD4^+$ $CCR4^-$ $CXCR3^+$	vs	Th2 - $CD4^+$ $CCR6^-$ $CCR4^+$ $CXCR3^-$
Tscm - $CD8^+$ $CD95^+$ $CD45RA^+$ $CCR7^+$	vs	Th17 - $CD4^+$ $CCR6^+$ $CCR4^+$ $CXCR3^-$
Tscm - $CD8^+$ $CD95^+$ $CD45RA^+$ $CCR7^+$	vs	Th1 - $CD4^+$ $CCR4^-$ $CXCR3^+$ $CXCR3^+$
Tscm - $CD8^+$ $CD95^+$ $CD45RA^+$ $CCR7^+$	vs	Th2 - $CD4^+$ $CCR6^-$ $CCR4^+$ $CXCR3^-$

4.6.2 T-test to compare MFI of Th1 and Th2 in TILs and PBMCs

As to comparing Th1 cells in Peripheral Blood Mononuclear Cells (PBMCs) MFI value to Th1 cells in Tumor-Infiltrating Lymphocytes (TILs) MFI value. The results are shown in figure 4.13 extracted from the Python script ”2_t_test.py” shown in the bottom.

For Th1 PBMCs versus Th1 TILs, the t-statistic is 4.41844, and the p-value is approximately 0.14165. The p-value is higher than the typical level of significance alpha of 0.05 suggesting that there is no statistically significant difference between the mean responses of Th1 PBMCs and Th1 TILs.

As for the Th2 PBMCs versus Th2 TILs, the second test compares Th2 cells in PBMCs to Th2 cells in TILs.

The t-statistic is 4.48855, and the p-value is approximately 0.13955. Similarly, this p-value does not indicate a statistically significant difference between the mean responses of Th2 PBMCs and Th2 TILs.

In summary, the statistical tests performed did not find significant differences between the mean values of Th1 and Th2 cells in PBMCs compared to TILs. Finally, even though there is a level of dependency between the crossed population situtation, this is Th1 against Th2, reported with the chi-squared test presented in above.

```
In [1]: runfile('C:/Users/pedro/1_thesis_scripts_Msc/3_Scripts_Methods/2_test_t_test.py', wdir='C:/Users/pedro/
1_thesis_scripts_Msc/3_Scripts_Methods')
Press ENTER to continue...
{'Th1_PBMCs': [1832.535, 1076.4], 'Th2_PBMCs': [1767.87, 1132.485], 'Th1_TILs': [1088.972, 607.293],
'Th2_TILs': [617.667, 401.41]}
Th1:
T-stat: 4.418449587547731       P-value:0.1416950294027229
independent: Th1 PBMCs and Th1 TILs do not have a significant relation
Th2:
T-stat: 4.488552423126107       P-value:0.13955276103673764
independent: Th2 PBMCs and Th2 TILs do not have a significant relation
```

Figure 4.13: t-test results - Th1 PBMC versus Th1 TIL and Th2 PBMC versus Th2 TIL

Conclusions and Future Work

Cancer essentially arises from a collection of mutations that cause cells to behave abnormally, sometimes in predictable ways and at other times randomly. These aberrations distinguish tumor cells from their normal counterparts, despite both originating from the same organism.

At their core, however, tumor cells diverge significantly from normal cells in terms of function and structure, although they share the same genetic foundation. The immune system plays a crucial role in identifying and eliminating cells that deviate from the norm due to genetic mutations, such as deletions or insertions, showcasing the immune system's capacity to target and destroy aberrant cells.

This ability underscores the significant variance among individuals, primarily attributed to the diversity of the immune system. While most genes not directly related to immunity are remarkably similar across individuals, it is the unique characteristics of each person's immune response that primarily define their distinct biological identity.

This analysis explored the differences between immune cells from the circulating system versus immune cells from the tumor environment and what impact they might have in terms of functional assessment of one type of immune cell, T-cells, like in the figure 5.1, using ELISA, shown on the next page. The proposed workflow holds significant potential for advancing personalized cancer immunotherapy.

The results extracted with ELISA from five patients are discrepant and inconsistent.

This fact can be due to donor variability, and controls such as Phytohaemagglutinin (PHA), do not appear. Also the overall IFNγ production from the tumor associated antigens (TAA) and neoantigens after medium control, result in marginal concentration values, raging from 0.1pg/mL to 0.5pg/mL.

However, the script works, but would require different ELISA results for further validation to compare with the obtained results here.

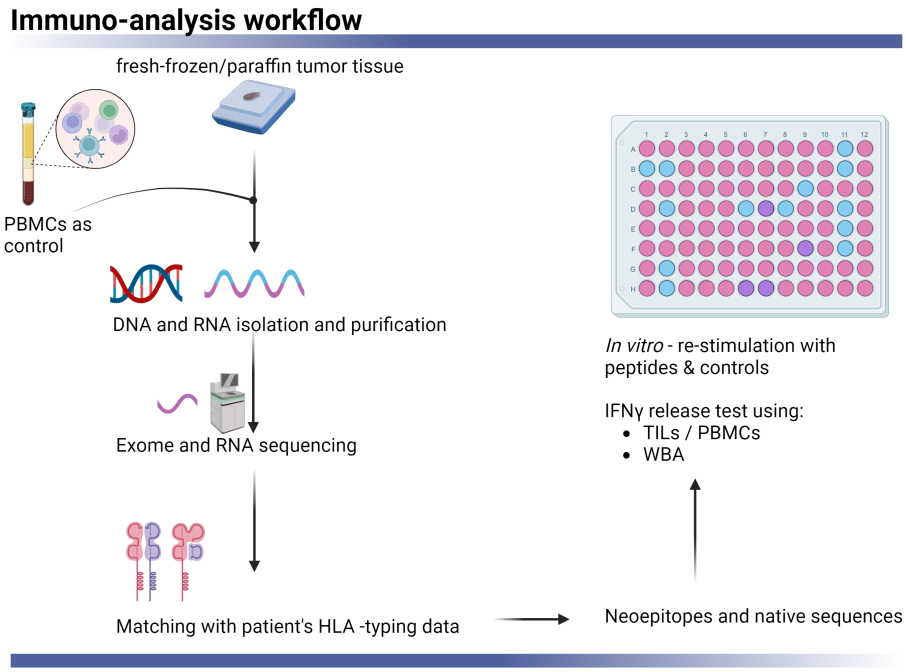


Figure 5.1: Immunoanalysis Workflow for Personalized Neoepitope Discovery and Immune Response Assay - designed in BioRender™[140]

The same discrepant results occur from Flow cytometry results extracted with flowjo.py script, this can be due to lack of optimization of the cytometry results since the program works as an extraction and graphing report.

Improvements should be developed in this package to control the gates which can be achieved with the used of gating ML, a machine learning tool that can optimize the gates from flow cytometry and therefore improving the obtained results.

However the scripts developed in statistical scope, gathering the required data extracted from elisa.py and flowjo.py work as expected and perform the required comparisons and correlations. Although the ELISA results affected negatively the linear regression or linear correlation performed with $linreg.py$, since the IFNγ production from co-culture with synthetic peptides is minimal.

Although a connection is observed among Survivin mix (1-33), Survivin (97-111), and Mesothelin MPF-11 Human antigens with different gated T-cells implying a possible T-cell effectiveness within this patient group when exposed to these peptides.

Moreover, the Th1 and Th2 comparison performed on 1_test_chi_squared.py have a great value as a T-cell product qualification assay before infusion, in the context of TIL therapy.

Another improvement to the pipeline would be the integration of fresh-frozen or paraffin-embedded tumor tissue analysis with PBMC as a control, to allow comprehensive identification of neoepitopes tailored to a patient's unique tumor profile.

The inclusion of DNA and RNA isolation in future, allows the sequencing, and matching with the patient's HLA typing, culminating in in vitro re-stimulation assays that measure IFNγ release.

This could result in a more targeted vaccine or therapy design, capable of eliciting a robust immune response by considering both shared and individual/private tumor antigen landscapes.

The exclusive ability to collect samples from patients with epithelial tumors easily undergoes decompensation, such as the results presented retrospectively, has made it possible to provide an objective window into the variations in the type of T-cells circulating in the tumor and the T-cells circulating in the patient's body.

In this way, immunomonitoring has once again demonstrated its fundamental role in the role in the follow-up of tumor patients. It enables a more personalised therapy in the long term, since in this particular case it allows the identification of recognition patterns on the part of the T-cells.

The dynamic interaction between cytometry and ELISA can further characterise the trajectory of the patient, requiring longitudinal measurements and not just single readings. The challenge is to integrate data from an epithelial tumor type population and continuous immunomonitoring at an individual level. Furthermore, real predictors with collinearity must be combined, including different physiological sensors and biomarkers.

5.0.1 Limitations

One of the major limitations of the current study is the exclusion of dendritic cells or immortalized B-cells for pulsing with synthetic peptides, which could potentially enhance the immunogenicity profile of the neoantigens being studied. However, the in silico pipeline established in this research is flexible and can be readily adapted to incorporate this additional layer in future experiments.

Another notable constraint is the small sample size, limited to only five samples, restricts the statistical power and the generalisation of the findings.

Additionally, the project encountered setbacks in finalizing the database and establishing proper linkages with the data_export.py script and MySQL, hindering the seamless integration and retrieval of data, which is essential for comprehensive analysis and interpretation of the results or for further development of longitudinal studies within immunomonitoring. A MySQL database diagram was developed but needs further improvement, as depicted in Figure 5.2.

These limitations outline areas for improvement and further research to bolster the robustness and applicability of the study's outcomes.

The study's methodology did not incorporate the use of sorted cells, relying solely on non-sorted cell populations. This decision may have a significant impact on the results, as sorted cells could provide a more refined and enriched population for analysis, potentially leading to different immunogenicity outcomes.

The absence of sorted cells may have limited the ability to discern subtle effects and specific immune responses attributable to particular cell types. This factor must be taken into consideration when interpreting the study's findings and could be addressed in future, to improve the accuracy and specificity of the immunological assays here developed.

To note that it was not considered yet the integration of immunohistochemistry and TCR sequencing with script modules was not considered yet into immunomonitoring clinical database in this project, although it could significantly enhance adoptive cell therapies for solid tumors.

The inclusion of TCR sequencing enables precise tracking and analysis of T-cell populations, aiding in the selection of tumor-specific T-cell receptors contributing to understanding the diversity and dynamics of T-cell repertoires over time, essential for optimizing T-cell therapies and predicting patient responses. It can lead to more effective adoptive cell therapies by ensuring that the T-cells used are most likely to recognize and target the cancer cells.

Also the integration of immunohistochemistry, allows to visualize the location and activation status of T-cells within tumor tissues, which provides insights into

the in vivo dynamics and functionality and effectiveness of the T-cell therapies.

This adds a detailed and functional understanding of therapeutic T-cells, leading to improved patient outcomes in solid tumor contexts.

Although not developed within this project but essential for transduced T-cells with targets with TCR or CAR constructs via virus, determining the viral copy number in genetically modified T-cells can be crucial for ensuring safety and efficacy.

5.0.2 Future Developments

There is room for further development and improvement.

Not least because some of the modules developed in the context of this thesis respond to the functionality of T-cells in a tumor environment.

However, in the context of genetically modified T-cells with a target via the TCR or CAR, there are necessary modules that have not been considered in this project, such as a viral copy number module if the genetic modification is carried out via the viral route. In addition, the proposed flow cytometry panels should include a detection in the case of TCR-T or CAR-T, such as G4S linker or LNGFR as examples of tag markers.[141]

In other words, the modules here developed, are specifically designed for the context of functional identification of T-cells from epithelial tumors, it has not been considered assays for genetically modified T-cells.

To this end, the first and most important step is to optimise some of the *in vitro* assays developed here, since the assays have unclear results in certain markers that would be important in the cytometry panels developed, since certain datasets do not contain information. Once all the parameters proposed here have been obtained, it could be important in the decision making process. It will be possible to develop a robust database (a proposed diagram, which needs improvement, is presented in figure 5.2 on the next page), with connection via the script under development, data_export.py that collects the parameters proposed.

Some of the modules developed are already prepared to receive data, such as ELISA data from WBA or TILs.

Here capable of suggesting more informed therapies and clinical decisions or the development of more specific and standardised clinical trials with the information collected.

In addition, it may be important to explore the removal of variables that may be considered without actually being considered, although this decision needs to be discussed within a immune cell manufacturing team.

Variables that may be considered without actually carrying much weight in the decision, such as variables related to the development of associated with the development of a particular cell therapy. In this way, the system can be optimised to make it simpler and more specific.

The model can be updated easily and quickly in the future. To do this all you need to do is correctly store all the cases you want to include in the suggested directory.

For ELISA assays, all you need is the Excel file with the correct structure,

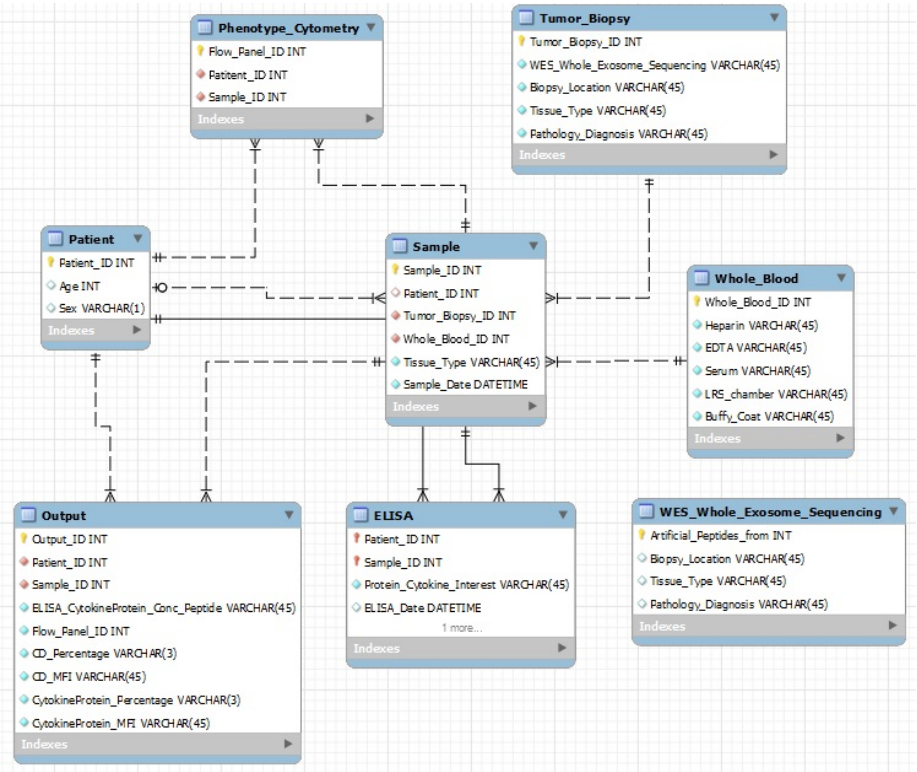


Figure 5.2: Proposed Database Diagram - designed in MySQL Workbench™

which can be used for ELISA assays that detect other cytokines as long as the data entry is correct and reliable.

For cytometry assays, simply place the fcs files obtained from the reading of the cell phenotype by the cytometry instrument in the directory and the gating strategy carried out in the commercial software FlowJo™, the specific workspace file for the fcs data and the percentages of cell markers and specific MFI are extracted. The set of modules is simply run to perform the analyses suggested here.

However, there is scope to introduce further scripts that may be required, for example, for the development of specific cell therapies:

- **Cell Counting module**

A critical aspect of immune cell therapy is administering the correct dose of immune cells to a patient. Cell counting ensures enough viable cells are manufactured for therapeutic efficacy.

During the manufacturing process of immune cells, counting should be consistent and repeatable. This data helps monitor process consistency and identify any deviations that could affect cell quality or yield, plus it ensures that the final product meets the pre-determined cell number and viability.

This way a script that can store the cell numbers obtained from the different timepoints of the manufacturing process would allow the overview of the immune cell growth overtime. Furthermore, it can ensure consistent counting protocols are followed across different batches and users.

- **Telomere analysis**

There is growing evidence that manipulating telomeres in TILs could potentially improve their fitness and therapeutic efficacy.[142]

Telomeres themselves, which are repetitive DNA sequences at the ends of chromosomes, are typically studied using DNA sequencing techniques. However, aspects of telomere regulation, such as the activity of telomerase (the enzyme responsible for maintaining telomeres), can also be explored using RNA sequencing (RNA-seq).[142, 143]

Research shows that TILs with longer telomeres tend to persist longer in patients undergoing adoptive cell transfer therapy, leading to better tumor regression outcomes. Cells with short telomeres are prone to senescence after transfer, while those with longer telomeres can continue to mediate antitumor effects.[142]

The enzyme telomerase, which extends telomeres, can enhance the proliferation of TILs. Overexpression of telomerase components such as hTERT can improve TIL fitness, making them more suitable for adoptive immunotherapy by boosting their proliferative capacity.[144]

Since telomere length affects the persistence and effectiveness of TILs, strategies targeting telomere maintenance in TILs could improve the effectiveness of adoptive immunotherapy. Therefore, an inclusion of such a module in ACT on TILs would be a good support.

- **TCR analysis**

TCR analysis will allow to confirm this hypothesis of which drugs modelates a certain phenotype and determine whether the obtained cell product corresponds to a monoclonal or polyclonal T-cell population. Since T-cells from some patients recognized the autologous tumor cells, would also be interesting to determine the Vα, Vβ, and Vγ, Vδ repertoires present in the PBMCs to ascertain whether tumor-specific T-cells were only located at the tumor site or also in circulation. This can be done using deep TCR sequencing.

- **Viral copy number**

CAR-T or TCR-T products that are virally transduced require validation of the viral copy number prior to infusion of the cell therapy product. This requires verification that the viral copy number is below 5 according to Food and Drug Administration (FDA) standards.[145]

Essentially, the number of viral copies quantified by Polymerase Chain Reaction (PCR) analysis and the percentage of cells transduced with CAR or TCR by cytometry analysis are compared between different groups of samples.

In this context, an improvement on script flowjo.py would be required to extract the percentage of CAR or TCR, and a module for PCR analysis would need to be developed in Python and incorporated into this pipeline.

- **Seahorse XF data analysis**

A module to analyze and extract data from Seahorse data, would allow to reflect immune cell oxygen consumption and therefore metabolic process or to indicate the rate at which cells produce lactic acid, a by product of glycolysis.

This would allow for optimization of immune cell products during manufacturing to track changes in metabolic activity, comparing this profile with different immune cells in different differentiation stages or treatment conditions on manufacturing optimization.

Which allows the identification of disruptions in immune cell metabolism and response to stimuli, so this is a valuable result for manufacturing.

- **Digital pathology module to evaluate histochemistry combined with spatial transcriptomics**

Such a script model can offer valuable insights into the tumor immune landscape in epithelial cancers, particularly regarding tumor-infiltrating lymphocytes (TILs) and chimeric antigen receptor (CAR) or T-cell receptor (TCR) products. Information on the tissue architecture and distribution of various cell types, including immune cells which can be used to identify and quantify TILs by staining for specific immune cell markers.

Then the digital pathology with high-resolution image analysis of tissue sections, allowing for more accurate and objective quantification.

Furthermore the use of spatial transcriptomics would reveal spatial localization of gene expression within tissue section. It would gain a deeper understanding of the immune response to epithelial cancers. This information can be used to develop more targeted immunotherapies

- **Integration site analysis**

In the context of viraly transduced T-cells, like CAR-T and TCR-T, it is wise to collect information about the integration site. Since Biasco et al were able to demonstrate through the analysis on retroviral integration sites that adoptively transferred $CD8^+$ T memory stem cells (Tscm), the earliest differentiated circulating memory T-cell population possessing superior stem cell–like qualities identified thus far, preferentially survived in vivo compared with more differentiated central memory (TCM) or effector memory T (TEM) cells in patients treated with genetically modified lymphocytes.[146]

A model to analyze in the context of patients treated with adoptive cell therapy would be essential to understand which differentiated genetically transduced T-cell within the immune cell product retains a tumor effect and therefore improve further ACT developments within immune cell products.

This tool would be applied in immune-monitoring of patients treated with ACT.

Besides the *in silico* improvements that can be explored and added to the pipeline, there are further improvements to be performed in the *in vivo* setting, like:

- **IFNγ production from co-culture assays with synthetic peptides**

It would be necessary to repeat these assays with a higher sample size to evaluate if the synthetic peptides used were active or not since the calibration curve was well calibrated but the IFNγ was marginal and should reports higher concentration values in theory.

- **T-homing flow cytometry panel**

Such improvements, are the flow cytometry panel on homing markers, since the interaction between chemokine receptors and their ligands is a critical factor for the proper homing of immune cells into tumor sites, a point that has been one of the main challenges of the ACT-based cancer immunotherapies for solid tumors. It is still necessary to perform some chemotaxis assays that confirm the migration capacity of the expanded tumor-infiltrating cells into the tumor. To complement these in vitro and in vivo assays, the chemokines released by the autologous tumor could be determined through an ELISA, allowing to further confirm through RNA-sequencing or flow cytometry whether the respective chemokine receptors are expressed by the expanded T-cell product.

- **Extract more information from T-cell flow cytometry panels**

The improvement of these ACT as single therapies will allow their future combination with $\alpha\beta$ T-cells in a unique cell-based cancer immunotherapy, which will offer the simultaneously targeting of MHC positive and negative tumor cells, resulting in tumor eradication.

Thus, it would be interesting to first demonstrate in vitro the synergistic action of non alpha beta T-cells and $\alpha\beta$ T-cells.

Besides evaluating the IFNγ production, IL-17 production, CD107a expression and tumor cell death could also be determined. Moreover, it would also be interesting to observe the live interactions between tumor cells and the different immune cell types, individually or combined, since $\gamma\delta$ T-cells have been suggested to take up killed tumor cells and to cross-present them to $\alpha\beta$ T-cells.

5.0.3 Final Considerations

Developing a personalised neoepitope vaccine or adoptive cell therapies off-the-shelf and tailored patient-specific combinations may seem an advanced concept.

