Analysis of murine CDR3β repertoires using machine learning techniques

Mattia Cinelli

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Infection and Immunity

University College London

October 30, 2018

I, Mattia Cinelli, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the thesis.

Labor Improbus Omnia Vicit

Abstract

This thesis presents my research on the development and application of state-of-the-art machine learning methods for the classification and analysis of murine complementarity-determining regions 3 (CDR3) repertoires. Using classification methods, I investigated the role and mechanisms of the CDR3 protein sequence. These are short protein regions present on the T-cell receptor (TCR), and I have aimed to identify the amino acids and positions that play a major role in the TCR, allowing it to recognise specific antigens and to activate the adaptive immune response.

The analyses performed are based on three different methods of machine learning: (i) The Support Vector Machine, used to carry out the classification analysis; (ii) An application of Bayesian theory, to isolate the most relevant CDR3 features; (iii) Markov chain and Hidden Markov Models, to study the variability of the repertoires and to identify specific regions of interest within the CDR3.

All of these methods have proved useful and have helped me to identify different features of the CDR3 repertoires. Indeed, specific position and combination of amino acid have been identified and considered relevant for repertoires classification. It has been detected the presence of three different levels of emerging conserved-areas in the CDR3, and investigated the role of the glycine and other amino acids within motifs and putative interaction site. Although the biological mechanisms of CDR3 are still not fully understood, my contribution to the field has been to increase our understanding of CDR3, including the identification of relevant position for the CDR3 interaction; motifs and patterns for the different groups of mice repertoires; and an improved overall classification of such repertoires.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof. Benny Chain for his extraordinary support during all period of my PhD. His example as scientist and man is a great source of inspiration, and I own him more then what could be expressed here.

I would also to thank my supervisors, Prof. John Shawe-Taylor and Dr. Andrew Phillips for their insightful comments and encouragement. This project would have not been impossible without the support of Microsoft Research scholarship and UCL studentship.

I want to thank my fellow colleague Ilaria, Jamie, Kanayo and Cristina and my friends Peppe and Laura. They supported me greatly in and out the lab.

Last but not the least, thanks to my parents, my sister and my wife for supporting me throughout writing this thesis and always being there for me.

Contents

List of Figures

List of Tables

Thesis Aim

Thanks to recent advances in high throughput DNA sequencing [1][2] and the impressive fall in the cost of sequencing [3], there has been a rapid increase in the size of TCR repertoires sequenced [4][5][6][7]. At this, exponential data growth must now follow an improvement in methods of analysis. Current methods fall short while analysing large repertoires and, like in the case of CDR3 repertoires, this is present and broadened, missing the potential that a large collection of data can offer.

Thus far, the repertoire analysis considers the CDR3 sequences as the smallest and most indivisible element of the repertoire. On this "atomic" element have been conducted all kinds of statistical investigation. Studies have inferred and revealed precious information such as the number of unique CDR3 sequences; whether they are rare or common; if they are shared among repertoires [8]; assessed theories on the mechanism of immune reaction, identifying, for example, that the immune responses are carried by a mixture of public and private specificities [9][10][11].

However, there are significant limitations to this approach. First, the "sequence similarity": two not-equal but very similar CDR3 that could potentially recognise the same antigen would be considered completely unrelated. It is well known that there are functional and structural hotspots, on peptides and CDR3s alike, that are fundamental for the TCR and pMHC (MHC carrying a peptide) interactions [12][13]. However, due to a lack of specific methods, this aspect is often missed. The second limitation is "motifs and size": in order to compare different protein sequences, the most common procedure is to try to identify shared domains, motifs or conserved elements. In bioinformatics, this is usually achieved by creating a multiple sequence alignment (MSA) [14]. However, these approaches are

not immediately applicable to CDR3 repertoires, because the number of proteins sequenced is too high and the sequences too short and diverse to allow for meaningful multi-alignment [15]. Third, the "repertoires similarity", comparison between repertoires is difficult and often completely ignored, preferring to focus on the sequences present within them.