However, with rapid technological and methodological advances leading to reduced costs and production time, it is not unlikely that this could become a standard of care for cancer patients in the future. Within a tumor sample, there can be a variety of cell types and states. Cancer is not homogenous, different parts of a tumor can behave differently, which complicates treatment. The adaptability of the immune system, particularly TCR diversity, which is crucial for recognizing a wide array of antigens presented by tumor cells. For solid or epithelial tumors the microbiome, microbes in a particular environment, is noted, with ”cross-reactivity” suggesting that the immune system might react to microbial antigens in ways that influence cancer progression or treatment response. Also the timing and location of these molecular events can have significant implications. The genomic and immunological complexity of cancer, implies different levels of biological information that still need to be elicited.

The complexity of neoantigen diversity in cancer immunotherapy still underscores the importance of meticulously storing patient data from immunological assays due to the overwhelmed meticulous experiments that require full attention of technicians and researchers. In the context of personalized medicine a strong patient data storage, especially in oncology, accounts as extremely important information.

Some patients may exhibit shared/common driver mutations and or private patient-specific mutations it will be crucial in the long run and to keep the drive of the ACT wave that still relies majorly on common shared tumor antigens.

Targeting the Tumor Mutational Burden (TMB) for cancer immunotherapy involves a range of strategies, from neoantigen-directed antibodies to CAR T-cells, and from shared neoantigen vaccines to TCR-transgenic T-cells and corrective gene therapy.

Each approach relies on the unique immunological profile of the tumor and the patient’s immune response, as it can be seen in figure 5.3, on the next page, varying from the presence of shared or private mutations at the epithelial tumor.

This highlights the need for a detailed and accurate information management system that can handle the diversity and complexity of this information.

Correct data storage and management of such patient-specific information enable the tailoring of treatment strategies that can effectively mobilize the body’s immune system. High TCR diversity in T-cells extracted from TIL and peripheral

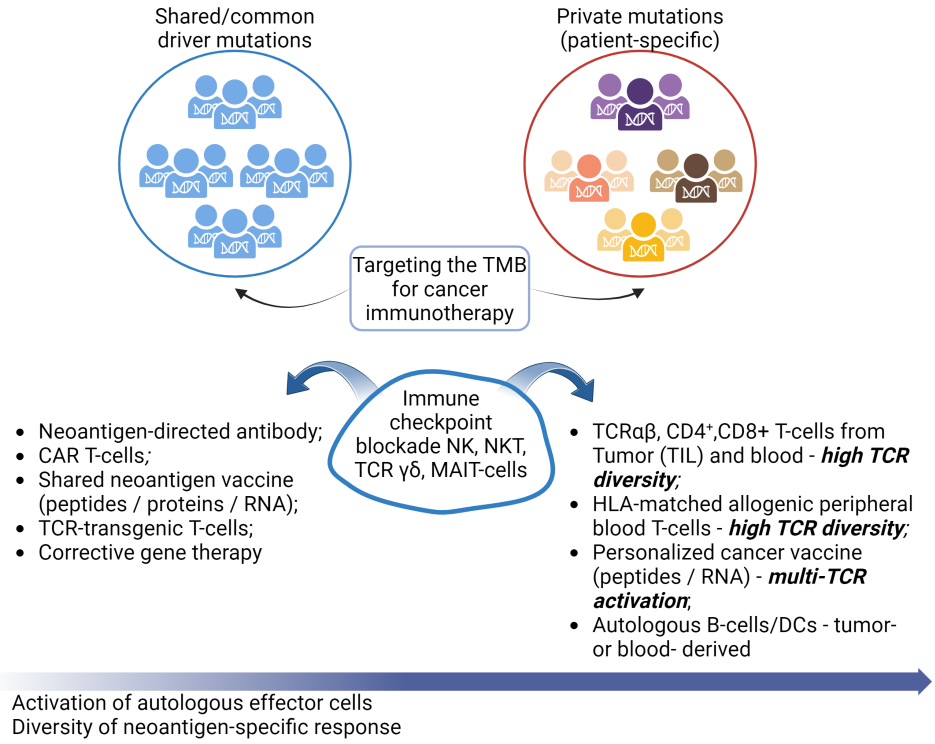


Figure 5.3: Shared and Private mutations, how to target them? - designed in BioRender™[140]

blood, along with the need for HLA-matched allogenic T-cells and personalized cancer vaccines, demand robust data systems capable of handling multifaceted and dynamic data sets.

Autologous effector cells, including B-cells and Dendritic Cells (DCs), derived from tumors or blood, further add layers to the data complexity.

Thus, reliable data storage systems are fundamental to ensuring that the individualized treatment plans are informed by accurate and comprehensive immunological profiles, acting as a decision support system capable of paving the way for more effective and responsive cancer immunotherapies.

Bibliography

[1] J. M. Lourenço. *The NOVAthesis LaTeX Template User's Manual*. NOVA University Lisbon. 2021. URL: https://github.com/joaomlourenco/novathesis/raw/main/template.pdf (cit. on p. i).

[2] *Cancer*. en. https://www.who.int/health-topics/cancer. Accessed: 2024-4-11 (cit. on p. 1).

[3] *Cancer*. en. https://www.who.int/news-room/fact-sheets/detail/cancer. Accessed: 2024-4-12 (cit. on p. 1).

[4] A. M. Dickinson, D. Caufield, and J. Campbell. "Cell Therapy Report on Current Guidelines Regarding Advanced Cell Therapy Pre-clinical Safety, Efficacy and Potency Testing". In: () (cit. on p. 1).

[5] *Guidelines relevant for advanced therapy medicinal products*. en. https://www.ema.europa.eu/en/human-regulatory-overview/advanced-therapy-medicinal-products-overview/guidelines-relevant-advanced-therapy-medicinal-products. Accessed: 2024-4-14 (cit. on p. 2).

[6] U. Aickelin, D. Dasgupta, and F. Gu. "Artificial immune systems". In: *Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques*. Springer, 2013, pp. 187–211 (cit. on p. 7).

[7] J. M. den Haan, R. Arens, and M. C. van Zelm. "The activation of the adaptive immune system: cross-talk between antigen-presenting cells, T cells and B cells". In: *Immunology letters* 162.2 (2014), pp. 103–112 (cit. on p. 7).

[8] H. Modjtahedi and A. Clarke. "The immune system". In: *The Biology of Cancer* (2007), pp. 79–98 (cit. on p. 8).

[9] C. Chou and M. O. Li. “Tissue-resident lymphocytes across innate and adaptive lineages”. In: *Frontiers in Immunology* 9 (2018), p. 399246 (cit. on p. 8).

[10] S. Sattler. “The role of the immune system beyond the fight against infection”. In: *The immunology of cardiovascular homeostasis and pathology* (2017), pp. 3–14 (cit. on p. 8).

[11] G. Gasteiger and A. Y. Rudensky. “Interactions between innate and adaptive lymphocytes”. In: *Nature Reviews Immunology* 14.9 (2014), pp. 631–639 (cit. on p. 8).

[12] L. C. Rankin and D. Artis. “Beyond host defense: emerging functions of the immune system in regulating complex tissue physiology”. In: *Cell* 173.3 (2018), pp. 554–567 (cit. on p. 8).

[13] K. Guo and S. Ma. “The immune system in transfusion-related acute lung injury prevention and therapy: update and perspective”. In: *Frontiers in Molecular Biosciences* 8 (2021), p. 639976 (cit. on p. 9).

[14] A. Serrano-del Valle et al. “Immunogenic cell death and immunotherapy of multiple myeloma”. In: *Frontiers in Cell and Developmental Biology* 7 (2019), p. 50 (cit. on p. 9).

[15] W. J. Mulder et al. “Therapeutic targeting of trained immunity”. In: *Nature reviews Drug discovery* 18.7 (2019), pp. 553–566 (cit. on p. 9).

[16] M. Strzelec et al. “Immunomodulation—a general review of the current state-of-the-art and new therapeutic strategies for targeting the immune system”. In: *Frontiers in Immunology* 14 (2023), p. 1127704 (cit. on p. 9).

[17] D. Zheng, T. Liwinski, and E. Elinav. “Interaction between microbiota and immunity in health and disease”. In: *Cell research* 30.6 (2020), pp. 492–506 (cit. on p. 9).

[18] M. G. Netea et al. “Innate and adaptive immune memory: an evolutionary continuum in the host’s response to pathogens”. In: *Cell host & microbe* 25.1 (2019), pp. 13–26 (cit. on p. 9).

[19] L. E. Harrington. “T-cell development”. In: *Clinical Immunology*. Elsevier, 2019, pp. 119–125 (cit. on pp. 9–11).

[20] K. M. Ashby and K. A. Hogquist. “A guide to thymic selection of T cells”. In: *Nature Reviews Immunology* 24.2 (2024), pp. 103–117 (cit. on p. 9).

[21] H. Takaba and H. Takayanagi. “The mechanisms of T cell selection in the thymus”. In: *Trends in immunology* 38.11 (2017), pp. 805–816 (cit. on pp. 10, 11).

[22] P. Romero et al. “Four functionally distinct populations of human effector-memory CD8+ T lymphocytes”. In: *The Journal of Immunology* 178.7 (2007), pp. 4112–4119 (cit. on p. 11).

[23] J. J. Campbell and E. C. Butcher. “Chemokines in tissue-specific and microenvironment specific lymphocyte homing”. In: *Current opinion in immunology* 12.3 (2000), pp. 336–341 (cit. on p. 11).

[24] J. J. Campbell et al. “CCR7 expression and memory T cell diversity in humans”. In: *The Journal of Immunology* 166.2 (2001), pp. 877–884 (cit. on p. 11).

[25] L. Gattinoni et al. “T memory stem cells in health and disease”. en. In: *Nat. Med.* 23.1 (2017-01), pp. 18–27 (cit. on pp. 11, 12, 14, 24, 25).

[26] D. Focosi et al. “CD57+ T lymphocytes and functional immune deficiency”. In: *Journal of leukocyte biology* 87.1 (2010), pp. 107–116 (cit. on p. 11).

[27] M. E. Peter and P. Krammer. “The CD95 (APO-1/Fas) DISC and beyond”. In: *Cell Death & Differentiation* 10.1 (2003), pp. 26–35 (cit. on p. 11).

[28] V. Golubovskaya and L. Wu. “Different subsets of T cells, memory, effector functions, and CAR-T immunotherapy”. In: *Cancers* 8.3 (2016), p. 36 (cit. on p. 12).

[29] G. Challis and H. Stam. “The spontaneous regression of cancer: a review of cases from 1900 to 1987”. In: *Acta oncologica* 29.5 (1990), pp. 545–550 (cit. on p. 13).

[30] S. Euvrard, J. Kanitakis, and A. Claudy. “Skin cancers after organ transplantation”. In: *New England Journal of Medicine* 348.17 (2003), pp. 1681–1691 (cit. on p. 13).

[31] S. Hoption Cann, J. Van Netten, and C. Van Netten. “Dr William Coley and tumour regression: a place in history or in the future”. In: *Postgraduate medical journal* 79.938 (2003), pp. 672–680 (cit. on p. 13).

[32] F. Azimi et al. “Tumor-infiltrating lymphocyte grade is an independent predictor of sentinel lymph node status and survival in patients with cutaneous melanoma”. In: *Journal of clinical oncology* 30.21 (2012), pp. 2678–2683 (cit. on pp. 13, 21).

[33] L. Scheetz et al. “Engineering patient-specific cancer immunotherapies”. In: *Nature biomedical engineering* 3.10 (2019), pp. 768–782 (cit. on p. 13).

[34] D. Hanahan and R. A. Weinberg. “The hallmarks of cancer”. In: *cell* 100.1 (2000), pp. 57–70 (cit. on pp. 13, 14).

[35] F. Cavallo et al. “2011: the immune hallmarks of cancer”. In: *Cancer Immunology, Immunotherapy* 60 (2011), pp. 319–326 (cit. on p. 14).

[36] D. Hanahan and R. A. Weinberg. “Hallmarks of cancer: the next generation”. In: *cell* 144.5 (2011), pp. 646–674 (cit. on pp. 14, 15).

[37] M. Philip and A. Schietinger. “CD8+ T cell differentiation and dysfunction in cancer”. In: *Nature Reviews Immunology* 22.4 (2022), pp. 209–223 (cit. on p. 15).

[38] M. Schäfer and S. Werner. “Cancer as an overhealing wound: an old hypothesis revisited”. In: *Nature reviews Molecular cell biology* 9.8 (2008), pp. 628–638 (cit. on p. 15).

[39] M. Burnet. “Cancer—a biological approach: I. The processes of control. II. The significance of somatic mutation”. In: *British medical journal* 1.5022 (1957), p. 779 (cit. on p. 15).

[40] A. N. Houghton. “A Perspective on Cancer Immunology and Immunotherapy”. In: *General Principles of Tumor Immunotherapy: Basic and Clinical Applications of Tumor Immunology*. Springer, 2007, pp. 3–15 (cit. on p. 15).

[41] V. Shankaran et al. “IFNγ and lymphocytes prevent primary tumour development and shape tumour immunogenicity”. In: *Nature* 410.6832 (2001), pp. 1107–1111 (cit. on p. 16).

[42] G. P. Dunn et al. “Cancer immunoediting: from immunosurveillance to tumor escape”. In: *Nature immunology* 3.11 (2002), pp. 991–998 (cit. on p. 16).

[43] S. H. van der Burg et al. “Vaccines for established cancer: overcoming the challenges posed by immune evasion”. In: *Nature Reviews Cancer* 16.4 (2016), pp. 219–233 (cit. on p. 16).

[44] O. Hamid et al. “Safety and tumor responses with lambrolizumab (anti–PD-1) in melanoma”. In: *New England Journal of Medicine* 369.2 (2013), pp. 134–144 (cit. on p. 17).

[45] F. S. Hodi et al. “Improved survival with ipilimumab in patients with metastatic melanoma”. In: *New England Journal of Medicine* 363.8 (2010), pp. 711–723 (cit. on p. 17).

[46] S. A. Rosenberg and N. P. Restifo. “Adoptive cell transfer as personalized immunotherapy for human cancer”. In: *Science* 348.6230 (2015), pp. 62–68 (cit. on pp. 17, 19, 21).

[47] V. Bronte and S. Mocellin. “Suppressive influences in the immune response to cancer”. In: *Journal of immunotherapy* 32.1 (2009), pp. 1–11 (cit. on p. 17).

[48] N. McGranahan and C. Swanton. “Clonal heterogeneity and tumor evolution: past, present, and the future”. In: *Cell* 168.4 (2017), pp. 613–628 (cit. on p. 17).

[49] B. Seliger, M. J. Maeurer, and S. Ferrone. “Antigen-processing machinery breakdown and tumor growth”. In: *Immunology today* 21.9 (2000), pp. 455–464 (cit. on p. 17).

[50] P. Sharma et al. “Primary, adaptive, and acquired resistance to cancer immunotherapy”. In: *Cell* 168.4 (2017), pp. 707–723 (cit. on pp. 17, 21).

[51] P. Sinha et al. “Tumor immunity: a balancing act between T cell activation, macrophage activation and tumor-induced immune suppression”. In: *Cancer Immunology, Immunotherapy* 54 (2005), pp. 1137–1142 (cit. on p. 17).

[52] J. Tolosa et al. “The endogenous retroviral envelope protein syncytin-1 inhibits LPS/PHA-stimulated cytokine responses in human blood and is sorted into placental exosomes”. In: *Placenta* 33.11 (2012), pp. 933–941 (cit. on p. 17).

[53] J. Yuan et al. “Correlation of clinical and immunological data in a metastatic melanoma patient with heterogeneous tumor responses to ipilimumab therapy”. en. In: *Cancer Immun.* 10 (2010-01), p. 1 (cit. on p. 17).

[54] N. Tanaka et al. “Whole-tissue biopsy phenotyping of three-dimensional tumours reveals patterns of cancer heterogeneity”. In: *Nature Biomedical Engineering* 1.10 (2017), pp. 796–806 (cit. on pp. 17, 18).

[55] M. Jamal-Hanjani et al. “Translational implications of tumor heterogeneity”. In: *Clinical cancer research* 21.6 (2015), pp. 1258–1266 (cit. on p. 18).

[56] M. H. Andersen et al. “Cytotoxic T cells”. In: *Journal of Investigative Dermatology* 126.1 (2006), pp. 32–41 (cit. on p. 18).

[57] S. Kreiter et al. “Mutant MHC class II epitopes drive therapeutic immune responses to cancer”. In: *Nature* 520.7549 (2015), pp. 692–696 (cit. on p. 18).

[58] C. Linnemann et al. “High-throughput epitope discovery reveals frequent recognition of neo-antigens by CD4+ T cells in human melanoma”. In: *Nature medicine* 21.1 (2015), pp. 81–85 (cit. on p. 18).

[59] E. Tran et al. “Cancer immunotherapy based on mutation-specific CD4+ T cells in a patient with epithelial cancer”. In: *Science* 344.6184 (2014), pp. 641–645 (cit. on p. 18).

[60] Y. Zhao, C. Niu, and J. Cui. “Gamma-delta ($\gamma\delta$) T cells: friend or foe in cancer development?” In: *Journal of translational medicine* 16 (2018), pp. 1–13 (cit. on p. 19).

[61] O. Melaiu et al. “Influence of the tumor microenvironment on NK cell function in solid tumors”. In: *Frontiers in immunology* 10 (2020), p. 3038 (cit. on p. 19).

[62] M. H. Kershaw, J. A. Westwood, and P. K. Darcy. “Gene-engineered T cells for cancer therapy”. In: *Nature Reviews Cancer* 13.8 (2013), pp. 525–541 (cit. on pp. 19, 21).

[63] A. Fukunaga et al. “CD8+ tumor-infiltrating lymphocytes together with CD4+ tumor-infiltrating lymphocytes and dendritic cells improve the prognosis of patients with pancreatic adenocarcinoma”. en. In: *Pancreas* 28.1 (2004-01), e26–31 (cit. on p. 19).

[64] J. Haanen et al. “Melanoma-specific tumor-infiltrating lymphocytes but not circulating melanoma-specific T cells may predict survival in resected advanced stage melanoma patients”. In: *Cancer immunology, immunotherapy* 55 (2006), pp. 451–458 (cit. on p. 19).

[65] N. R. West et al. “Tumor-infiltrating lymphocytes predict response to anthra cycline-based chemotherapy in estrogen receptor-negative breast cancer”. In: *Breast cancer research* 13 (2011), pp. 1–13 (cit. on p. 19).

[66] Y. Geng et al. *Prognostic role of tumor-infiltrating lymphocytes in lung cancer: a meta-analysis*. 2015 (cit. on p. 19).

[67] K. Itoh, C. D. Platsoucas, and C. M. Balch. “Autologous tumor-specific cytotoxic T lymphocytes in the infiltrate of human metastatic melanomas. Activation by interleukin 2 and autologous tumor cells, and involvement of

the T cell receptor." In: *The Journal of experimental medicine* 168.4 (1988), pp. 1419–1441 (cit. on p. 20).

[68] I. Yron et al. "In vitro growth of murine T cells. V. The isolation and growth of lymphoid cells infiltrating syngeneic solid tumors." In: *Journal of immunology (Baltimore, Md.: 1950)* 125.1 (1980), pp. 238–245 (cit. on p. 20).

[69] S. A. Rosenberg, P. Spiess, and R. Lafreniere. "A new approach to the adoptive immunotherapy of cancer with tumor-infiltrating lymphocytes". In: *Science* 233.4770 (1986), pp. 1318–1321 (cit. on p. 20).

[70] P. J. Spiess, J. C. Yang, and S. A. Rosenberg. "In vivo antitumor activity of tumor-infiltrating lymphocytes expanded in recombinant interleukin-2". In: *Journal of the National Cancer Institute* 79.5 (1987), pp. 1067–1075 (cit. on p. 20).

[71] S. A. Rosenberg et al. "Use of tumor-infiltrating lymphocytes and interleukin-2 in the immunotherapy of patients with metastatic melanoma". In: *New England Journal of Medicine* 319.25 (1988), pp. 1676–1680 (cit. on p. 20).

[72] V. A. Fernández et al. "Biomarkers for response to TIL therapy: a comprehensive review". In: *Journal for Immunotherapy of Cancer* 12.3 (2024) (cit. on pp. 20, 28).

[73] J. Weber et al. "White paper on adoptive cell therapy for cancer with tumor-infiltrating lymphocytes: a report of the CTEP subcommittee on adoptive cell therapy". In: *Clinical Cancer Research* 17.7 (2011), pp. 1664–1673 (cit. on p. 21).

[74] J. Galon et al. "Type, density, and location of immune cells within human colorectal tumors predict clinical outcome". In: *Science* 313.5795 (2006), pp. 1960–1964 (cit. on p. 21).

[75] S. M. Mahmoud et al. "Tumor-infiltrating CD8+ lymphocytes predict clinical outcome in breast cancer". In: *Journal of clinical oncology* 29.15 (2011), pp. 1949–1955 (cit. on p. 21).

[76] F. Pagès et al. "Effector memory T cells, early metastasis, and survival in colorectal cancer". In: *New England journal of medicine* 353.25 (2005), pp. 2654–2666 (cit. on p. 21).

[77] L. B. Alexandrov et al. "Signatures of mutational processes in human cancer". In: *Nature* 500.7463 (2013), pp. 415–421 (cit. on pp. 21, 32).

[78] L. Gattinoni et al. “Removal of homeostatic cytokine sinks by lymphodepletion enhances the efficacy of adoptively transferred tumor-specific CD8+ T cells”. In: *The Journal of experimental medicine* 202.7 (2005), pp. 907–912 (cit. on p. 21).

[79] Y. Naito et al. “CD8+ T cells infiltrated within cancer cell nests as a prognostic factor in human colorectal cancer”. In: *Cancer research* 58.16 (1998), pp. 3491–3494 (cit. on p. 21).

[80] E. Borcoman et al. “Patterns of response and progression to immunotherapy”. In: *American Society of Clinical Oncology Educational Book* 38 (2018), pp. 169–178 (cit. on p. 21).

[81] M. Sadelain, R. Brentjens, and I. Rivière. “The basic principles of chimeric antigen receptor design”. In: *Cancer discovery* 3.4 (2013), pp. 388–398 (cit. on pp. 21, 22).

[82] K. Hirabayashi et al. “Dual-targeting CAR-T cells with optimal co-stimulation and metabolic fitness enhance antitumor activity and prevent escape in solid tumors”. In: *Nature cancer* 2.9 (2021), pp. 904–918 (cit. on p. 22).

[83] P. Peters. “T cell receptor modification by alpha and beta chain domain cross-over for adoptive T cell therapy”. PhD thesis. lmu, 2018 (cit. on p. 22).

[84] F. Manfredi et al. “TCR redirected T cells for cancer treatment: achievements, hurdles, and goals”. In: *Frontiers in immunology* 11 (2020), p. 1689 (cit. on p. 22).

[85] R. Xu et al. “Neoantigen-targeted TCR-T cell therapy for solid tumors: how far from clinical application”. In: *Cancer Letters* 546 (2022), p. 215840 (cit. on pp. 22, 23).

[86] T. Karpanen and J. Olweus. “The potential of donor T-cell repertoires in neoantigen-targeted cancer immunotherapy”. In: *Frontiers in immunology* 8 (2017), p. 309747 (cit. on p. 23).

[87] A. E. Zamora, J. C. Crawford, and P. G. Thomas. “Hitting the target: how T cells detect and eliminate tumors”. In: *The Journal of Immunology* 200.2 (2018), pp. 392–399 (cit. on p. 23).

[88] C. A. Klebanoff et al. “T cell receptor therapeutics: immunological targeting of the intracellular cancer proteome”. In: *Nature Reviews Drug Discovery* 22.12 (2023), pp. 996–1017 (cit. on p. 23).

[89] A. K. Bentzen and S. R. Hadrup. “Evolution of MHC-based technologies used for detection of antigen-responsive T cells”. In: *Cancer immunology, immunotherapy* 66 (2017), pp. 657–666 (cit. on pp. 26, 30).

[90] J. Robinson et al. “IMGT/HLA and IMGT/MHC: sequence databases for the study of the major histocompatibility complex”. In: *Nucleic acids research* 31.1 (2003), pp. 311–314 (cit. on p. 26).

[91] M. Niemann et al. “Peptides derived from mismatched paternal human leukocyte antigen predicted to be presented by HLA-DRB1,-DRB3/4/5,-DQ, and-DP induce child-specific antibodies in pregnant women”. In: *Frontiers in Immunology* 12 (2021), p. 797360 (cit. on p. 26).

[92] N. McGranahan et al. “Allele-specific HLA loss and immune escape in lung cancer evolution”. In: *Cell* 171.6 (2017), pp. 1259–1271 (cit. on p. 26).

[93] A. K. Sewell. “Why must T cells be cross-reactive?” In: *Nature Reviews Immunology* 12.9 (2012), pp. 669–677 (cit. on p. 27).

[94] M. Cobbold et al. “MHC class I–associated phosphopeptides are the targets of memory-like immunity in leukemia”. In: *Science translational medicine* 5.203 (2013), 203ra125–203ra125 (cit. on p. 27).

[95] T. A. Groothuis et al. “MHC class I alleles and their exploration of the antigen-processing machinery”. In: *Immunological reviews* 207.1 (2005), pp. 60–76 (cit. on p. 27).

[96] L. Sibilio et al. “A single bottleneck in HLA-C assembly”. In: *Journal of biological chemistry* 283.3 (2008), pp. 1267–1274 (cit. on p. 27).

[97] T. P. Arstila et al. “A direct estimate of the human $\alpha\beta$ T cell receptor diversity”. In: *Science* 286.5441 (1999), pp. 958–961 (cit. on p. 28).

[98] D. Mason. “A very high level of crossreactivity is an essential feature of the T-cell receptor”. In: *Immunology today* 19.9 (1998), pp. 395–404 (cit. on p. 28).

[99] R. A. Montgomery et al. “HLA in transplantation”. In: *Nature reviews nephrology* 14.9 (2018), pp. 558–570 (cit. on p. 29).

[100] S. E. Hamilton et al. “The generation of protective memory-like CD8+ T cells during homeostatic proliferation requires CD4+ T cells”. In: *Nature immunology* 7.5 (2006), pp. 475–481 (cit. on p. 29).

[101] T. Gebhardt et al. “Different patterns of peripheral migration by memory CD4+ and CD8+ T cells”. In: *Nature* 477.7363 (2011), pp. 216–219 (cit. on p. 30).

[102] S. Sakaguchi et al. “Regulatory T cells and human disease”. In: *Annual review of immunology* 38 (2020), pp. 541–566 (cit. on p. 30).

[103] X. Yao et al. “Levels of peripheral CD4+ FoxP3+ regulatory T cells are negatively associated with clinical response to adoptive immunotherapy of human cancer”. In: *Blood, The Journal of the American Society of Hematology* 119.24 (2012), pp. 5688–5696 (cit. on p. 30).

[104] P. van der Bruggen et al. “A gene encoding an antigen recognized by cytolytic T lymphocytes on a human melanoma”. In: *Science* 254.5038 (1991), pp. 1643–1647 (cit. on p. 31).

[105] L. R. Olsen et al. “TANTIGEN: a comprehensive database of tumor T cell antigens”. In: *Cancer Immunology, Immunotherapy* 66 (2017), pp. 731–735 (cit. on p. 31).

[106] C. Yee et al. “Melanocyte destruction after antigen-specific immunotherapy of melanoma: direct evidence of T cell–mediated vitiligo”. In: *The Journal of experimental medicine* 192.11 (2000), pp. 1637–1644 (cit. on p. 31).

[107] B. Perdiguero et al. “Deletion of the viral anti-apoptotic gene F1L in the HIV/AIDS vaccine candidate MVA-C enhances immune responses against HIV-1 antigens”. In: *PloS one* 7.10 (2012), e48524 (cit. on p. 31).

[108] P. Hombrink et al. “Mixed functional characteristics correlating with TCR-ligand koff-rate of MHC-tetramer reactive T cells within the naive T-cell repertoire”. In: *European journal of immunology* 43.11 (2013), pp. 3038–3050 (cit. on p. 32).

[109] M. Efremova et al. “Neoantigens generated by individual mutations and their role in cancer immunity and immunotherapy”. In: *Frontiers in immunology* 8 (2017), p. 311621 (cit. on pp. 33, 34).

[110] E. F. Fritsch et al. “HLA-binding properties of tumor neoepitopes in humans”. In: *Cancer immunology research* 2.6 (2014), pp. 522–529 (cit. on pp. 33, 34).

[111] M. Vormehr. “Mutated neo-antigens as targets for individualized cancer immunotherapy”. PhD thesis. Mainz, Univ., Diss., 2016, 2016 (cit. on pp. 34, 35).

[112] N. Riaz et al. “The role of neoantigens in response to immune checkpoint blockade”. In: *International immunology* 28.8 (2016), pp. 411–419 (cit. on pp. 35, 36).

[113] M. Bassani-Sternberg and G. Coukos. “Mass spectrometry-based antigen discovery for cancer immunotherapy”. In: *Current opinion in immunology* 41 (2016), pp. 9–17 (cit. on p. 35).

[114] N. Jacquelot et al. “Chemokine receptor patterns in lymphocytes mirror metastatic spreading in melanoma”. In: *The Journal of clinical investigation* 126.3 (2016), pp. 921–937 (cit. on p. 37).

[115] K. Hung et al. “The central role of CD4+ T cells in the antitumor immune response”. In: *The Journal of experimental medicine* 188.12 (1998), pp. 2357–2368 (cit. on p. 37).

[116] A. Lanzavecchia and F. Sallusto. “Dynamics of T lymphocyte responses: intermediates, effectors, and memory cells”. In: *Science* 290.5489 (2000), pp. 92–97 (cit. on p. 37).

[117] F. Benchetrit et al. “Interleukin-17 inhibits tumor cell growth by means of a T-cell–dependent mechanism”. In: *Blood, The Journal of the American Society of Hematology* 99.6 (2002), pp. 2114–2121 (cit. on p. 37).

[118] H. Liang et al. “Interleukin-17 Facilitates the Immune Suppressor Capacity of High-Grade G lioma-Derived CD 4 (+) CD 25 (+) F oxp3 (+) T Cells Via Releasing Transforming Growth Factor Beta”. In: *Scandinavian journal of immunology* 80.2 (2014), pp. 144–150 (cit. on p. 37).

[119] S. Punt et al. “The correlations between IL-17 vs. Th17 cells and cancer patient survival: a systematic review”. In: *Oncoimmunology* 4.2 (2015), e984547 (cit. on p. 37).

[120] S. Becattini et al. “Functional heterogeneity of human memory CD4+ T cell clones primed by pathogens or vaccines”. In: *Science* 347.6220 (2015), pp. 400–406 (cit. on p. 37).

[121] A. Morrot. “Lifelong protection mediated by stem cell-like CD8+ T memory subset cells (Tscm) induced by vaccination”. In: *Annals of Translational Medicine* 4.11 (2016) (cit. on p. 37).

[122] R. Barreira da Silva et al. “Dipeptidylpeptidase 4 inhibition enhances lymphocyte trafficking, improving both naturally occurring tumor immunity and immunotherapy”. In: *Nature immunology* 16.8 (2015), pp. 850–858 (cit. on p. 38).

[123] M. E. Mikucki et al. “Unlocking tumor vascular barriers with CXCR3: implications for cancer immunotherapy”. In: *Oncoimmunology* 5.5 (2016), e1116675 (cit. on p. 38).

[124] J. Zhou et al. “Prognostic and therapeutic value of CD103+ cells in renal cell carcinoma”. In: *Experimental and therapeutic medicine* 15.6 (2018), pp. 4979–4986 (cit. on p. 39).

[125] F. L. Komdeur et al. “CD103+ tumor-infiltrating lymphocytes are tumor-reactive intraepithelial CD8+ T cells associated with prognostic benefit and therapy response in cervical cancer”. In: *Oncoimmunology* 6.9 (2017), e1338230 (cit. on p. 39).

[126] F. Djenidi et al. “CD8+ CD103+ tumor–infiltrating lymphocytes are tumor-specific tissue-resident memory T cells and a prognostic factor for survival in lung cancer patients”. In: *The Journal of Immunology* 194.7 (2015), pp. 3475–3486 (cit. on p. 39).

[127] P. Lohneis et al. “Cytotoxic tumour-infiltrating T lymphocytes influence outcome in resected pancreatic ductal adenocarcinoma”. In: *European journal of cancer* 83 (2017), pp. 290–301 (cit. on p. 39).

[128] K. Ademmer et al. “Effector T lymphocyte subsets in human pancreatic cancer: detection of CD8+ CD18+ cells and CD8+ CD103+ cells by multi-epitope imaging”. In: *Clinical & Experimental Immunology* 112.1 (1998), pp. 21–26 (cit. on p. 39).

[129] J. French et al. “T cell adhesion and cytolysis of pancreatic cancer cells: a role for E-cadherin in immunotherapy?” In: *British journal of cancer* 87.9 (2002), pp. 1034–1041 (cit. on p. 39).

[130] D. J. Gough et al. “IFNγ signaling—Does it mean JAK–STAT?” In: *Cytokine & growth factor reviews* 19.5-6 (2008), pp. 383–394 (cit. on p. 39).

[131] S. R. Chan et al. “STAT1-deficient mice spontaneously develop estrogen receptor α-positive luminal mammary carcinomas”. In: *Breast Cancer Research* 14 (2012), pp. 1–21 (cit. on p. 39).

[132] N. Sadanaga et al. “Local secretion of IFN-γ induces an antitumor response: comparison between T cells plus IL-2 and IFN-γ transfected tumor cells”. In: *Journal of Immunotherapy* 22.4 (1999), pp. 315–323 (cit. on p. 39).

[133] H. Ikeda, L. J. Old, and R. D. Schreiber. “The roles of IFNγ in protection against tumor development and cancer immunoediting”. In: *Cytokine & growth factor reviews* 13.2 (2002), pp. 95–109 (cit. on p. 39).

[134] G. L. Beatty and Y. Paterson. “IFN-γ-dependent inhibition of tumor angiogenesis by tumor-infiltrating CD4+ T cells requires tumor responsiveness to IFN-γ”. In: *The Journal of Immunology* 166.4 (2001), pp. 2276–2282 (cit. on p. 39).

[135] W.-G. Yu et al. “IL-12-induced tumor regression correlates with in situ activity of IFN-γ produced by tumor-infiltrating cells and its secondary induction of anti-tumor pathways”. In: *Journal of leukocyte biology* 62.4 (1997), pp. 450–457 (cit. on p. 39).

[136] Y. Ji and W. Zhang. “Th17 cells: positive or negative role in tumor?” In: *Cancer Immunology, Immunotherapy* 59 (2010), pp. 979–987 (cit. on p. 39).

[137] D. Kortleve et al. “TCR-engineered T-cells directed against Ropporin-1 constitute a safe and effective treatment for triple-negative breast cancer in near-clinical models”. en. 2024-01 (cit. on p. 40).

[138] D. Slavkovic-Lukic et al. “Rapid Screening of CAR T Cell Functional Improvement Strategies by Highly Multiplexed Single-Cell Secretomics”. In: *Cancer Immunotherapy: Methods and Protocols*. Springer, 2023, pp. 135–149 (cit. on p. 60).

[139] S. Krishna et al. “Stem-like CD8 T cells mediate response of adoptive cell immunotherapy against human cancer”. en. In: *Science* 370.6522 (2020-12), pp. 1328–1334 (cit. on p. 66).

[140] E. de Sousa et al. “Targeting Neoepitopes to Treat Solid Malignancies: Immunosurgery”. en. In: *Front. Immunol.* 12 (2021-07), p. 592031 (cit. on pp. 78, 89).

[141] U. Blache et al. “Advanced Flow Cytometry Assays for Immune Monitoring of CAR-T Cell Applications”. en. In: *Front. Immunol.* 12 (2021-05), p. 658314 (cit. on p. 82).

[142] X. Shen et al. “Persistence of Tumor Infiltrating Lymphocytes in Adoptive Immunotherapy Correlates With Telomere Length”. In: *Journal of Immunotherapy* 30 (2007), pp. 123–129. DOI: 10.1097/01.cji.0000211321.07654.b8 (cit. on p. 84).

[143] J. Zhou et al. “Telomere Length of Transferred Lymphocytes Correlates with In Vivo Persistence and Tumor Regression in Melanoma Patients Receiving Cell Transfer Therapy1”. In: *The Journal of Immunology* 175 (2005), pp. 7046–7052. DOI: 10.4049/jimmunol.175.10.7046 (cit. on p. 84).

[144] L. L. Smith, H. Coller, and J. M. Roberts. “Telomerase modulates expression of growth-controlling genes and enhances cell proliferation”. In: *Nature Cell Biology* 5 (2003), pp. 474–479. DOI: 10.1038/ncb985 (cit. on p. 84).

[145] A. Chen, Z. Velickovic, and J. Rasko. “Vector copy number quality control testing for CAR T-cells: critical validation parameters”. In: *Cytotherapy* 22.5, Supplement (2020-05), S142 (cit. on p. 85).

[146] L. Biasco et al. “In vivo tracking of T cells in humans unveils decade-long survival and activity of genetically modified T memory stem cells”. en. In: *Sci. Transl. Med.* 7.273 (2015-02), 273ra13 (cit. on p. 86).

A

Laboratory Consumables, Reagents and Equipments

Table A.1: Table of Antibodies and Fluorescent Dyes for Flow Cytometry

Component	Fluorophore	Catalog No.	Manufacturer	Clone/ Isotype
hu-aCD3	APC - A750	A94680	Beckman Coulter	UCHT1
hu-aCD4	PB	B49197	Beckman Coulter	13BB.2
hu-aCD4	APC	344614	Biolegend	SK3
hu-aCD8	KrO	B00067	Beckman Coulter	B9.11
hu-aCD8	A700	B76279	Beckman Coulter	B9.11
hu-aCD27	PC7	356412	Biolegend	M-T271
hu-aCD28	ECD	6607111	Beckman Coulter	CD28.2
hu-aCD39	ECD	328224	Biolegend	A1
hu-aCD45	KrO	A96416	Beckman Coulter	J33
hu-aCD45RA	APC A700	304120	Biolegend	HI100
hu-aCD57	PB	359608	Biolegend	HNK-1
hu-aCD69	PC7	A80710	Beckman Coulter	TP1.55.3

hu-aCD95	BV785	305646	BIolegend	DX2
hu-aCD103	FITC	B49222	Beckman Coulter	2G5
hu-aCD107a	APC	641581	BD	H4A3
hu-aCD183 (CXCR3)	BV650	353729	Biolegend	GO25H7
hu-aCD194 (CCR4)	ECD	359420	Biolegend	I.291H4
hu-aCD197 (CCR6)	PC7	B68132	Beckman Coulter	B-R35
hu-aCD197 (CCR7)	PE	B30632	Beckman Coulter	G043H7
hu-aCD199 (CCR9)	APC	567976	BD	C9Mab-1
hu-aCD223 (LAG3)	BV650	416-2239-42	Invitrogen	3DS223H
hu-aCD279 (PD1)	PC5.5	B36123	Beckman Coulter	PD1.3
hu-aTCRgd	PC5.5	A99021	Beckman Coulter	IMMU510
hu-aCD137 (4-1BB)	APC	550890	BD	4B4-1
hu-aIL-2	PC7	A53365	Beckman Coulter	CF1
hu-aIL-17	BV786	563745	BD	N49-653
hu-aIFNg	FITC	IM271U	Beckman Coulter	45.15
hu-aTNFα	PE	340512	BD	6401.1111

Fluorescent Dye - LIVE/DEAD™ Fixable Aqua	N/A	L34957	Invitrogen	N/A
Fluorescent Dye - LIVE/DEAD Fixable Blue Dead Cell Dye	N/A	L23105	Invitrogen	N/A

Table A.2: Table of Consumables and Reagents

Consumable	Catalog No.	Manufacturer
24-Well Cell Culture Plate	662160	Greiner bio-one
96-Well Plate	10265352	Corning
96-Well Plate Immuno MaxiSorp	442404	Thermo Scientific
Disposable Scalpel	DTX-507	Swann Morton
ArC™ negative beads	A10346	Invitrogen, Life
ArC™ reactive beads	A10346	Invitrogen
Negative beads	A10497	Invitrogen
UltraComp eBeads™ Plus	01-3333-42	Invitrogen
BD Cytofix/Cytoperm™	51-2090KZ	BD Biosciences
HS . Human Serum	H4422	Sigma-Aldrich
DPBS1x	14040133	Gibco
FBS	A5256701	Gibco
AIM-V Medium	12055083	Gibco
TMB - Substrate Solution	N301	Thermo Scientific
PMA phorbol-12-myristate-13-acetate	P8139-1MG	Sigma-Aldrich
Ficoll-Hypaque gradient	17-1440-03	GE Healthcare
Tube 15mL - centrifuge tubes	CLS430290	Corning
Tube 50mL - centrifuge tubes	CLS430791	Corning
BD GolgiStop™	554724	BD Biosciences
ELISA: Human IFNg (HRP)	3420-1H-20	MABTECH
Reagent Diluent Concentrate 2	DY995	R&D
anti-human CD3 antibody - clone OKT3	317302	Biolegend
Human IL-2	130-097-743	Miltenyi Biotec
Human IL-7	130-095-367	Miltenyi Biotec
Human IL15	130-095-760	Miltenyi Biotec
Human Il-21	130-095-767	Miltenyi Biotec

Table A.3: Table of Equipments or Devices

Device	Manufacturer	Usage
CytoFLEX LX HTS™	Beckman Coulter	flow cytometry
Tecan SPARk 10M microplate reader	SPARK	ELISA plate reader
HERAcell 240i CO2 Inkubator	Thermo Scientific	incubator
Heraeus Multifuge X3 FR	Thermo Scientific	centrifuge
HERASAFE 2030i Hood	Thermo Scientific	sterile work bench
IKA® Vortex1	IKA®	vortex
Neubauer Counting Chamber	Paul Marienfeld GmbH	cell counting

Python Packages and Python Scripts

Table B.1: Packages Requirements for Python scripts

Package	Version
FlowKit	1.0.1
matplotlib	3.8.0
numpy	1.24.1
pandas	1.5.3
scipy	1.11.3
xlrd	2.0.1
graphviz	0.20.1
mysql	0.0.3
mysql-connector-python	8.2.0
ttkbootstrap	1.10.1

Python Script: elisa.py

```
1  import os
2  from scipy.optimize import curve_fit
3  import numpy as np
4  import matplotlib.pyplot as plt
5  import matplotlib.patches as mpatches
6  import xlrd
7  import csv
8  from itertools import compress
9  import shutil
10
11
12 # Functions
13
```

```
14
15  def read_plate(sheet):
16      """
17          Given a spreadsheet of a plate, read it and extract the data
18
19          :param sheet: The workspace file path
20
21          :return: patient_id: Patient ID
22          :return: sample_id: Sample ID
23          :return: cytokine: Analyzed cytokine name
24          :return: plate_values: dictionary to hold target labels and the
            ↪  respective ODs
25      """
26      # Get patient ID and cytokine tested
27      patient_id = int(sheet.cell_value(0, 1))
28      sample_id = int(sheet.cell_value(1, 1))
29      cytokine = sheet.cell_value(2, 1)
30
31      # Plate layout dictionary
32      plate_layout = {}
33
34      # How many rows the plate layout table is ahead of the start
35      layout_row_offset = 13
36
37      # How many columns the plate layout table (and the other tables) is ahead of
        ↪  the start
38      col_offset = 2
39
40      # Fill the dictionary with the row and column numbers of cells containing
        ↪  each label
41      for row in range(0, 8):
42          for col in range(0, 12):
43              label = sheet.cell_value(row + layout_row_offset, col + col_offset)
44              if label not in plate_layout:
45                  plate_layout[label] = [[row, col]]
46              else:
47                  plate_layout[label].append([row, col])
48
49      # How many rows the values table is ahead of the start
50      values_row_offset = 4
51
52      # How many rows the dilution factor table is ahead of the start
53      dilution_row_offset = 22
54
55      # Dictionary to hold plate labels and respective values
56      plate_values = {}
```

```
57
58      # Fill values dictionary with label as key and respective average value
        ↪ times the respective dilution factor
59      for label, indices in plate_layout.items():
60          sum = 0
61          for cell in indices:
62              sum += sheet.cell_value(cell[0] + values_row_offset, cell[1] +
                ↪ col_offset)
63          value = sum / len(indices)
64          dilution_factor = sheet.cell_value(cell[0] + dilution_row_offset,
            ↪ cell[1] + col_offset)
65          plate_values[label] = value * dilution_factor
66
67      # If the medium exists as control in the plate, subtract it from all other
        ↪ mappings (except Standards)
68      if 'medium' in plate_layout:
69          for k, v in plate_values.items():
70              if not k.startswith('STD') and k != 'medium':
71                  plate_values[k] = v - plate_values['medium']
72
73      return patient_id, sample_id, cytokine, plate_values
74
75
76  # Curve fit function
77  def log4pl(x, A, B, C, D):
78      """
79          4 parameter logistic regression function
80
81          :param x: value
82          :param A: minimum value of OD
83          :param B: slope of the curve at point C
84          :param C: point of inflection
85          :param D: maximum value of OD
86          :return: concentration value
87      """
88      return ((A - D) / (1.0 + ((x / C) ** B))) + D
89
90
91  def to_pct(value, max):
92      """
93          Convert the given value to percentage, where max is the reference for
            ↪ 100%
94
95          :param value: value
96          :param max: maximum value
97          :return: value in percentage
```

```
    """
    return value / max * 100


def calculate_change(medium, peptide):
    return round(float(peptide * 100 / medium), 3)


#
↪  ----------------------------------------------------------------------------
# Execution starts here

if os.path.exists(f'{os.getcwd()}/Patients'):
    shutil.rmtree(f'{os.getcwd()}/Patients', ignore_errors=True)

controls = ['PHA', 'OKT3', 'medium']
antigens_controls = ['CMV', 'EBNA', 'M1 (mix)', 'ESAT6', 'Haemagluttinin']

base_dir = "Data/DATA_Raw_files/ELISA"
analysis_type = ''

if not os.path.exists('Patients'):
    os.mkdir('Patients')

for dir in os.listdir(base_dir):
    if str(dir) == 'WBA':
        analysis_type = 'WBA'
    else:
        analysis_type = 'TIL'

    for file in os.listdir(f'{base_dir}/{dir}'):
        if file.endswith('xls'):
            wb = xlrd.open_workbook(f'{base_dir}/{dir}/{file}')
            for i in range(len(wb.sheet_names())):
                sheet = wb.sheet_by_index(i)
                # Get the plate values
                patient_id, sample_id, cytokine, plate_values =
                ↪  read_plate(sheet)

                print(f'Analysis Type: {analysis_type} | Patient {patient_id} |
                ↪  Sample {sample_id}')

                # Get standards
                plate_values_std = {k: v for k, v in plate_values.items() if
                ↪  k.startswith('STD')}

```

```
                # Filter out targets that did not react
                targets = {k: v for k, v in plate_values.items() if v > 0 and
                ↪  not k.startswith('STD') and k != 'medium'}

                with open('Patients/peptides.txt', 'a') as f:
                    for peptide in targets.keys():
                        f.write(f'{peptide}\n')

                print(f'Targets Used: {list(plate_values.keys())}')
                print(f'Targets That Reacted: {list(targets.keys())}\n')

                filepath = f'Patients/Patient
                ↪  {patient_id}/{analysis_type}/Sample {sample_id}/Cytokine
                ↪  {cytokine}'

                if not os.path.exists(filepath):
                    os.makedirs(filepath)

                # Build standard curve
                x = [1000, 500, 250, 125, 62.5, 31.25, 15.625, 7.8125]  #
                ↪  Concentration
                y = list(plate_values_std.values())  # Optical Density

                # X values to draw the curve
                x_min = min(x)
                x_max = max(x)
                step = 0.01
                x_new = np.arange(x_min, x_max, step)

                params, _ = curve_fit(log4pl, x, y)
                A, B, C, D = params[0], params[1], params[2], params[3]
                print("4PL parameters: A = " + str(round(A, 4)) + ", B = " +
                ↪  str(round(B, 4)) +
                      ", C = " + str(round(C, 4)) + ", D = " + str(round(D, 4)))

                # Y values to draw the curve
                yfit1_new = ((A - D) / (1.0 + ((x_new / C) ** B))) + D

                # Plot the curve
                plt.plot(x, y, 'r+', label="y-original")
                plt.plot(x_new, yfit1_new, label="y=((A-D)/(1.0+((x/C)^B))) +
                ↪  D")
                plt.xlabel('OD')
                plt.ylabel('Standards (log)')
                plt.xscale("log")
                plt.legend(loc='best', fancybox=True, shadow=True)
```

```
                plt.grid(True)
                plt.savefig(f'{filepath}/{analysis_type}_standard_curve.png')
                plt.clf()

                absError = log4pl(x, A, B, C, D) - y
                SE = np.square(absError)  # squared errors
                MSE = np.mean(SE)  # mean squared errors
                RMSE = np.sqrt(MSE)  # Root Mean Squared Error, RMSE
                Rsquared = 1.0 - (np.var(absError) / np.var(y))
                print('RMSE:', RMSE)
                print('R-squared:', Rsquared)
                print()

                err = y - log4pl(x, A, B, C, D)
                print(f'Err: {err}')
                ss = sum(np.square(err))
                print(f'SS: {ss}')

                # Plot bar chart
                concentrations = {k: round(log4pl(v, A, B, C, D), 4) for k, v in
                ↪  targets.items()}
                data_min = min(concentrations.values())
                data_max = max(concentrations.values())

                fig, ax = plt.subplots(figsize=(16, 9))

                labels = list(concentrations.keys())
                values = list(concentrations.values())

                plt.xlabel(f'{cytokine} pg/mL', fontsize=15)

                controls_mask = [bool(item in controls) for item in labels]
                antigens_mask = [bool(item in antigens_controls) for item in
                ↪  labels]
                none_mask = [not (a or b) for a, b in zip(controls_mask,
                ↪  antigens_mask)]

                ax.barh(list(compress(labels, controls_mask)),
                ↪  list(compress(values, controls_mask)), color='blue')
                ax.barh(list(compress(labels, antigens_mask)),
                ↪  list(compress(values, antigens_mask)), color='red')
                ax.barh(list(compress(labels, none_mask)), list(compress(values,
                ↪  none_mask)), color='green')

                plt.yticks(fontsize=12)

```

```
                # Add padding between axes and labels
                ax.xaxis.set_tick_params(pad=5)
                ax.yaxis.set_tick_params(pad=10)

                # Add x, y gridlines
                ax.grid(color='grey',
                        linestyle='-.', linewidth=0.5,
                        alpha=0.2)

                # Add annotation to bars
                for i in ax.patches:
                    plt.text(i.get_width(), i.get_y() + 0.35,
                             f' {str(round((i.get_width()), 4))}',
                             fontsize=10, fontweight='bold',
                             color='grey')

                plt.title(f'Patient {patient_id} - Sample {sample_id}',
                ↪  fontsize=25)

                red_patch = mpatches.Patch(color='red', label='Viral Antigens')
                blue_patch = mpatches.Patch(color='blue', label='Controls')
                green_patch = mpatches.Patch(color='green', label='Peptides')

                plt.legend(handles=[red_patch, blue_patch, green_patch])

                # Show Plot
                plt.subplots_adjust(left=0.2)
                plt.xlim(data_min - data_min/100, data_max + data_max/50)

                plt.savefig(f'{filepath}/{analysis_type}_concentrations.png')
                plt.clf()
                plt.subplots_adjust(left=0.1)

                # Get ODs in order to calculate relative percentages
                sorted_ods = sorted(targets.items(), key=lambda x: x[1])
                sorted_ods = dict(sorted_ods)

                od_values = list(sorted_ods.values())
                max_value = od_values[len(od_values) - 1]

                sorted_ods = {k: round(to_pct(v, max_value), 2) for k, v in
                ↪  sorted_ods.items()}

                peptide_reactions = {k: calculate_change(plate_values['medium'],
                ↪  v) for k, v in targets.items()}

```

```
263                 print(peptide_reactions)
264
265                 # Write Peptide reactions data to sample file
266                 with open(os.path.join(filepath, f'peptide_reactions.csv'), 'w',
                    ↪ newline='\n') as f:
267                     w = csv.writer(f)
268                     w.writerow(peptide_reactions.keys())
269                     w.writerow(peptide_reactions.values())
270
271                 # Write Peptide reactions data to global file
272                 with open(f'{os.getcwd()}/Patients/{analysis_type}
273                 _global_peptide_reactions.csv', 'a', newline='') as f:
274                     w = csv.writer(f)
275                     w.writerow(peptide_reactions.keys())
276                     w.writerow(peptide_reactions.values())
277
278                 # Remove controls
279                 sorted_ods = {k: v for k, v in sorted_ods.items() if k not in
                    ↪ controls and k not in antigens_controls}
280                 concentrations = {k: v for k, v in concentrations.items() if k
                    ↪ not in controls and k not in antigens_controls}
281
282                 print(f'Sorted ODs: {sorted_ods}\n')
283                 print(f'Concentrations: {concentrations}')
284
285                 # Write OD data to sample file
286                 with open(os.path.join(filepath, f'{analysis_type}_ODs.csv'),
                    ↪ 'w', newline='\n') as f:
287                     w = csv.writer(f)
288                     w.writerow(sorted_ods.keys())
289                     w.writerow(sorted_ods.values())
290
291                 # Write OD data to global file
292                 with open(f'{os.getcwd()}
293                 /Patients/{analysis_type}_global_ODs.csv', 'a', newline='') as
                    ↪ f:
294                     w = csv.writer(f)
295                     w.writerow(sorted_ods.keys())
296                     w.writerow(sorted_ods.values())
297
298                 # Write concentrations data to sample file
299                 with open(os.path.join(filepath,
                    ↪ f'{analysis_type}_Concentrations.csv'), 'w', newline='\n')
                    ↪ as f:
300                     w = csv.writer(f)
301                     w.writerow(labels)
```

```
                    w.writerow(values)

                # Write concentrations data to global file
                with open(f'{os.getcwd()}/Patients/{analysis_type}
                _global_concentrations.csv', 'a', newline='') as f:
                    w = csv.writer(f)
                    w.writerow(labels)
                    w.writerow(values)


```

Python Script: flowjo.py

```
# NOTE: This script requires graphviz to be installed in order to run:
↪  https://graphviz.org/download/

import os
import flowkit as fk
import csv
import graphviz
import shutil


# Functions

def get_fluoro_labels(sample, events):
    """
        Extract fluorescent event labels from a sample

        :param sample: The sample instance
        :param events: The gate events
        :return: an array containing the names of the columns that correspond to
        ↪  the fluorescent labels
    """
    fluoro_labels = []
    for chan_idx in sample.fluoro_indices:
        fluoro_labels.append(sample.pnn_labels[chan_idx])

    events_labels = []
    for label in fluoro_labels:
        for other_label in list(events.columns):
            if label in other_label:
                events_labels.append(other_label)
    return events_labels
```

```
30
31
32  def get_wsp_files(dir):
33      """
34          Recursive function to get all the workspace files from a directory
35
36          :param dir: The base directory
37          :return: an array with the paths to all workspace files
38      """
39      wsp_files = []
40      for item in os.listdir(dir):
41          item_path = os.path.join(dir, item)
42          # if the item is a directory, get all workspace files from that
            ↪  directory
43          if os.path.isdir(item_path):
44              aux_list = get_wsp_files(item_path)
45              for file in aux_list:
46                  wsp_files.append(file)
47          # else, if it is a file, add it to the list if it is a workspace file
48          elif os.path.isfile(item_path):
49              if item.endswith('.wsp'):
50                  wsp_files.append(item_path)
51      return wsp_files
52
53
54  def extract_data(gate, gate_path, gate_pct, mfi_comp, parent, sample, sample_id,
    ↪  sample_results, wsp, labels_dict):
55      """
56          Refactored piece of code that extracts the MFI for required gates and
            ↪  gate percentages
57          and appends them to the respective lists
58
59          :param gate: The gate
60          :param gate_path: A tuple that represents the path to the gate
61          :param gate_pct: The gate percentages list
62          :param mfi_comp: The MFI comparison list
63          :param parent: The parent gate
64          :param sample: The sample instance
65          :param sample_id: The ID of the sample instance
66          :param sample_results: The sample results dataframe
67          :param wsp: the workspace file
68          :param labels_dict: labels dictionary
69
70          :return: None
71      """
72
```

```
    # Get the index for the correct report entry
    idx = -1
    for item in list(sample_results.report.loc[sample_results.report['gate_path'] == gate_path].gate_name):
        idx += 1
        if gate == item.strip():
            break

    # retrieve the relative percent from the report that matches the gate being handled
    relative_pct = float(sample_results.report.loc[sample_results.report['gate_path'] == gate_path].iloc[[idx]]
                         .relative_percent)

    # add the value to the gate percentages dictionary
    gate_name = ''
    if pre == 'Thoming':
        if th1_gate(gate=gate, gate_path=gate_path, parent=parent):
            gate_name = gate + " Th1" if "th1" not in gate.lower() else gate
        elif th2_gate(gate, gate_path, parent):
            gate_name = gate + " Th2" if "th2" not in gate.lower() else gate
        elif th17_gate(gate, gate_path, parent):
            gate_name = gate + " Th17" if "th17" not in gate.lower() else gate
        elif th1star_gate(gate, gate_path, parent):
            gate_name = gate + " Th1*" if "th1*" not in gate.lower() or "th1+" not in gate.lower() else gate
        elif tscm_gate(gate, gate_path, parent):
            gate_name = gate + " Tscm" if "tscm" not in gate.lower() else gate
        gate_pct[f'{parent} | {gate_name}'] = round(relative_pct, 3)

    gate_pct[f'{parent} | {gate}'] = round(relative_pct, 3)

    # Now to calculate the MFI
    gate_events = wsp.get_gate_events(sample_id, gate_name=gate, gate_path=gate_path)

    event_labels = get_fluoro_labels(sample, gate_events)
    gate_events = gate_events[event_labels]

    # Calculate the average of the values on the gate events dataframe
    gate_mfi = gate_events.mean()

    # Get the labels from the mfi dataframe
    labels = gate_mfi.index.values.tolist()
```

```
112
113     gate = gate.strip()
114
115     if pre == 'ICS':
116         # Get the dictionary of indices in order to match the labels to the
            ↪  gates
117
118         # ICS labels
119         list1_ics = ['FL1-H IFN-y B525-FITC-H', 'FL1-A IFN-y B525-FITC-A',
            ↪  'FL3-H TCR GD B690-PC5.5-H',
120                      'FL3-A TCR GD B690-PC5.5-A', 'FL4-H TNF-a Y585-PE-H',
                         ↪  'FL4-A TNF-a Y585-PE-A',
121                      'FL8-H IL-2 Y763-PC7-H', 'FL8-A IL-2 Y763-PC7-A', 'FL10-H
                         ↪  CD8 R712-APCA700-H',
122                      'FL10-A CD8 R712-APCA700-A', 'FL11-H CD3 R763-APCA750-H',
                         ↪  'FL11-A CD3 R763-APCA750-A',
123                      'FL12-H CD4 V450-PB-H', 'FL12-A CD4 V450-PB-A', 'FL14-H
                         ↪  Live Dead V610-H',
124                      'FL14-A Live Dead V610-A', 'FL16-H IL-17A V763-H', 'FL16-A
                         ↪  IL-17A V763-A']
125         list2_ics = ['FL1-H INFg B525-FITC-H', 'FL1-A INFg B525-FITC-A', 'FL8-H
            ↪  CD3 Y763-PC7-H', 'FL8-A CD3 Y763-PC7-A',
126                      'FL11-H CD8 R763-APCA750-H', 'FL11-A CD8 R763-APCA750-A',
                         ↪  'FL12-H CD4 V450-PB-H',
127                      'FL12-A CD4 V450-PB-A', 'FL14-H Live V610-H', 'FL14-A Live
                         ↪  V610-A', 'FL16-H IL17 V763-H',
128                      'FL16-A IL17 V763-A']
129
130         list_num = None
131         if labels == list1_ics:
132             list_num = 1
133         elif labels == list2_ics:
134             list_num = 2
135
136         indices_dict = labels_dict[list_num]
137
138         idx = indices_dict[gate]
139
140         if gate in indices_dict:  # if the gate is in the dictionary
141             mfi = round(gate_mfi[idx], 2)
142             if mfi != mfi:  # if there are no events, the mfi dataframe will not
                ↪  have the index found
143                 mfi_comp[f'{parent} | {gate}'] = 'No Events'
144             else:
145                 mfi_comp[f'{parent} | {gate}'] = mfi
146
```

```
147     elif pre == 'Tcell':
148         # Get the dictionary of indices in order to match the labels to the
            ↪  gates
149 
150         # Tcell labels
151         list1_tcell = ['FITC-A', 'BV421-A', 'BV510-A', 'BV605-A', 'BV650-A',
            ↪  'BV785-A', 'APC-A', 'APC-Alexa 700-A',
152                        'APC-Alexa 700-A', 'APC-Cy7-A', 'PE-A', 'PE-Texas Red-A',
                           ↪  'PE-Cy5-5-A', 'PE-Cy7-A']
153         # Equal to
154         list4_tcell = ['FITC-A CD45RA', 'BV421-A CD57', 'BV510-A CD45', 'BV605-A
            ↪  Live Dead', 'BV650-A LAG-3',
155                        'BV711-A CD8intra', 'BV785-A CD95', 'APC-A CD4',
                           ↪  'APC-Alexa 700-A CD8extra',
156                        'APC-Alexa 700-A CD8extra', 'APC-Cy7-A CD3', 'PE-A CCR7',
                           ↪  'PE-Texas Red-A CD28', 'PE-Cy7-A CD27',
157                        'PerCP-Cy5-5-A PD-1']
158 
159         list2_tcell = ['FL1-H CD103 B525-FITC-H', 'FL1-A CD103 B525-FITC-A',
            ↪  'FL2-H CD39 B610-ECD-H',
160                        'FL2-A CD39 B610-ECD-A', 'FL3-H GD B690-PC5.5-H', 'FL3-A
                           ↪  GD B690-PC5.5-A',
161                        'FL4-H CCR7 Y585-PE-H', 'FL4-A CCR7 Y585-PE-A', 'FL8-H
                           ↪  CD69 Y763-PC7-H', 'FL8-A CD69 Y763-PC7-A',
162                        'FL9-H 4-1BB R660-APC-H', 'FL9-A 4-1BB R660-APC-A',
                           ↪  'FL10-H CD45RA R712-APCA700-H',
163                        'FL10-A CD45RA R712-APCA700-A', 'FL11-H CD3 APC-Cy7
                           ↪  R763-APCA750-H',
164                        'FL11-A CD3 APC-Cy7 R763-APCA750-A', 'FL12-H CD4
                           ↪  V450-PB-H', 'FL12-A CD4 V450-PB-A',
165                        'FL13-H CD8 V525-KrO-H', 'FL13-A CD8 V525-KrO-A', 'FL14-H
                           ↪  LiveDead V610-H',
166                        'FL14-A LiveDead V610-A', 'FL15-H LAG3 V660-H', 'FL15-A
                           ↪  LAG3 V660-A', 'FL16-H CD95 V763-H',
167                        'FL16-A CD95 V763-A']
168         # Equal to
169         list3_tcell = ['FL1-H B525-FITC-H', 'FL1-A B525-FITC-A', 'FL2-H
            ↪  B610-ECD-H', 'FL2-A B610-ECD-A',
170                        'FL3-H B690-PC5.5-H', 'FL3-A B690-PC5.5-A', 'FL4-H
                           ↪  Y585-PE-H', 'FL4-A Y585-PE-A',
171                        'FL8-H Y763-PC7-H', 'FL8-A Y763-PC7-A', 'FL9-H
                           ↪  R660-APC-H', 'FL9-A R660-APC-A',
172                        'FL10-H R712-APCA700-H', 'FL10-A R712-APCA700-A', 'FL11-H
                           ↪  R763-APCA750-H',
173                        'FL11-A R763-APCA750-A', 'FL12-H V450-PB-H', 'FL12-A
                           ↪  V450-PB-A', 'FL13-H V525-KrO-H',
```

```
                        'FL13-A V525-KrO-A', 'FL14-H V610-H', 'FL14-A V610-A',
                        ↪  'FL15-H V660-H', 'FL15-A V660-A',
                        'FL16-H V763-H', 'FL16-A V763-A']

        list_num = None
        if labels == list1_tcell or labels == list4_tcell:
            list_num = 1
        elif labels == list2_tcell or labels == list3_tcell:
            list_num = 2

        if labels == list1_tcell:
            list1_tcell.insert(5, 'BV711-A')

        indices_dict = labels_dict[list_num]

        gates = {}  # Gates data structure

        # Add gates to the gates data structure given the gate being handled
        if gate in tcm_gate_aliases:
            gates['TCM'] = ['CD45RA', 'CCR7']
        elif gate in tem_gate_aliases:
            gates['TEM'] = ['CD45RA', 'CCR7']
        elif gate in teff_gate_aliases:
            gates['TEFF'] = ['CD45RA', 'CCR7']
        elif gate in precursor_gate_aliases:
            gates['PRECURSOR'] = ['CD45RA', 'CCR7']
        elif len(gate.strip().split(' ')) > 1:
            check_gate_suffix(gate, gates)
        else:
            if gate.endswith('+') or gate.endswith('-'):
                gates[gate] = [gate[:-1]]
            else:
                gates[gate] = [gate]

        extract_gate_mfi(gate_mfi, gates, indices_dict, mfi_comp)

    elif pre == 'Thoming':
        # Get the dictionary of indices in order to match the labels to the
        ↪  gates

        list1_thoming = ['FITC-A CD103', 'PerCP-Cy5-5-A CCR4', 'BV421-A CD4',
        ↪  'BV510-A CD8', 'BV605-A LIVE',
                         'BV650-A CXCR3', 'BV785-A CD95', 'APC-A CCR9 AF677',
                         ↪  'APC-Alexa 700-A CD45RA',
                         'APC-Alexa 700-A CD45RA', 'APC-Cy7-A CD3', 'PE-A CCR7',
                         ↪  'PE-Cy7-A CCR6']
```

```
215         list2_thoming = ['FL1-H CD103 B525-FITC-H', 'FL1-A CD103 B525-FITC-A',
            ↪  'FL2-H CCR4 B610-ECD-H',
216                          'FL2-A CCR4 B610-ECD-A', 'FL3-H GD B690-PC5.5-H',
                             ↪  'FL3-A GD B690-PC5.5-A',
217                          'FL4-H CCR7 Y585-PE-H', 'FL4-A CCR7 Y585-PE-A', 'FL8-H
                             ↪  CCR6 Y763-PC7-H',
218                          'FL8-A CCR6 Y763-PC7-A', 'FL9-H CCR9 R660-APC-H',
                             ↪  'FL9-A CCR9 R660-APC-A',
219                          'FL10-H CD45RA R712-APCA700-H', 'FL10-A CD45RA
                             ↪  R712-APCA700-A', 'FL11-H CD3 R763-APCA750-H',
220                          'FL11-A CD3 R763-APCA750-A', 'FL12-H CD4 V450-PB-H',
                             ↪  'FL12-A CD4 V450-PB-A',
221                          'FL13-H CD8 V525-KrO-H', 'FL13-A CD8 V525-KrO-A',
                             ↪  'FL14-H Live dead V610-H',
222                          'FL14-A Live dead V610-A', 'FL15-H CxCR3 V660-H',
                             ↪  'FL15-A CxCR3 V660-A', 'FL16-H CD95 V763-H',
223                          'FL16-A CD95 V763-A']
224         list3_thoming = ['FITC-A CD103', 'BV421-A CD4', 'BV510-A CD8', 'BV605-A
            ↪  Live Dead', 'BV650-A CXCR3',
225                          'BV711-A CD8 intra', 'BV785-A CD95', 'APC-A CCR9',
                             ↪  'APC-Alexa 700-A CD45RA',
226                          'APC-Alexa 700-A CD45RA', 'APC-Cy7-A CD3', 'PE-A CCR7',
                             ↪  'PE-Texas Red-A CCR4', 'PE-Cy7-A CCR6',
227                          'PerCP-Cy5-5-A TCRgd']
228         list4_thoming = ['FL1-H CD103 B525-FITC-H', 'FL1-A CD103 B525-FITC-A',
            ↪  'FL2-H CCR4 B610-ECD-H',
229                          'FL2-A CCR4 B610-ECD-A', 'FL4-H CCR7 Y585-PE-H', 'FL4-A
                             ↪  CCR7 Y585-PE-A',
230                          'FL7-H TCRgd Y710-PC5.5-H', 'FL7-A TCRgd Y710-PC5.5-A',
                             ↪  'FL8-H CCR6 Y763-PC7-H',
231                          'FL8-A CCR6 Y763-PC7-A', 'FL9-H CCR9 R660-APC-H',
                             ↪  'FL9-A CCR9 R660-APC-A',
232                          'FL10-H CD45RA R712-APCA700-H', 'FL10-A CD45RA
                             ↪  R712-APCA700-A', 'FL11-H CD3 R763-APCA750-H',
233                          'FL11-A CD3 R763-APCA750-A', 'FL12-H CD4 V450-PB-H',
                             ↪  'FL12-A CD4 V450-PB-A',
234                          'FL13-H CD8 V525-KrO-H', 'FL13-A CD8 V525-KrO-A',
                             ↪  'FL14-H Live dead V610-H',
235                          'FL14-A Live dead V610-A', 'FL15-H CXCR3 V660-H',
                             ↪  'FL15-A CXCR3 V660-A', 'FL16-H CD95 V763-H',
236                          'FL16-A CD95 V763-A']
237
238         list_num = None
239         if labels == list1_thoming:
240             list_num = 1
241         elif labels == list2_thoming:
```

```
242             list_num = 2
243         elif labels == list3_thoming:
244             list_num = 3
245         elif labels == list4_thoming:
246             list_num = 4
247 
248         indices_dict = labels_dict[list_num]
249 
250         gates = {}  # Gates data structure
251 
252         # Add gates to the gates data structure given the gate being handled
253         if th1_gate(gate, gate_path, parent):
254             gates['Th1'] = ['CXCR3', 'CCR6']
255         elif th2_gate(gate, gate_path, parent):
256             gates['Th2'] = ['CCR4', 'CCR6']
257         elif th17_gate(gate, gate_path, parent):
258             gates['Th17'] = ['CCR4', 'CCR6']
259         elif th1star_gate(gate, gate_path, parent):
260             gates['Th1*'] = ['CCR4', 'CXCR3', 'CCR6']
261         elif tscm_gate(gate, gate_path, parent):
262             gates['Tscm'] = ['CD45RA', 'CCR7', 'CD95']
263         elif len(gate.strip().split(' ')) > 1:
264             check_gate_suffix(gate, gates)
265         else:
266             if gate.endswith('+') or gate.endswith('-'):
267                 gates[gate] = [gate[:-1]]
268             else:
269                 gates[gate] = [gate]
270 
271         extract_gate_mfi(gate_mfi, gates, indices_dict, mfi_comp)
272 
273 
274 def extract_gate_mfi(gate_mfi, gates, indices_dict, mfi_comp):
275     """
276         Get the MFI for the gate or gates from the mfi dataframe and store it
277 
278         :param gate_mfi: the MFI dataframe
279         :param gates: the gates to extract
280         :param indices_dict: the dictionary for the indices where all gates are
            ↪  located in the mfi dataframe
281         :param mfi_comp: the MFI comparison data structure
282         :return: None
283     """
284 
285     for label, gate_list in gates.items():
286         if type(gate_list) == list:
```

```
            mfi_comp[label] = {}
            for item in gate_list:
                idx = indices_dict[item]
                mfi = float(round(gate_mfi[idx], 2))
                if mfi != mfi:
                    mfi_comp[label][item] = 'No Events'
                else:
                    mfi_comp[label][item] = mfi
        else:
            idx = indices_dict[gate_list]
            mfi = round(gate_mfi[idx], 2)
            if mfi != mfi:
                mfi_comp[label] = 'No Events'
            else:
                mfi_comp[label] = mfi


def check_gate_suffix(gate, gates):
    """
        Checks the gate suffix in order to match the indices dictionary
        :param gate: the gate being handled
        :param gates: the gates data structure
        :return: None
    """

    gate = gate.strip()
    gates[gate] = []
    for item in gate.split(' '):
        if item != '+' and item != '-':
            if item[-1] == '+' or item[-1] == '-':
                gates[gate].append(item[:-1])
            else:
                gates[gate].append(item)


def th1star_gate(gate, gate_path, parent):
    """
        Checks if the gate being handled is the Th1* gate
        :param gate: the gate being handled
        :param gate_path: the gate path
        :param parent: the parent of the gate
        :return: true if it is, false otherwise
    """

    return 'th1+' in str(gate).lower() or 'th1*' in str(gate).lower() or (
            (gate == 'CCR4+ CXCR3+' and parent == 'CCR6+') and 'CD4+' in
            ↪ gate_path)
```

```
333
334
335 def th17_gate(gate, gate_path, parent):
336     """
337         Checks if the gate being handled is the Th17 gate
338         :param gate: the gate being handled
339         :param gate_path: the gate path
340         :param parent: the parent of the gate
341         :return: true if it is, false otherwise
342     """
343
344     return 'th17' in str(gate).lower() or ((gate == 'CCR4+' and parent ==
    ↪  'CCR6+') and 'CD4+' in gate_path)
345
346
347 def th2_gate(gate, gate_path, parent):
348     """
349         Checks if the gate being handled is the Th2 gate
350         :param gate: the gate being handled
351         :param gate_path: the gate path
352         :param parent: the parent of the gate
353         :return: true if it is, false otherwise
354     """
355
356     return 'th2' in str(gate).lower() or ((gate == 'CCR4+' and parent ==
    ↪  'CCR6-') and 'CD4+' in gate_path)
357
358
359 def th1_gate(gate, gate_path, parent):
360     """
361         Checks if the gate being handled is the Th1 gate
362         :param gate: the gate being handled
363         :param gate_str: the gate converted to lowercase
364         :param gate_path: the gate path
365         :param parent: the parent of the gate
366         :return: true if it is, false otherwise
367     """
368
369     th1 = 'th1' in str(gate).lower() and 'th17' not in str(gate).lower() and
    ↪  'th1+' not in str(
370         gate).lower() and 'th1*' not in str(gate).lower()
371     return th1 or ((gate == 'CXCR3+' and parent == 'CCR6-') and 'CD4+' in
    ↪  gate_path)
372
373
374 def tscm_gate(gate, gate_path, parent):
```

```
    """
        Checks if the gate being handled is the Tscm gate
        :param gate: the gate being handled
        :param gate_str: the gate being handled in converted to lowercase
        :param gate_path: the gate path
        :param parent: the parent of the gate
        :return: true if it is, false otherwise
    """

    return 'tscm' in str(gate).lower() or ((gate == 'CD45RA+ CCR7+' and parent == 'CD95+') and 'CD8+' in gate_path)


def thoming_gate_of_interest(gate, gate_path):
    """
        Checks if the gate is one of the Thoming gates to be analyzed
        :param gate: the gate being handled
        :param gate_path: the gate path
        :return: true if it is, false otherwise
    """

    gate_str = str(gate).lower()
    parent = gate_path[-1]
    th1_gate = 'th1' in gate_str and 'th17' not in gate_str and 'th1+' not in gate_str and 'th1*' not in gate_str
    tscm_gate = 'tscm' in gate_str or ((gate == 'CD45RA+ CCR7+' and parent == 'CD95+') and 'CD8+' in gate_path)
    return tscm_gate or \
        (((th1_gate or 'th2' in gate_str or 'th1*' in gate_str or 'th1+' in gate_str or 'th17' in gate_str) or
          (gate == 'CCR4+ CXCR3+' and parent == 'CCR6+') or  # Th1*
          (gate == 'CCR4+' and parent == 'CCR6+') or  # Th17
          (gate == 'CCR4+' and parent == 'CCR6-') or  # Th2
          (gate == 'CXCR3+' and parent == 'CCR6-')) and 'CD4+' in gate_path)  # Th1


def write_sample_id(sample_id):
    with open('Samples/sample_ids.txt', 'a') as f:
        f.write(f'{sample_id[:-4]}\n')


def analyze(wsp_file, analysis, pre):
    """
        Analyze the workspace and its corresponding sample files given the type of
```

```
        analysis(TIL, PBMC) and the prefix (ICS, Tcell, Thoming)

        :param wsp_file: The workspace file path
        :param analysis: The type of analysis
        :param pre: The prefix of the analysis

        :return: None
    """

    print(f'\nWORKSPACE {wsp_file}\n')

    # Create a Workspace with the path to our WSP file and FCS files
    wsp = fk.Workspace(wsp_file, fcs_samples=os.path.dirname(wsp_file),
    ↪  ignore_missing_files=True)

    # Loop through all sample groups
    sample_group = 'All Samples'

    # Analyze samples in order to fetch analysis results
    wsp.analyze_samples(sample_group)

    # Get sample file names
    sample_list = wsp.get_sample_ids(group_name=sample_group)

    # loop through samples
    for sample_id in sample_list:
        print("#####################################################")
        print(f'\nSAMPLE {sample_id}\n')

        write_sample_id(sample_id)

        # Get the sample from its ID
        sample = wsp.get_sample(sample_id)

        # Get the sample date
        date = sample.acquisition_date

        # Get gate hierarchy
        hierarchy = wsp.get_gate_hierarchy(sample_id)
        print(hierarchy)

        # Get the sample results dataframe
        sample_results = wsp.get_gating_results(sample_id)

        # Data structures to hold our data
        mfi_comp = {}
```

```
        gate_pct = {}

        # gate aliases checking for Gamma Delta + gate
        gd_gate_name = ''
        for alias in gd_gate_aliases:
            if alias in hierarchy:
                gd_gate_name = alias
                break

        if pre == 'ICS':
            # gate aliases checking for IFN-g gate
            ifng_gate_name = ''
            for alias in ifng_gate_aliases:
                if alias in hierarchy:
                    ifng_gate_name = alias
                    break

            ics_gates = [f'CD4+/{ifng_gate_name}', f'CD8+/{ifng_gate_name}',
            ↪  f'{gd_gate_name}/{ifng_gate_name}']

            labels_dict_ics = {
                1: {ifng_gate_name: 1, 'CD3+': 11, 'CD4+': 13, 'CD8+': 9,
                ↪  gd_gate_name: 3},
                2: {ifng_gate_name: 1, 'CD3+': 3, 'CD4+': 7, 'CD8+': 5}
            }

            # The gates we want to extract data from
            for item in ics_gates:
                parent, gate = item.split('/')
                # The gate structure
                for gate_id, gate_path in wsp.get_gate_ids(sample_id):
                    # if the gate id is a gate of interest, the last item in the
                    ↪  gate path is the parent
                    # and the gate is a descendant of the CD3+ gate
                    if gate_id == gate and gate_path[-1] == parent:
                        # extract_data function
                        extract_data(gate, gate_path, gate_pct, mfi_comp,
                        ↪  parent, sample, sample_id,
                                     sample_results, wsp, labels_dict_ics)
        elif pre == 'Tcell':
            # gate aliases checking for LAG-3 gate
            lag3_gate_name = ''
            for alias in lag3_gate_aliases:
                if alias in hierarchy:
                    lag3_gate_name = alias
                    break
```

```
502
503             # gate aliases checking for TCM gate
504             tcm_gate_name = ''
505             for alias in tcm_gate_aliases:
506                 if alias in hierarchy:
507                     tcm_gate_name = alias
508                     break
509
510             # gate aliases checking for TEM gate
511             tem_gate_name = ''
512             for alias in tem_gate_aliases:
513                 if alias in hierarchy:
514                     tem_gate_name = alias
515                     break
516
517             # gate aliases checking for Teff gate
518             teff_gate_name = ''
519             for alias in teff_gate_aliases:
520                 if alias in hierarchy:
521                     teff_gate_name = alias
522                     break
523
524             # gate aliases checking for Precursor gate
525             precursor_gate_name = ''
526             for alias in precursor_gate_aliases:
527                 if alias in hierarchy:
528                     precursor_gate_name = alias
529                     break
530
531             if lag3_gate_name.endswith('+'):
532                 lag3_gate_name_plus = lag3_gate_name[:-1]
533             else:
534                 lag3_gate_name_plus = lag3_gate_name
535
536             labels_dict_tcell = {
537                 1: {'CD45RA': 0, 'CD57': 1, 'CD45': 2, precursor_gate_name: 3,
                    ↪  lag3_gate_name_plus.strip(): 4,
538                     'CD95': 6, 'CD4': 7, 'CD8': 8,
539                     'CD3': 10, 'CCR7': 11, 'CD28': 12, 'CD27': 13, 'PD-1': 14},
540                 2: {'CD103': 1, 'CD39': 3, 'GD': 5, 'CCR7': 7, 'CD69': 9,
                    ↪  '4-1BB': 11, 'CD45RA': 13, 'CD3': 15,
541                     'CD4': 17, 'CD8': 19, precursor_gate_name: 21,
                        ↪  lag3_gate_name_plus.strip(): 23, 'CD95': 25},
542             }
543
544             if analysis == 'PBMCs':
```

```
                # Check if the CD4+ gate (alone) is in the gating strategy.
                # If it is not, it is joined with the CD8+ gate
                cd4_in = False
                for gate, _ in wsp.get_gate_ids(sample_id):
                    if gate == 'CD4+':
                        cd4_in = True
                        break

                if not cd4_in:
                    tcell_pbmc_gates = [f'CD4+ CD8+/{precursor_gate_name}',
                    ↪  f'CD4+ CD8+/{tcm_gate_name}',
                                        f'CD4+ CD8+/{tem_gate_name}', f'CD4+
                                        ↪  CD8+/{teff_gate_name}',
                                        'CD4+ CD8+/CD95+', f'CD4+
                                        ↪  CD8+/{lag3_gate_name}', 'CD4+
                                        ↪  CD8+/CD103+',
                                        f'GD+/{precursor_gate_name}',
                                        ↪  f'GD+/{tcm_gate_name}',
                                        ↪  f'GD+/{tem_gate_name}',
                                        f'GD+/{teff_gate_name}', 'GD+/CD95+',
                                        ↪  f'GD+/{lag3_gate_name}',
                                        ↪  'GD+/CD103+']
                else:
                    tcell_pbmc_gates = [f'CD4+/{tcm_gate_name}',
                                        f'CD4+/{tem_gate_name}',
                                        ↪  f'CD4+/{teff_gate_name}',
                                        'CD4+/CD95+', f'CD4+/{lag3_gate_name}',
                                        ↪  'CD4+/CD103+',
                                        f'CD8+/{tcm_gate_name}',
                                        f'CD8+/{tem_gate_name}',
                                        ↪  f'CD8+/{teff_gate_name}',
                                        'CD8+/CD95+', f'CD8+/{lag3_gate_name}',
                                        ↪  'CD8+/CD103+',
                                        ↪  f'GD+/{tcm_gate_name}',
                                        f'GD+/{tem_gate_name}',
                                        f'GD+/{teff_gate_name}', 'GD+/CD95+',
                                        ↪  f'GD+/{lag3_gate_name}',
                                        ↪  'GD+/CD103+']

                # The gates we want to extract data from
                for item in tcell_pbmc_gates:
                    parent, gate = item.split('/')
                    # The gate structure
                    for gate_id, gate_path in wsp.get_gate_ids(sample_id):
                        # if the gate id is a gate of interest, the last item in
                        ↪  the gate path is the parent
```

```
575                         # and the gate is a descendant of the CD3+ gate
576                         if gate in gate_id and gate_path[-1] == parent and
                            ↪  'CD3+' in gate_path \
577                                 or str(gate) == 'CD39- CD69-':
578                             extract_data(gate, gate_path, gate_pct, mfi_comp,
                                ↪  parent, sample,
579                                          sample_id, sample_results, wsp,
                                             ↪  labels_dict_tcell)
580
581             elif analysis == 'TILs':
582                 # The gate structure
583                 for gate, gate_path in wsp.get_gate_ids(sample_id):
584                     # Use everything except (CD4+ CD8+), all gates that have 2
                        ↪  minus (-) signs
585                     # and all gates with no child gates
586                     if gate != 'CD4+ CD8+' and 'CD4+ CD8+' not in gate_path and
                        ↪  str(gate).count('-') != 2 \
587                             and not wsp.get_child_gate_ids(sample_id, gate,
                                ↪  gate_path) or str(gate) == 'CD39- CD69-':
588                         parent = gate_path[-1]
589                         extract_data(gate, gate_path, gate_pct, mfi_comp,
                            ↪  parent, sample,
590                                      sample_id, sample_results, wsp,
                                         ↪  labels_dict_tcell)
591
592         elif pre == 'Thoming':
593
594             # gate aliases checking for Precursor gate
595             precursor_gate_name = ''
596             for alias in precursor_gate_aliases:
597                 if alias in hierarchy:
598                     precursor_gate_name = alias
599                     break
600
601             labels_dict_thoming = {
602                 1: {'CD103': 0, 'CCR4': 1, 'CD4': 2, 'CD8': 3,
                    ↪  precursor_gate_name: 4, 'CXCR3': 5, 'CD95': 6,
603                     'CCR9': 7, 'CD45RA': 8, 'CD3': 10, 'CCR7': 11, 'CCR6': 12},
604                 2: {'CD103': 1, 'CCR4': 3, gd_gate_name: 5, 'CCR7': 7, 'CCR6':
                    ↪  9, 'CCR9': 11, 'CD45RA': 13, 'CD3': 15,
605                     'CD4': 17, 'CD8': 19, precursor_gate_name: 21, 'CXCR3': 23,
                        ↪  'CD95': 25},
606                 3: {'CD103': 0, 'CD4': 1, 'CD8': 2, precursor_gate_name: 3,
                    ↪  'CXCR3': 4, 'CD95': 6, 'CCR9': 7,
607                     'CD45RA': 8, 'CD3': 10, 'CCR7': 11, 'CCR4': 12, 'CCR6': 13,
                        ↪  gd_gate_name: 14},
```

```
608                 4: {'CD103': 1, 'CCR4': 3, 'CCR7': 5, gd_gate_name: 7, 'CCR6':
                    ↪  9, 'CCR9': 11, 'CD45RA': 13,
609                     'CD3': 15, 'CD4': 17, 'CD8': 19, precursor_gate_name: 21,
                        ↪  'CXCR3': 23, 'CD95': 25}
610             }
611
612             # The gate structure
613             for gate, gate_path in wsp.get_gate_ids(sample_id):
614                 # Check for the gates of interest
615                 if thoming_gate_of_interest(gate, gate_path):
616                     extract_data(gate.strip(), gate_path, gate_pct, mfi_comp,
                        ↪  gate_path[-1], sample,
617                                  sample_id, sample_results, wsp,
                                     ↪  labels_dict_thoming)
618
619         print(mfi_comp)
620         print(gate_pct)
621
622         # Cut off the file type extension
623         sample_id_path = sample_id[: -4]
624
625         # Create the directory structure
626         if not
            ↪  os.path.exists(f'Samples/{pre}_{analysis}/{sample_id_path}_{date}'):
627             os.makedirs(f'Samples/{pre}_{analysis}/{sample_id_path}_{date}')
628
629         # Build directed graph to extract gate hierarchy
630         dot = graphviz.Digraph(f'{sample_id_path}_gate_hierarchy', strict=True)
631         for gate, gate_path in wsp.get_gate_ids(sample_id):
632             if ':' in gate:
633                 gate = gate.replace(':', '')
634             dot.node(gate)
635             for i in range(len(gate_path)):
636                 if i == len(gate_path) - 1:
637                     dot.edge(gate_path[i], gate)
638                 else:
639                     dot.edge(gate_path[i], gate_path[i + 1])
640         dot.format = 'png'
641         dot.render(directory=f'Samples/{pre}_{analysis}
642         /{sample_id_path}_{date}') \
643             .replace('\\', '/')
644
645         # if the mfi dataframe is not empty (i.e. we found gates of interest)
646         if mfi_comp:
647             with open(f'{os.getcwd()}/Samples
648             /{pre}_{analysis}/{sample_id_path}_{date}/{sample_id_path}_mfi.csv',
                ↪  'w',
```

```
                        newline='\n') as f:
                writer = csv.DictWriter(f, fieldnames=mfi_comp.keys())
                writer.writeheader()
                writer.writerow(mfi_comp)

            with open(f'{os.getcwd()}/Samples/{pre}_{analysis}/global_mfi.csv',
            ↪  'a', newline='\n') as f_object:
                writer_object = csv.DictWriter(f_object,
                ↪  fieldnames=mfi_comp.keys())
                writer_object.writeheader()
                writer_object.writerow(mfi_comp)

        # if the gate percentages dictionary is not empty (i.e. we found gates
        ↪  of interest)
        if gate_pct:
            with open(f'Samples/{pre}_{analysis}/{sample_id_path}_{date}
            /{sample_id_path}_gate_percentages.csv', 'w',
                        newline='\n') as f:
                writer = csv.DictWriter(f, fieldnames=gate_pct.keys())
                writer.writeheader()
                writer.writerow(gate_pct)

            with open(f'{os.getcwd()}/Samples/{pre}_{analysis}
            /global_gate_pct.csv', 'a', newline='\n') as f_object:
                writer_object = csv.DictWriter(f_object,
                ↪  fieldnames=gate_pct.keys())
                writer_object.writeheader()
                writer_object.writerow(gate_pct)
        print("#####################################################")


# ------------------------------------------------------------------------
# Execution starts here

# base directory where the files are stored
base_dir = "Data/DATA_Raw_files/FLOW"

# All gate aliases found
gd_gate_aliases = ['Gamma delta + ', 'GD+', 'TCRgd+', 'TCR gd+', 'gd+']
ifng_gate_aliases = ['INFg+', 'IFN-g+']
lag3_gate_aliases = ['LAG3+', 'LAG3', 'LAG-3 +', 'Lag3+']
tcm_gate_aliases = ['TCM CD45RA- CCR7+', 'TCM']
teff_gate_aliases = ['Teff CD45RA+ CCR7-', 'Teff', 'TEFF']
tem_gate_aliases = ['TEM CD45RA- CCR7-', 'TEM']
precursor_gate_aliases = ['precursors', 'precursor', 'Precursor',
'PRECURSOR', 'Naive CD45RA+ CCR7+', 'Live Dead',
```

```
                          'Live-Dead']

if __name__ == '__main__':

    if os.path.exists(f'{os.getcwd()}/Samples'):
        shutil.rmtree(f'{os.getcwd()}/Samples', ignore_errors=True)

    os.mkdir('Samples')

    wsp_files = get_wsp_files(base_dir)

    # Loop through workspace files
    for wsp_file in wsp_files:

        analysis = ''
        if 'PBMC' in wsp_file:
            analysis = 'PBMCs'
        elif 'TIL' in wsp_file:
            analysis = 'TILs'

        pre = ''
        if 'ICS' in wsp_file:
            pre = 'ICS'
        elif 'Tcell' in wsp_file:
            pre = 'Tcell'
        elif 'Thoming' in wsp_file:
            pre = 'Thoming'

        analyze(wsp_file, analysis, pre)
```

Python Script: gate_pct_graphs.py

```
import numpy as np
from matplotlib import pyplot as plt
from matplotlib.lines import Line2D

import global_func
import statistics


def get_comp_data(gate_list, group):
    avg = {}
    print(gate_list)
    print(group)
```

```
13      get_group_avg(avg, gate_list, group)
14      print(avg)
15      return avg
16
17
18  def get_group_avg(avg, gate_list, group):
19      for gate in gate_list:
20          data_til = global_func.get_gate_data(gate, 'FLOW', group, 'TILs',
            ↪  'gate_pct')
21          data_pbmc = global_func.get_gate_data(gate, 'FLOW', group, 'PBMCs',
            ↪  'gate_pct')
22          print(data_til)
23          print(data_pbmc)
24
25          avg[gate] = {'TILs': {'mean': 0, 'std': 0}, 'PBMCs': {'mean': 0, 'std':
            ↪  0}}
26
27          if data_til:
28              data_values = [float(value) for dict_item in data_til for value in
                ↪  list(dict_item.values())]
29              avg[gate]['TILs']['mean'] = round(statistics.mean(data_values), 3)
30              avg[gate]['TILs']['std'] = round(statistics.stdev(data_values), 3)
31          if data_pbmc:
32              data_values = [float(value) for dict_item in data_pbmc for value in
                ↪  list(dict_item.values())]
33              avg[gate]['PBMCs']['mean'] = round(statistics.mean(data_values), 3)
34              avg[gate]['PBMCs']['std'] = round(statistics.stdev(data_values), 3)
35
36
37  ics_gate_list = ['CD4+', 'CD8+', 'gd+']
38  tcell_gate_list = ['precursor', 'TCM', 'TEM', 'TEFF', 'CD103+', 'CD95+', 'CD39+
    ↪  CD69+', 'CD39- CD69-']
39  thoming_gate_list = ['Th1', 'Th2', 'Th17', 'Tscm']
40
41  ics_data = get_comp_data(ics_gate_list, 'ICS')
42  tcell_data = get_comp_data(tcell_gate_list, 'Tcell')
43  thoming_data = get_comp_data(thoming_gate_list, 'Thoming')
44
45  final_data = {'ICS': ics_data, 'Tcell': tcell_data, 'Thoming': thoming_data}
46
47  for key, dataset in final_data.items():
48      print(dataset)
49
50      _, ax = plt.subplots(figsize=(16, 9))
51
52      if key == 'ICS':
```

```
        plt.title(f'{key} IFN- ', fontsize=30)
    else:
        plt.title(f'{key}', fontsize=30)

    legend = ['PBMCs', 'TILs']
    custom_lines = [Line2D([0], [0], color="cyan", lw=6),
                    Line2D([0], [0], color="purple", lw=6)]

    ax.legend(custom_lines, legend, loc='best', fontsize=15)

    gates = list(dataset.keys())
    if key == 'Tcell':
        gates = ['CD8+ | ' + gate for gate in gates]
    elif key == 'Thoming':
        gates = ['CD4+ | ' + gate if gate.lower() != 'tscm' else 'CD8+ | ' +
        ↪  gate for gate in gates]

    x = np.arange(len(dataset))
    width = 0.4

    pbmcs_avg = []
    pbmcs_std = []
    tils_avg = []
    tils_std = []
    for values in list(dataset.values()):
        pbmcs_avg.append(values['PBMCs']['mean'])
        pbmcs_std.append(values['PBMCs']['std'])
        tils_avg.append(values['TILs']['mean'])
        tils_std.append(values['TILs']['std'])

    ax.bar(x - 0.2, pbmcs_avg, width, color='cyan')
    ax.bar(x + 0.2, tils_avg, width, color='purple')

    all_std = pbmcs_std + tils_std
    iterator = iter(all_std)

    for p in ax.patches:
        rect_x = p.get_x()  # get the bottom left x corner of the bar
        rect_y = p.get_y()
        w = p.get_width()  # get width of bar
        h = p.get_height()  # get height of bar
        if h > 0:
            rect_std = next(iterator)
            min_y = h - rect_std  # use h to get min from dict z
            max_y = h + rect_std  # use h to get max from dict z
            plt.vlines(rect_x + w / 2, min_y, max_y, color='k')  # draw a
            ↪  vertical line
```

```
98              plt.hlines(max_y, (rect_x + w / 2) - w / 4, (rect_x + w / 2) + w /
                ↪  4, color='k')
99              plt.hlines(min_y, (rect_x + w / 2) - w / 4, (rect_x + w / 2) + w /
                ↪  4, color='k')
100         else:
101             plt.text(rect_x + w / 4, rect_y, "n.a.", fontsize=15)
102
103     # plt.xlabel("Gates", fontsize=20)
104     plt.ylabel("Gate Percentages", fontsize=20)
105
106     plt.xticks(x, gates, rotation=30, ha="center", fontsize=20)
107     plt.yticks(fontsize=20)
108     plt.subplots_adjust(bottom=0.25)
109
110     plt.savefig(f'Samples/{key}_gate_percentages.png')
111     plt.clf()
```

Python Script: global_func.py

```
1   import os
2
3
4   # Functions
5
6   def read_csv(csv_file):
7       """
8           Read the data from a csv and transform it into a dictionary
9       :param csv_file: the csv file path
10      :return: the data from the csv file path, in a dictionary
11      """
12      csv_data = []
13      with open(csv_file, 'r') as f:
14          lines = f.read().splitlines()
15          for line in lines:
16              idx = lines.index(line)
17              if idx % 2 == 1:
18                  if 'mfi' in csv_file and 'ICS' not in csv_file:
19                      values_list = [x for x in line.split('\"') if x.strip() !=
                        ↪  '' and x != ',']
20                      copy_values_list = values_list[:]
21                      for item in copy_values_list:
22                          if item.startswith(',') or item.endswith(','):
23                              item_idx = values_list.index(item)
24                              values_list.remove(item)
```

```
                            aux_list = [x for x in item.split(',') if x != '']
                            for i in range(len(aux_list)):
                                values_list.insert(item_idx + i, aux_list[i])
                    csv_data.append(dict(zip(lines[idx - 1].split(','),
                    ↪  values_list)))
                else:
                    csv_data.append(dict(zip(lines[idx - 1].split(','),
                    ↪  line.split(','))))
    return csv_data


def get_data(prefix, group, analysis, data_type):
    """
        Get the data given the conditions
    :param prefix: FLOW or ELISA
    :param group: ICS, Tcell or Thoming
    :param analysis: PBMCs, TILs or WBA
    :param data_type: gate_mfi or gate_pct
    :return: the data, as a list
    """
    if prefix.upper() == 'ELISA':
        return data[prefix][group][analysis]
    else:
        if group == '':
            result = []
            for aux_group in ['ICS', 'Tcell', 'Thoming']:
                result.extend(data[prefix][aux_group][analysis][data_type])
            return result
        return data[prefix][group][analysis][data_type]


def get_gate_data(gate, prefix, group, analysis, data_type):
    """
        Get data from a specific gate
    :param gate: the gate
    :param prefix: FLOW or ELISA
    :param group: ICS, Tcell or Thoming
    :param analysis: PBMCs or TILs
    :param data_type: gate_mfi or gate_pct
    :return: the data, as a list
    """
    res_data = []
    gate_data = get_data(prefix, group, analysis, data_type)
    for dict_item in gate_data:
        for key, value in dict_item.items():
            if gate.lower() in key.lower() or gate in key.upper():
```

```
                res_data.append({key: value})
    return res_data


#
↪  ----------------------------------------------------------------------------
# Execution starts here

# Import Data

# Build data structure
data = {'ELISA': {'Concentrations': {'TIL': [], 'WBA': []}, 'OD': {'TIL': [],
↪  'WBA': []},
                  'Reactions': {'TIL': [], 'WBA': []}},
        'FLOW': {'ICS': {'PBMCs': {'gate_mfi': {}, 'gate_pct': {}}, 'TILs':
        ↪  {'gate_mfi': {}, 'gate_pct': {}}},
                 'Tcell': {'PBMCs': {'gate_mfi': {}, 'gate_pct': {}}, 'TILs':
                 ↪  {'gate_mfi': {}, 'gate_pct': {}}},
                 'Thoming': {'PBMCs': {'gate_mfi': {}, 'gate_pct': {}}, 'TILs':
                 ↪  {'gate_mfi': {}, 'gate_pct': {}}}
                 }
        }

# Import ELISA data
data['ELISA']['Concentrations']['TIL'] =
↪  read_csv('Patients/TIL_global_concentrations.csv')
data['ELISA']['Concentrations']['WBA'] =
↪  read_csv('Patients/WBA_global_concentrations.csv')
data['ELISA']['OD']['TIL'] = read_csv('Patients/TIL_global_ODs.csv')
data['ELISA']['OD']['WBA'] = read_csv('Patients/WBA_global_ODs.csv')
data['ELISA']['Reactions']['TIL'] =
↪  read_csv('Patients/TIL_global_peptide_reactions.csv')
data['ELISA']['Reactions']['WBA'] =
↪  read_csv('Patients/WBA_global_peptide_reactions.csv')

# Import FLOW data
for directory in os.listdir('Samples'):
    if os.path.isdir(os.path.join('Samples', directory)):
        for item in os.listdir(os.path.join('Samples', directory)):
            path = os.path.join('Samples', directory, item)
            if os.path.isfile(path):
                if 'ICS_PBMCs' in path:
                    if 'mfi' in path:
                        data['FLOW']['ICS']['PBMCs']['gate_mfi'] =
                        ↪  read_csv(path)
                    elif 'gate_pct' in path:
```

```
                        data['FLOW']['ICS']['PBMCs']['gate_pct'] =
                        ↪ read_csv(path)
                elif 'ICS_TILs' in path:
                    if 'mfi' in path:
                        data['FLOW']['ICS']['TILs']['gate_mfi'] = read_csv(path)
                    elif 'gate_pct' in path:
                        data['FLOW']['ICS']['TILs']['gate_pct'] = read_csv(path)
                elif 'Tcell_PBMCs' in path:
                    if 'mfi' in path:
                        data['FLOW']['Tcell']['PBMCs']['gate_mfi'] =
                        ↪ read_csv(path)
                    elif 'gate_pct' in path:
                        data['FLOW']['Tcell']['PBMCs']['gate_pct'] =
                        ↪ read_csv(path)
                elif 'Tcell_TILs' in path:
                    if 'mfi' in path:
                        data['FLOW']['Tcell']['TILs']['gate_mfi'] =
                        ↪ read_csv(path)
                    elif 'gate_pct' in path:
                        data['FLOW']['Tcell']['TILs']['gate_pct'] =
                        ↪ read_csv(path)
                elif 'Thoming_PBMCs' in path:
                    if 'mfi' in path:
                        data['FLOW']['Thoming']['PBMCs']['gate_mfi'] =
                        ↪ read_csv(path)
                    elif 'gate_pct' in path:
                        data['FLOW']['Thoming']['PBMCs']['gate_pct'] =
                        ↪ read_csv(path)
                elif 'Thoming_TILs' in path:
                    if 'mfi' in path:
                        data['FLOW']['Thoming']['TILs']['gate_mfi'] =
                        ↪ read_csv(path)
                    elif 'gate_pct' in path:
                        data['FLOW']['Thoming']['TILs']['gate_pct'] =
                        ↪ read_csv(path)
```

Python Script: linreg.py

```
from scipy.stats import linregress
import numpy as np
from global_func import get_data, get_gate_data
import matplotlib.pyplot as plt
from scipy.stats import t
import os
```

```
7  import csv
8
9
10 # Functions
11
12 def get_avg(peptide, data):
13     """
14         Get the average mfi for a peptide in the data provided
15     :param peptide: the peptide string
16     :param data: the data to be checked
17     :return: the average mfi of the peptide in the data list
18     """
19     res = []
20     for dict_item in data:
21         if peptide in dict_item:
22             res.append(float(dict_item[peptide]))
23     return round(sum(res) / len(res), 3)
24
25
26 def perform_linear_regression(data_min, data_max, elisa_data, analysis, gate,
   ↪  elisa_type):
27     """
28         Performs linear regression with the given data
29
30     :param data_min: the minimum value of the gate data
31     :param data_max: the maximum value of the gate data
32     :param elisa_data: the elisa data
33     :param analysis: analysis string (PBMCs or TILs)
34     :param gate: the gate (CD4, CD8, GD)
35     :param elisa_type: the ELISA analysis type (TIL or WBA)
36     :return: None
37     """
38     reg_array = np.random.uniform(low=float(data_min), high=float(data_max),
       ↪  size=(len(elisa_data),))
39     res = linregress(reg_array, elisa_data)
40
41     print(f'R-squared: {res.rvalue**2:.6f}')
42
43     tinv = lambda p, df: abs(t.ppf(p / 2, df))
44     ts = tinv(0.05, len(reg_array) - 2)
45
46     print(f"slope (95%): {res.slope:.6f} +/- {ts * res.stderr:.6f}")
47     print(f"intercept (95%): {res.intercept:.6f}"
48           f" +/- {ts * res.intercept_stderr:.6f}")
49
50     extracted_data = {'R-Squared': round(res.rvalue**2, 6),
```

```
                        'Slope': f'{round(res.slope, 6)} +/- {round(ts +
                        ↪  res.stderr, 6)}',
                        'Intercept': f'{round(res.intercept, 6)} +/- {round(ts *
                        ↪  res.intercept_stderr, 6)}'}

    filepath = f'LinearRegression/{gate}_IFN-g_{analysis}_vs_ELISA_{elisa_type}'
    if not os.path.exists(filepath):
        os.makedirs(filepath)

    # Write data to sample file
    with open(f'{filepath}/data.csv', 'w', newline='\n') as f:
        w = csv.writer(f)
        w.writerow(extracted_data.keys())
        w.writerow(extracted_data.values())

    plt.plot(reg_array, elisa_data, 'o', label='data points')
    plt.plot(reg_array, res.intercept + res.slope * reg_array, 'r',
    ↪  label='fitted line')

    if gate == 'GD':
        plt.title(f' + IFN- + {analysis} VS ELISA {elisa_type}', fontsize=15)
    else:
        plt.title(f'{gate}+ IFN- + {analysis} VS ELISA {elisa_type}',
        ↪  fontsize=15)

    # plt.xticks([])
    # plt.yticks([])

    if gate == 'GD':
        plt.xlabel(f'%  + IFN- + from ICS {analysis}')
    else:
        plt.xlabel(f'% {gate}+ IFN- + from ICS {analysis}')

    plt.ylabel(f'% IFN- + Supernatant Concentration from ELISA {elisa_type}',
    ↪  fontsize=8)

    plt.legend()

    plt.savefig(f'{filepath}/graph.png')
    plt.clf()

    print()


#
↪  ----------------------------------------------------------------------------
```

```
91  # Execution starts here
92
93  # If results directorydoes not exist, create it
94  if not os.path.exists('LinearRegression'):
95      os.makedirs('LinearRegression')
96
97  # get elisa data
98  elisa_wba = get_data('ELISA', 'Reactions', 'WBA', '')
99  elisa_til = get_data('ELISA', 'Reactions', 'TIL', '')
100
101 print(f'Elisa WBA: {elisa_wba}')
102 print(f'Elisa TIL: {elisa_til}')
103
104 # Rebuild elisa data into a dictionary to take only the average of the peptides
    ↪  it holds
105 final_elisa_wba = {}
106 for dict_item in elisa_wba:
107     for k in dict_item.keys():
108         if k not in final_elisa_wba:
109             final_elisa_wba[k] = get_avg(k, elisa_wba)
110
111 final_elisa_til = {}
112 for dict_item in elisa_til:
113     for k in dict_item.keys():
114         if k not in final_elisa_til:
115             final_elisa_til[k] = get_avg(k, elisa_til)
116
117 print(f'Elisa WBA Values: {list(final_elisa_wba.values())}')
118 print(f'Elisa TIL Values: {list(final_elisa_til.values())}')
119 print()
120
121 # PBMCs gate percentages vs ELISA WBA
122 # CD8
123 cd8_pbmcs_data = [float(list(entry.values())[0]) for entry in
    ↪  get_gate_data('cd8', 'FLOW', 'ICS', 'PBMCs', 'gate_pct')]
124 perform_linear_regression(min(cd8_pbmcs_data), max(cd8_pbmcs_data),
    ↪  list(final_elisa_wba.values()), 'PBMCs', 'CD8', 'WBA')
125
126 # CD4
127 cd4_pbmcs_data = [list(entry.values())[0] for entry in get_gate_data('cd4',
    ↪  'FLOW', 'ICS', 'PBMCs', 'gate_pct')]
128 perform_linear_regression(min(cd4_pbmcs_data), max(cd4_pbmcs_data),
    ↪  list(final_elisa_wba.values()), 'PBMCs', 'CD4', 'WBA')
129 # GD
130 gd_pbmcs_data = [list(entry.values())[0] for entry in get_gate_data('gd',
    ↪  'FLOW', 'ICS', 'PBMCs', 'gate_pct')]
```

```
perform_linear_regression(min(gd_pbmcs_data), max(gd_pbmcs_data), list(final_elisa_wba.values()), 'PBMCs', 'GD', 'WBA')

# TILs gate percentages vs ELISA TIL
# CD8
cd8_tils_data = [list(entry.values())[0] for entry in get_gate_data('cd8', 'FLOW', 'ICS', 'TILs', 'gate_pct')]
perform_linear_regression(min(cd8_tils_data), max(cd8_tils_data), list(final_elisa_til.values()), 'TILs', 'CD8', 'TIL')

# CD4
cd4_tils_data = [list(entry.values())[0] for entry in get_gate_data('cd4', 'FLOW', 'ICS', 'TILs', 'gate_pct')]
perform_linear_regression(min(cd4_tils_data), max(cd4_tils_data), list(final_elisa_til.values()), 'TILs', 'CD4', 'TIL')

# GD
gd_tils_data = [list(entry.values())[0] for entry in get_gate_data('gd', 'FLOW', 'ICS', 'TILs', 'gate_pct')]
perform_linear_regression(min(gd_tils_data), max(gd_tils_data), list(final_elisa_til.values()), 'TILs', 'GD', 'TIL')
```

Python Script: 1_test_chi_squared.py

```
import scipy.stats as stats
import os
from global_func import get_data, get_gate_data
from tkinter import simpledialog


# Functions


def print_comp_data(comp_str, list1, list2):
    """
            Print both lists of data

            :param comp_str: The string that describes the two data lists
            :param list1: the first list
            :param list2: the second list

            :return: None
    """
    print(comp_str)
```

```
21      print()
22      print(list1)
23      print()
24      print(list2)
25      print()
26
27
28  def perform_chi2_test(list1, list2, comp_str):
29      """
30              Perform the chi-squared contingency test with the
31              values on the two lists
32
33              :param list1: the first list
34              :param list2: the second list
35              :param comp_str: The string that describes the two data lists
36
37              :return: None
38      """
39      # Create the directory structure
40      if not os.path.exists('Stats_Tests/Chi2'):
41          os.makedirs('Stats_Tests/Chi2')
42
43      written_ind = False
44      written_dep = False
45
46      for dict_item in list1:
47          for other_dict_item in list2:
48              for key1, value1 in dict_item.items():
49                  for key2, value2 in other_dict_item.items():
50                      value1 = float(value1)
51                      value2 = float(value2)
52                      print(f'Chi-Squared test for {key1}:{value1} and
                        ↪  {key2}:{value2}')
53                      if value1 != 100 or value2 != 100:
54                          values_1 = [value1, float(100 - value1)]
55                          values_2 = [value2, float(100 - value2)]
56                          chi2, pvalue, dof, expected_freq =
                            ↪  stats.chi2_contingency([values_1, values_2])
57                          print(f'Chi-Squared: {chi2}\t\tP-Value:
                            ↪  {pvalue}\t\tDegrees of Freedom: {dof}\t\t'
58                                f'Expected Frequency: {expected_freq}')
59
60                          if pvalue <= alpha:
61                              print(f'dependent: the relation between {key1} and
                                ↪  {key2} is significant')
62                              file = open(f'Stats_Tests
```

```
                                /Chi2/results_chi2_dependent_{alpha}.csv', 'a')
                                if not written_dep:
                                    file.write(comp_str + '\n')
                                    written_dep = True

                                file.write(f'{key1} versus {key2}')
                                file.write('\n')
                                file.write(f'Chi2: {chi2}, P-Value: {pvalue}, DOF:
                                ↪  {dof}, Expected_Freq: {expected_freq}')
                                file.write('\n')
                            else:
                                print(f'independent: {key1} and {key2} do not have a
                                ↪  significant relation')
                                file = open(f'Stats_Tests
                                /Chi2/results_chi2_independent_{alpha}.csv', 'a')
                                if not written_ind:
                                    file.write(comp_str + '\n')
                                    written_ind = True

                                file.write(f'{key1} versus {key2}')
                                file.write('\n')
                                file.write(f'Chi2: {chi2}, P-Value: {pvalue}, DOF:
                                ↪  {dof}, Expected_Freq: {expected_freq}')
                                file.write('\n')
                    break
        file.write('\n')
        file.close()


# ------------------------------------------------------------------------
# Execution starts here

# Use tkinter to get user input for significance level (alpha) for p-value
↪  comparison
alpha = simpledialog.askfloat('Chi-Squared Contingency Tests',
                              'Input alpha value for p-value comparisons:')

if alpha is None or alpha < 0:
    print('ERROR: Alpha value is negative or not defined.')
    exit()

input('Press ENTER to continue...')

# If the files to be created already exist, delete them
if os.path.exists(f'Stats_Tests/Chi2/results_chi2_dependent_{alpha}.csv'):
    os.remove(f'Stats_Tests/Chi2/results_chi2_dependent_{alpha}.csv')
```

```
105 if os.path.exists(f'Stats_Tests/Chi2/results_chi2_independent_{alpha}.csv'):
106     os.remove(f'Stats_Tests/Chi2/results_chi2_independent_{alpha}.csv')
107 
108 # gate % flow_Thoming_PBMCs                 versus  gate % flow_Thoming_TILs
109 flow_thoming_pbmcs_gate_pct = get_data('FLOW', 'Thoming', 'PBMCs', 'gate_pct')
110 flow_thoming_tils_gate_pct = get_data('FLOW', 'Thoming', 'TILs', 'gate_pct')
111 print_comp_data('GATE PCT THOMING PBMCS VS GATE PCT THOMING TILS',
    ↪  flow_thoming_pbmcs_gate_pct,
112                 flow_thoming_tils_gate_pct)
113 perform_chi2_test(flow_thoming_pbmcs_gate_pct, flow_thoming_tils_gate_pct,
    ↪  'Thoming PBMCs VS Thoming TILs')
114 
115 # gate % flow_Tcell_PBMCs                 versus  %(each peptide) ELISA_WBA
116 flow_tcell_pbmcs_gate_pct = get_data('FLOW', 'Tcell', 'PBMCs', 'gate_pct')
117 elisa_wba = get_data('ELISA', 'OD', 'WBA', '')
118 print_comp_data('GATE PCT TCELL PBMCS VS ELISA', flow_tcell_pbmcs_gate_pct,
    ↪  elisa_wba)
119 perform_chi2_test(flow_tcell_pbmcs_gate_pct, elisa_wba, 'Tcell PBMCs VS ELISA
    ↪  WBA')
120 
121 # gate % flow_ICS_PBMCs                   versus  %(each peptide) ELISA_WBA
122 flow_ics_pbmcs_gate_pct = get_data('FLOW', 'ICS', 'PBMCs', 'gate_pct')
123 print_comp_data('GATE PCT ICS PBMCS VS ELISA', flow_ics_pbmcs_gate_pct,
    ↪  elisa_wba)
124 perform_chi2_test(flow_ics_pbmcs_gate_pct, elisa_wba, 'ICS PBMCs VS ELISA WBA')
125 
126 #         ICS_PBMCs_CD3_IFNgamma   COMPARE with ELISA_WBA
127 print('MFI CD3 ICS PBMCS VS ELISA')
128 print('No data.')
129 print()
130 
131 #         ICS_PBMCs_CD8_IFNgamma   COMPARE with ELISA_WBA
132 cd8_data = get_gate_data('cd8', 'FLOW', 'ICS', 'PBMCs', 'gate_pct')
133 print_comp_data('GATE PCT CD8 ICS PBMCS VS ELISA', cd8_data, elisa_wba)
134 perform_chi2_test(cd8_data, elisa_wba, 'ICS PBMCS CD8 INF-G VS ELISA WBA')
135 
136 #         ICS_PBMCs_CD4_IFNgamma   COMPARE with ELISA_WBA
137 cd4_data = get_gate_data('cd4', 'FLOW', 'ICS', 'PBMCs', 'gate_pct')
138 print_comp_data('GATE PCT CD4 ICS PBMCS VS ELISA', cd4_data, elisa_wba)
139 perform_chi2_test(cd4_data, elisa_wba, 'ICS PBMCS CD4 INF-G VS ELISA WBA')
140 
141 #         ICS_PBMCs_gd_IFNgamma    COMPARE with ELISA_WBA
142 gd_data = get_gate_data('gd', 'FLOW', 'ICS', 'PBMCs', 'gate_pct')
143 print_comp_data('GATE PCT GD ICS PBMCS VS ELISA', gd_data, elisa_wba)
144 perform_chi2_test(gd_data, elisa_wba, 'ICS PBMCS GD INF-G VS ELISA WBA')
```

Python Script: 2_test_t_test.py

```
import scipy.stats as stats
from global_func import get_data
from tkinter import simpledialog
import os


def get_average(key, th_list):
    """
        Return the average of the gate mfi for the given Thoming list

        :param key: the gate to be checked
        :param th_list: th1 or th2 list
        :return: the average of the gate specified in the th list specified
    """
    res = []
    for dict_item in th_list:
        dict_item = eval(dict_item)
        if key in dict_item:
            res.append(dict_item[key])
    return round(sum(res) / len(res), 3)


def get_values_for_key(key):
    """
        filter the averages dictionary given the key (Th1 or Th2)
    :param key: Th1 or Th2
    :return: Dictionary with PBMCs and TILs for the given key
    """
    return {k: v for k, v in averages.items() if k.startswith(key)}

# ----------------------------------------------------------------------
# Execution starts here


# Use tkinter to get user input for significance level (alpha) for p-value
↪  comparison
alpha = simpledialog.askfloat('T-Test Dependent Tests',
                              'Input alpha value for p-value comparisons:')

if alpha is None or alpha < 0:
    print('ERROR: Alpha value is negative or not defined.')
```

```
41      exit()
42
43  input('Press ENTER to continue...')
44
45  th1_gates = ['CXCR3', 'CCR6']
46  th2_gates = ['CCR4', 'CCR6']
47  all_gates = list(set(th1_gates + th2_gates))
48
49  th1_pbmcs = [dict_item['Th1'] for dict_item in get_data('FLOW', 'Thoming',
    ↪  'PBMCs', 'gate_mfi') if 'Th1' in dict_item]
50  th2_pbmcs = [dict_item['Th2'] for dict_item in get_data('FLOW', 'Thoming',
    ↪  'PBMCs', 'gate_mfi') if 'Th2' in dict_item]
51
52  th1_tils = [dict_item['Th1'] for dict_item in get_data('FLOW', 'Thoming',
    ↪  'TILs', 'gate_mfi') if 'Th1' in dict_item]
53  th2_tils = [dict_item['Th2'] for dict_item in get_data('FLOW', 'Thoming',
    ↪  'TILs', 'gate_mfi') if 'Th2' in dict_item]
54
55  # Get the averages for all gates composing each Thoming gate for the respective
    ↪  analysis type
56  averages = {'Th1_PBMCs': [get_average('CXCR3', th1_pbmcs), get_average('CCR6',
    ↪  th1_pbmcs)],
57              'Th2_PBMCs': [get_average('CCR4', th2_pbmcs), get_average('CCR6',
                ↪  th2_pbmcs)],
58              'Th1_TILs': [get_average('CXCR3', th1_tils), get_average('CCR6',
                ↪  th1_tils)],
59              'Th2_TILs': [get_average('CCR4', th2_tils), get_average('CCR6',
                ↪  th2_tils)]
60              }
61
62  print(averages)
63
64  th1_pbmcs_norm = stats.anderson(averages['Th1_PBMCs'],
    ↪  dist='norm').fit_result.success
65  th1_tils_norm = stats.anderson(averages['Th1_TILs'],
    ↪  dist='norm').fit_result.success
66
67  th2_pbmcs_norm = stats.anderson(averages['Th2_PBMCs'],
    ↪  dist='norm').fit_result.success
68  th2_tils_norm = stats.anderson(averages['Th2_TILs'],
    ↪  dist='norm').fit_result.success
69
70  if th1_pbmcs_norm and th1_tils_norm:
71      # Perform related T-test for Th1
72      th1 = get_values_for_key('Th1')
73      th1_t_stat, th1_p_value = stats.ttest_rel(th1['Th1_PBMCs'], th1['Th1_TILs'])
```

```
    print(f'Th1:\nT-stat: {th1_t_stat}\t\tP-value:{th1_p_value}')

    if th1_p_value <= alpha:
        print('Reject the null hypothesis: the averages of CXCR3 and CCR6 for
        ↪  Th1 PBMCs and Th1 TILs are different')
    else:
        print('Accept the null hypothesis: the averages of CXCR3 and CCR6 for
        ↪  Th1 PBMCs and Th1 TILs are similar')

else:
    th1_pbmcs_wilcoxon = stats.wilcoxon(averages['Th1_PBMCs'])
    print(th1_pbmcs_wilcoxon)
    th1_tils_wilcoxon = stats.wilcoxon(averages['Th1_TILs'])
    print(th1_tils_wilcoxon)


if th2_pbmcs_norm and th2_tils_norm:
    # Perform related T-test for Th2
    th2 = get_values_for_key('Th2')
    th2_t_stat, th2_p_value = stats.ttest_rel(th2['Th2_PBMCs'], th2['Th2_TILs'])
    print(f'Th2:\nT-stat: {th2_t_stat}\t\tP-value:{th2_p_value}')

    if th2_p_value <= alpha:
        print('Reject the null hypothesis: the averages of CCR4 and CCR6 for Th2
        ↪  PBMCs and Th1 TILs are different')
    else:
        print('Accept the null hypothesis: the averages of CCR4 and CCR6 for Th2
        ↪  PBMCs and Th1 TILs are similar')

else:
    th2_pbmcs_wilcoxon = stats.wilcoxon(averages['Th2_PBMCs'])
    print(th2_pbmcs_wilcoxon)
    th2_tils_wilcoxon = stats.wilcoxon(averages['Th2_TILs'])
    print(th2_tils_wilcoxon)
```

Python Script: data_export.py

```
import mysql.connector
import tkinter as tk
from tkinter import messagebox
import ttkbootstrap as ttk


# Functions

```

```
9  def query_database(query):
10     res = None
11     cursor.execute(query)
12     if query.lower().startswith('select') or query.lower().startswith('show'):
13         res = cursor.fetchall()
14     return res
15
16
17 def build_form(master_window, table):
18     cols = table_cols[table]
19     print(cols)
20     for col in cols:
21         print(f'{col} : {form_items[col]}')
22     pass
23
24
25 def build_menu(table):
26     if len(table) == 0:
27         messagebox.showerror('Table Undefined', 'Please choose a table to insert
       ↪  data!')
28     else:
29         new_window = ttk.Toplevel(root)
30         menu_frame = tk.Frame(master=new_window, background='red')
31         menu_frame.pack(fill="both", expand=True, padx=20, pady=20)
32         build_form(new_window, table, table_cols[table])
33
34
35 def change_label(menu, txt):
36     menu['text'] = txt
37
38
39 # ------------------------------------------------------------------------
40 # Execution starts here
41
42 tables = ['elisa', 'output', 'patient', 'phenotype_cytometry', 'sample',
   ↪  'tumor_biopsy',
43         'wes_whole_exosome_sequencing', 'whole_blood']
44
45 table_cols = {
46     'elisa': ('Patient_ID', 'Sample_ID', 'Protein_Cytokine_Interest',
       ↪  'ELISA_Date', 'ELISA_Plate_ID'),
47     'output': ('Output_ID', 'Patient_ID', 'Sample_ID',
       ↪  'ELISA_CytokineProtein_Conc_Peptide', 'Flow_Panel_ID',
48                'CD_Percentage', 'CD_MFI', 'CytokineProtein_Percentage',
                   ↪  'CytokineProtein_MFI'),
49     'patient': ('Patient_ID', 'Age', 'Sex'),
```

```
    'phenotype_cytometry': ('Flow_Panel_ID', 'Patitent_ID', 'Sample_ID'),
    'sample': ('Sample_ID', 'Patient_ID', 'Tumor_Biopsy_ID', 'Whole_Blood_ID',
    ↪ 'Tissue_Type', 'Sample_Date'),
    'tumor_biopsy': ('Tumor_Biopsy_ID', 'WES_Whole_Exosome_Sequencing',
    ↪ 'Biopsy_Location', 'Tissue_Type',
                     'Pathology_Diagnosis'),
    'wes_whole_exosome_sequencing': ('Artificial_Peptides_from',
    ↪ 'Biopsy_Location', 'Tissue_Type', 'Pathology_Diagnosis'),
    'whole_blood': ('Whole_Blood_ID', 'Heparin', 'EDTA', 'Serum', 'LRS_chamber',
    ↪ 'Buffy_Coat')
}

flow_panel_id = ['ICS', 'T-cell panel', 'Thoming']

pathology_diagnosis = ['Intrahepatic Cholangiocarcinoma', 'Metastatic Colon
↪ Adenocarcinoma',
                       'Colon Adenocarcinoma', 'Colon Neuroendocrine Tumor',
                       ↪ 'Rectum Adenocarcinoma',
                       'Pancreas Neuroendocrine Tumor', 'Pancreas
                       ↪ Adenocarcinoma', 'Retroperitoneum Liposarcoma',
                       'Metastatic Melanoma possibly, uveal melanoma',
                       'Glioblastoma', 'Ductal Adenocarcinoma',
                       'Ampullary Adenomyomatous Hyperplasia',
                       'Stomach Gastrointestinal Stromal Tumour',
                       'Billiary Tract Adenocarcinoma']


def get_samples():
    with open('Samples/sample_ids.txt', 'r') as f:
        lines = f.readlines()
        lines = set(list(map(lambda x: x[:-1], lines)))
        return lines


def get_peptides():
    with open('Patients/peptides.txt', 'r') as f:
        lines = f.readlines()
        lines = set(list(map(lambda x: x[:-1], lines)))
        return lines


form_items = {
    'Patient_ID': {'label': 'Select Patient:', 'data_type': 'dropdown',
    ↪ 'dropdown_list': [1, 2, 3, 4, 5]},
    'Sample_ID': {'label': 'Select Sample:', 'data_type': 'dropdown',
    ↪ 'dropdown_list': get_samples()},
```

```
87      'Protein_Cytokine_Interest': {'label': 'Select Cytokine:', 'data_type':
        ↪  'str'},
88      'ELISA_Date': {'label': 'Select Date:', 'data_type': 'date'},
89      'ELISA_Plate_ID': {'label': 'Select Plate:', 'data_type': 'int'},
90      'Output_ID': {'label': '', 'data_type': 'inc_int'},
91      'ELISA_CytokineProtein_Conc_Peptide': {'label': 'Select Peptide',
        ↪  'data_type': 'dropdown', 'dropdown_list': get_peptides()},
92      'Flow_Panel_ID': {'label': 'Select Flow Panel ID:', 'data_type': 'dropdown',
        ↪  'dropdown_list': flow_panel_id},
93      'Pathology_Diagnosis': {'label': 'Select Pathology Diagnosis:', 'data_type':
        ↪  'dropdown', 'dropdown_list': pathology_diagnosis},
94  }
95
96  root = ttk.Window(themename='superhero')
97  root.title("User Input")
98  root.resizable(False, False)
99
100 main_frame = ttk.Frame(master=root)
101 main_frame.pack(fill="both", expand=True, padx=20, pady=20)
102
103 table_var_label = ttk.StringVar()
104
105 table_label = ttk.Label(main_frame, textvariable=table_var_label)
106 table_var_label.set('Table Name')
107 table_label.grid(row=0, column=1, padx=5)
108
109 # Flow Panel menu
110 clicked_table_var = ttk.StringVar()
111
112 dropdown_table = ttk.Menubutton(main_frame, bootstyle='primary', text='Pick the
    ↪  Table')
113 dropdown_table.grid(row=0, column=2)
114
115 inside_dropdown_table = ttk.Menu(dropdown_table)
116 for x in tables:
117     inside_dropdown_table.add_radiobutton(label=x, variable=clicked_table_var,
118                                           command=lambda x=x:
                                              ↪  change_label(dropdown_table, x))
119
120 dropdown_table['menu'] = inside_dropdown_table
121
122 next_button = ttk.Button(root, text="Next", command=lambda:
    ↪  build_menu(clicked_table_var.get()),
123                          bootstyle='success, outline')
124 next_button.pack(pady=10)
125
```

```
126 root.mainloop()
127
128 # # Pathology Diagnosis menu
129 # clicked_pathology_diagnosis = ttk.StringVar()
130 #
131 # dropdown_path_diag = ttk.Menubutton(root, bootstyle='primary', text='Pick the
    ↪  Pathology Diagnosis')
132 # dropdown_path_diag.pack(pady=10)
133 #
134 # inside_dropdown_path_diag = ttk.Menu(dropdown_path_diag)
135 # for x in pathology_diagnosis:
136 #     inside_dropdown_path_diag.add_radiobutton(label=x,
    ↪  variable=clicked_pathology_diagnosis,
137 #     command=lambda x=x: change_label(dropdown_path_diag, x))
138 #
139 # dropdown_path_diag['menu'] = inside_dropdown_path_diag
140 #
141 # # Button for closing
142 # exit_button = ttk.Button(root, text="Save & Exit", command=root.destroy,
    ↪  bootstyle='success, outline')
143 # exit_button.pack(pady=10, side=BOTTOM)
144
145 conn = mysql.connector.connect(
146     host='localhost',
147     user='root',
148     password='root',
149     database='biomedicaldb'
150 )
151
152 cursor = conn.cursor()
153
154 # tables = [table[0] for table in query_database('SHOW TABLES')]
155 # print(tables)
156 #
157 # for table in tables:
158 #     entries = query_database(f'SELECT * FROM biomedicaldb.{table}')
159 #     print(f'{table} - {cursor.column_names}')
160
```

Key T-Helper CD Markers and correlated Cytokines

Table C.1: Markers for $CD4^+$ T-cells

T-cell sub-set	Main characteristic	Cell surface markers	Additional Marker 1	Additional Marker 2	Additional Marker 3	Additional Marker 4	Major transcription factor(s)	Major cytokines
Th1	pro-inflammatory	CXCR3+	CCR6-	CCR4-			T-bet, STAT1, STAT4	IFN-γ, TNF-α, IL-2
Th2	anti-inflammatory	CCR4+	CCR6-	CXCR3-			GATA3, STAT6, c-MAF	IL-4, IL-13, IL-5, IL-10
Th17	pro-inflammatory	CCR4+	CCR6+	CD161+	CXCR3-		STAT3, RORγT, RORα	IL-17A, IL-17F, IL-22
Treg	anti-inflammatory	CD39+	CD73+	CD103+	GARP-LAP+	GITR+	FoxP3, STAT5, Smad3	TGFβ, IL-10, IL-35

D

Human Tumor Antigen Peptides and Controls for Co-Culture

Table D.1: Example of Human Tumor Antigens

Antigen Class	Antigen Example	Antigen Description	Associated Tumor
i) Germline	MAGE, CAGE, BAGE, NY-ESO-1	Normally expressed only in tests/placenta fetal cells, but re-expressed in cancer cells	Melanoma and other tumors
ii) Onco-fetal	CEA, AFP	Expressed by tumors and also by fetal tissues but they are produced in much lower concentration by adult tissues.	Hepatoma and other tumors
iii) Lineage-restricted differentiation	gp100, MART-1, tyrosinase, TRP-1, PSA, PAP, PSM, PSMA, Proteinase-3	Normally only seen at particular phases of differentiation of a cell type	Melanoma, Prostate Cancer, Myeloid Leukemia
iv) Overexpression	MUC1, CEA, Survivin, HER-2/neu	Present in normal tissues, but with an increased expression in cancer	Breast Cancer and Ovarian Cancer
v) Mutated molecules or Translocation products	β-catenin, CDK4, MART-2, MUM3, BCR-ABL, TEL/AML-1, Ras, p53	Arising from specific mutations	Melanoma, Leukemia, various Carcinomas
vi) Viral	EBV-EBNA, HTVL-1, Tax, HPV 16-E7	Viral antigens expressed in virally induced cancers	Lymphoma, Adult T-cell Leukaemia, Cervical Cancer
vii) Idiotypic	surface immunoglobulin	Located in the variable regions of antibodies	Lymphoma, Myeloma

Table D.2: List of controls used for co-culture assays

Vial	Peptide Sequence	Information	Length	Concentration - Stock	Concentration - Used
PHA	N/A	Cat. No. L8902-25mg	N/A	2 mg/mL	5 µg/mL
OKT3	N/A	Cat. No. 317302	N/A	0.5 mg/mL	5 µg/mL
M1 (mix)	N/A	Peptide Pool of 61 15-mers with 11 aa overlap, Matrix protein 1 (UniProt ID: B4UPA8)	N/A	2 mg/mL	1 µg/mL
Haemagglutinin	VEPGDKITFEATGNL	Wildtype peptide	15	1.5 mg/mL	1 µg/mL
EBNA	N/A	Peptide Pool of 234 15-mers with 11 aa overlap, EBNA 3 (UniProt ID: P12977)	N/A	2 mg/mL	1 µg/mL
CMV	N/A	Peptide Pool of 138 15-mers with 11 aa overlap, pp65 (UniProt ID: P06725)	N/A	2 mg/mL	1 µg/mL
ESAT6	ELNNALQNL	Wildtype peptide from ESAT-6, Mycobacterium tuberculosis	9	2 mg/mL	1 µg/mL
EGFR	LEEKKGNYVVTDH	Wildtype peptide	13	1.5 mg/mL	1 µg/mL

Table D.3: List of peptides used for co-culture assays

vial	peptide sequence	gene	Length	information	Concentration - Stock	Concentration - Used
KRAS_P1 (WT)	YKLVVVGAGGVGKSALT	KRAS	17	wildtype peptide with position G12 in the middle with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P2	YKLVVVGAVGVGKSALT	KRAS	17	G12V mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P3	YKLVVVGADGVGKSALT	KRAS	17	G12D mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P4	YKLVVVGARGVGKSALT	KRAS	17	G12R mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P5	YKLVVVGASGVGKSALT	KRAS	17	G12S mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P6	YKLVVVGACGVGKSALT	KRAS	17	G12C mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P7	YKLVVVGAAGVGKSALT	KRAS	17	G12A mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P8 (WT)	KLVVVGAGGVGKSALTI	KRAS	17	wildtype peptide with position G13 in the middle with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P9	KLVVVGAGDVGKSALTI	KRAS	17	G13D mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P10 (WT)	LDILDTAGQEEYSAMRD	KRAS	17	wildtype peptide with position Q61 in the middle with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P11	LDILDTAGKEEYSAMRD	KRAS	17	Q61K mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P12	LDILDTAGHEEYSAMRD	KRAS	17	Q61H mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P13	LDILDTAGREEYSAMRD	KRAS	17	Q61R mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P14 (WT)	GIPFIETSAKTRQRVED	KRAS	17	wildtype peptide with position A146 in the middle with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
KRAS_P15	GIPFIETSTKTRQRVED	KRAS	17	A146T mutation peptide with biotin at the C-terminus	1,5 mg/mL	1 µg/mL
MART-1/Melan-A26-35	ELAGIGILTV	MLANA	10	melanoma antigen peptide	2 mg/mL	1 µg/mL
MART-1/Melan-A27-35	AAGIGILTV	MLANA	9	melanoma antigen peptide	2 mg/mL	1 µg/mL
Survivin (97-111)	TLGEFLKLDRERAKN	BIRC5	15	wildtype peptide	2 mg/mL	1 µg/mL
Survivin mix (1-33)	N/A	BIRC5	N/A	15AA each (peptides 1-33)	2 mg/mL	1 µg/mL
Mesothelin-MPF_1	MALPTARPLLGSCGT	MSLN	15	Mesothelin peptide	2 mg/mL	1 µg/mL
Mesothelin-MPF_8	HRLSEPPEDLDALPL	MSLN	15	Mesothelin peptide	2 mg/mL	1 µg/mL

Mesothelin-MPF_11	LLPRGAPERQRLLPA	MSLN	15	Mesothelin peptide	2 mg/mL	1 µg/mL
Mesothelin-MPF_12	ALACWGVRGSLLSEA	MSLN	15	Mesothelin peptide	2 mg/mL	1 µg/mL
Mesothelin-MPF_14	FVAESAEVLLPRLVS	MSLN	15	Mesothelin peptide	2 mg/mL	1 µg/mL
Mesothelin-MPF_17	SVSTMDALRGLLPVL	MSLN	15	Mesothelin peptide	2 mg/mL	1 µg/mL
Mesothelin GPI P40 (596-610)	LQGGIPNGYLVLDLS	MSLN	15	Mesothelin peptide	2 mg/ml	1 µg/mL
Mesothelin GPI P41 (601-605)	MQEALSGTPCLLGPG	MSLN	15	Mesothelin peptide	2 mg/ml	1 µg/mL
NY-ESO-1-80-94	ARGPERSLLEFYLAM	CTAG1B	15	wildtype peptide	1,5 mg/ml	1 µg/mL
NY-ESO-1-89-103	EFYLAMPFATPMEAE	CTAG1B	15	wildtype peptide	1,5 mg/ml	1 µg/mL
NY-ESO-1-157-171	SLLMWITQCFLPVFL	CTAG1B	15	wildtype peptide	1,5 mg/ml	1 µg/mL
NY-ESO-1_1	MQAEGRGTGGSTGDA	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_2	GRGTGGSTGDADGPG	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_3	GGSTGDADGPGGPGI	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_4	GDADGPGGPGIPDGP	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_5	GPGGPGIPDGPGGNA	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_6	PGIPDGPGGNAGGPG	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_7	DGPGGNAGGPGEAGA	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_8	GNAGGPGEAGATGGR	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_9	GPGEAGATGGRGPRG	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_10	AGATGGRGPRGAGAA	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_11	GGRGPRGAGAARASG	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_12	PRGAGAARASGPGGG	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_13	GAARASGPGGGAPRG	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_14	ASGPGGGAPRGPHGG	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_15	GGGAPRGPHGGAASG	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_16	PRGPHGGAASGLNGC	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_17	HGGAASGLNGCCRCG	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_18	ASGLNGCCRCGARGP	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_19	NGCCRCGARGPESRL	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL

NY-ESO-1_20	RCGARGPESRLLEFY	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_21	RGPESRLLEFYLAMP	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_22	SRLLEFYLAMPFATP	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_23	EFYLAMPFATPMEAE	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_24	AMPFATPMEAELARR	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_25	ATPMEAELARRSLAQ	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_26	EAELARRSLAQDAPP	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_27	ARRSLAQDAPPLPVP	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_28	LAQDAPPLPVPGVLL	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_29	APPLPVPGVLLKEFT	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_30	PVPGVLLKEFTVSGN	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_31	VLLKEFTVSGNILTI	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_32	EFTVSGNILTIRLTA	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_33	SGNILTIRLTAADHR	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_34	LTIRLTAADHRQLQL	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_35	LTAADHRQLQLSISS	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_36	DHRQLQLSISSCLQQ	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_37	LQLSISSCLQQLSLL	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_38	ISSCLQQLSLLMWIT	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_39	LQQLSLLMWITQCFL	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_40	SLLMWITQCFLPVFL	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_41	WITQCFLPVFLAQPP	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_42	CFLPVFLAQPPSGQR	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL
NY-ESO-1_43	FLPVFLAQPPSGQRR	CTAG1B	15	wildtype peptide for mix	1,5 mg/ml	1 µg/mL

Flow Cytometry Panels and Gating Strategies

Table E.1: Intracellular Cytokine Staining and Degranulation - ICS and CD107a panel

Flow Cytometer Configuration		**PANEL**
Laser Wavelength	Detector/Filter Range	ICS
405nm	450/45	CD4 - PB - [5uL/100uL]
	601/20	Live Dead Aqua
488nm	763/43	IL17 - BV786 - [5uL/100uL]
561nm	525/40	IFNg - FITC - [5uL/100uL]
	585/42	TNFα - PE - [5uL/100uL]
	710/50	TCRgd - PC5.5 - [5uL/100uL]
638nm	763/43	IL2 - PC7 - [5uL/100uL]
	660/10	CD107a - APC - [10uL/100uL]
	712/25	CD8 - A700 - [5uL/100uL]

Table E.2: T-homing, T-cell subset and T-cell activation panel

Flow Cytometer Configuration		**PANEL**		
Laser Wavelength	Detector/Filter Range	Thoming	T-cell subset	T-cell activation
375nm	405/30			Live Dead Blue
405nm	450/45	CD4 - PB - [5uL/100uL]	CD57 - PB - [5uL/100uL]	CD4 - PB - [5uL/100uL]
	525/40	CD8 - KrO - [5uL/100uL]	CD45 - KrO - [2uL/100uL]	CD8 - KrO - [5uL/100uL]
	601/20	Live Dead Aqua	Live Dead Aqua	
	660/10	CXCR3 - BV650 - [2uL/100uL]	LAG3 - BV650 - [2uL/100uL]	LAG3 - BV650 - [2uL/100uL]
488nm	763/43	CD95 - BV675 - [5uL/100uL]	CD95 - BV785 - [5uL/100uL]	CD95 - BV785 - [5uL/100uL]
	525/40	CD103 - FITC - [5uL/100uL]	CD45RA - FITC - [5uL/100uL]	CD103 - FITC - [5uL/100uL]
561nm	610/20	CCR4 - ECD - [5uL/100uL]	CD28 - ECD - [5uL/100uL]	CD39 - ECD - [5uL/100uL]
	585/42	CCR7 - PE - [5uL/100uL]	CCR7 - PE - [5uL/100uL]	CCR7 - PE - [5uL/100uL]
	710/50	TCRgd - PC5.5 - [5uL/100uL]	PD1 - PC5.5 - [5uL/100uL]	TCRgd - PC5.5 - [5uL/100uL]
638nm	763/43	CCR6 - PC7 - [2uL/100uL]	CD27 - PC7 - [5uL/100uL]	CD69 - PC7 - [5uL/100uL]
	660/10	CCR9 - APC - [2uL/100uL]	CD4 - APC - [5uL/100uL]	4-1BB - APC - [2uL/100uL]
	712/25	CD45RA - APC A700 - [5uL/100uL]	CD8 - A700 - [5uL/100uL]	CD45RA - APC A700 - [5uL/100uL]

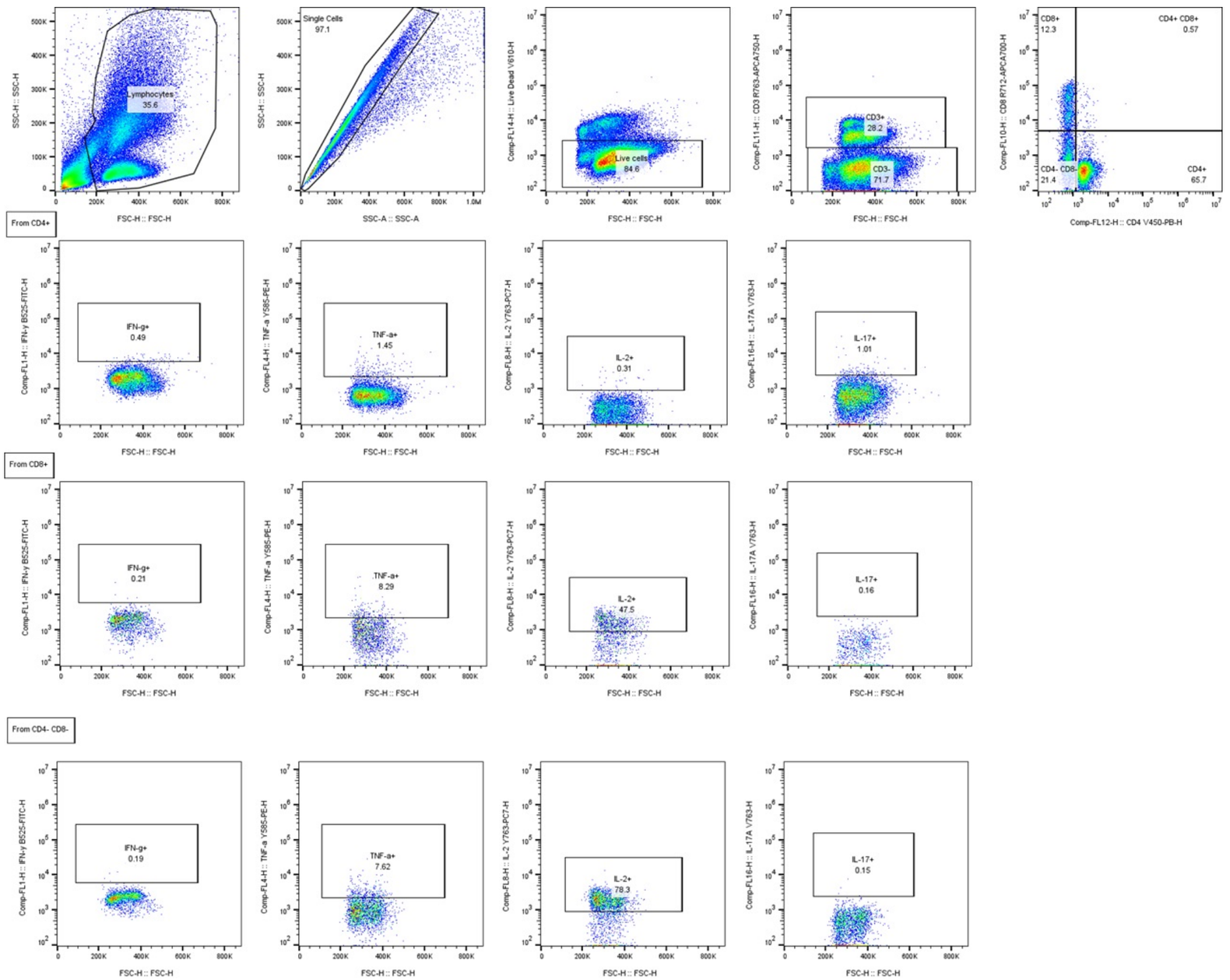


Figure E.1: Intracellular Cytokine Staining gating startegy

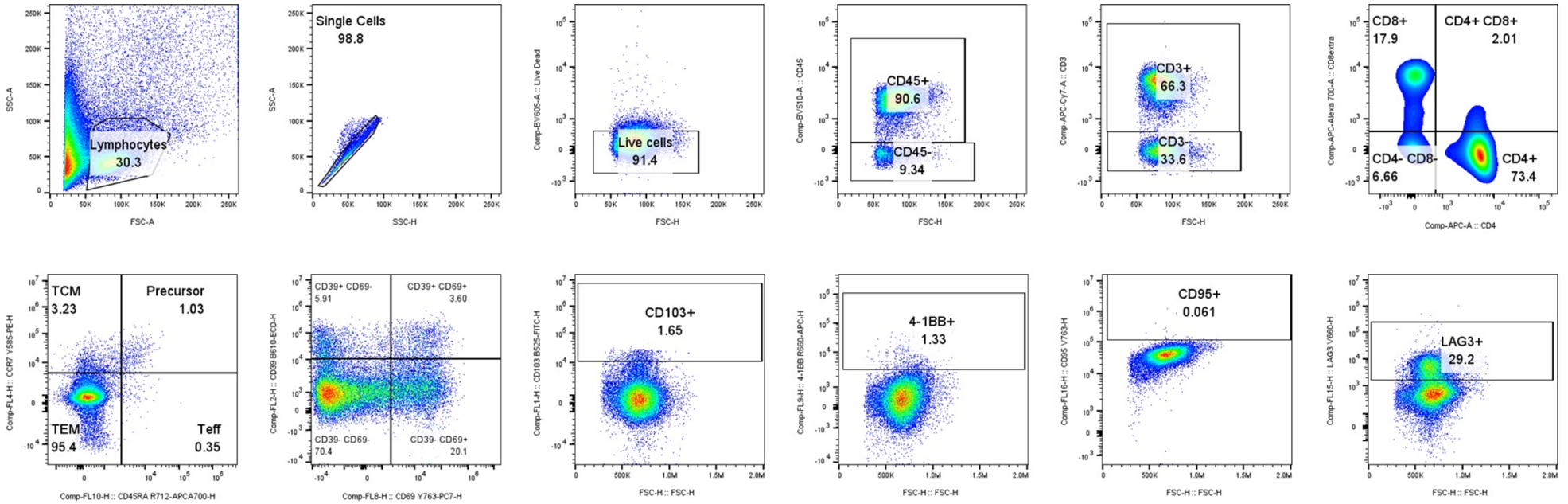


Figure E.2: T-cell Activation gating strategy

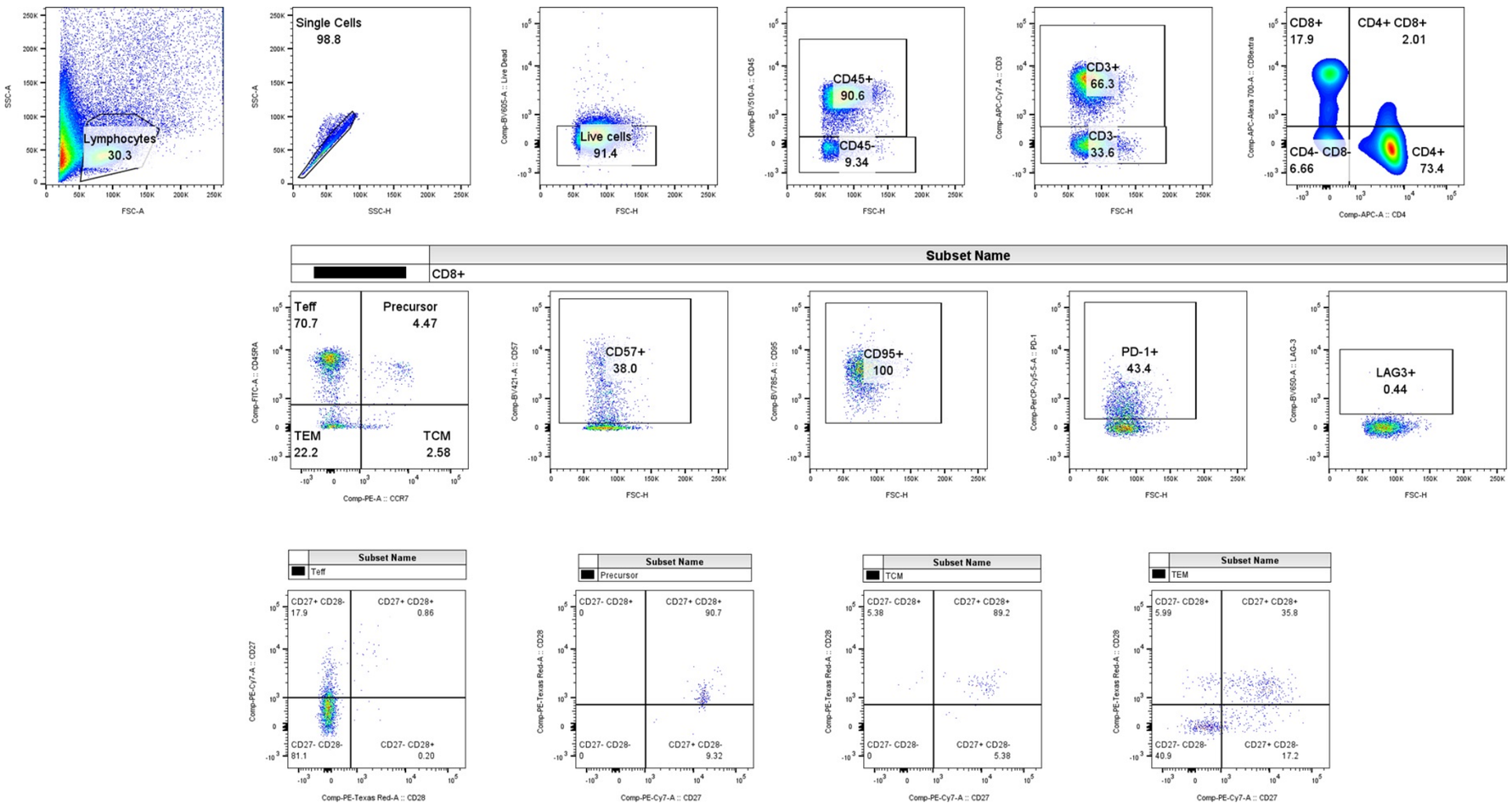


Figure E.3: T-cell subset gating strategy

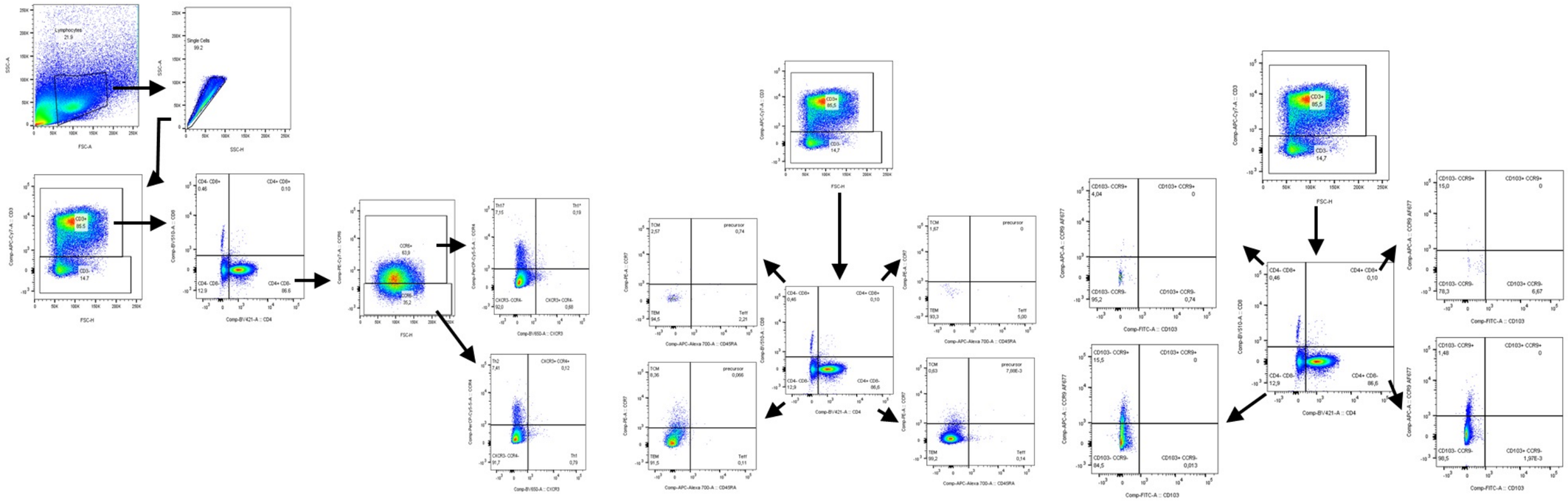


Figure E.4: T-homing subset gating strategy

Mestrado em Engenharia Biomédica

Construction of a decision support system for the effectiveness of Immune Cell Therapies

Pedro Noronha de Castro Machado Teixeira

2024