There is, therefore, a need for developing of new methods that go beyond the analysis of entire CDR3s. Most importantly, new methods are needed that can identify the amino acids and positions within the CDR3 that play a key role in antigen discrimination and detection using a more "subatomic" approach, i.e. starting from the single amino acid and extending to larger patch of the sequence.

To accomplish these tasks, I chose to design and apply methods for the classification of CDR3 repertoires. Being able to correctly classify different types of repertoires gives us two important opportunities: first, the possibility of classification could open relational studies among the repertoires as well as investigation of the immune response against different antigens and exploring how immune response begins and mutates. Second, classification would enable us to look within each repertoire and investigate the "features" (specific amino acids and/or positions) that made the classification possible and see if these elements have a biological relevance for the immune response. These opportunities can help improve our understanding of the immune response and our ability to predict it.

Thesis Outline

1. **Part I Biological Introduction**: A functional description of T cells, T-cell receptors and CDR3. This part introduces the biological background needed to understand the aims of the projects:

   1.1 An overview of the immune system: Chapters 1.

   1.2 A deeper focus on the adaptive immune system: Chapter 2.

   1.3 A focus on the T cells: Chapter 3.

2. **Part II Results**: A summary of the database used, and the main results obtained. Each of the main results is presented in the form of a summary of the methods, workflow and conclusions.

   2.1 Analysis of the CDR3 dataset: Chapter 4.

   2.2 Review of [16] and application of the Support Vector Machine as possible classification method for CDR3 repertoires:

      2.2.1 Support Vector Machines: Chapter 5.

      2.2.2 Bag of Wards: Chapter 6.

      2.2.3 K-means algorithm: Chapter 7.

      2.2.4 Experiments, workflow and results: Chapter 8.

   2.3 Bayes' theorem and its application as feature selection for SVM classification.

      2.3.1 The Bayes' theorem: Chapter 9.

      2.3.2 Bayes' theorem as classification method: Chapter 10.

2.3.3 Feature selection using one dimensional naïve Bayes' classifier increases the OSR of support vector machine classification of CDR3 repertoires: Chapter 11. See [17] for more.

2.4 The Hidden Markov Model as a way to identify CDR3 motifs and classify them.

2.4.1 Markov Chains: Chapter 12.

2.4.2 Hidden Markov Model Theory: Chapter 13.

2.4.3 HMM-Based Programs Utilised: Chapter 14.

2.4.4 Experiments with Hammock: Chapter 15.

2.4.5 Hammock Results as features of an SVM classification test: Chapter 16.

3. **Part III Discussion**: Summary of the conclusions for each chapter and ideas for further analysis: Chapter 17.

The structure of this thesis reflects approximately my three years of doctoral research, with the repertoires analysis and SVM in my first year, Bayes' theorem in the second and HMM in my third. I will give in these chapters a clear explanation of the experiments as well as a description of the evolution of thinking that led me from one method to another.

Part I

Biological Background

Chapter 1

The Immune System

1.1 Overview

In this chapter, I briefly summarise the main features of the immune system, then focus on the T-cell receptor (TCR) and one of its most important components: the complementarity-determining region (CDR). Here, I refer primarily to the immune system in humans and, where specified, to the murine immune system.

1.2 Introduction

As old as life itself, all organisms have developed mechanisms to defend themselves from external intruders. Collectively, these mechanisms form the immune system of the organism, which has the role of stopping any pathogen that could cause harm or death to the host [18][19][20]. The word immune comes from the Latin, *immunitas* [21], which means to exempt, or to be free from, and it identifies the vital need for the organism to separate what belongs to the organism, called "self" and what does not, called "non-self". The need to protect the organism from possible threats has resulted, through evolution, in many forms of defence, with different approaches and degrees of complexity. Modern immunology divides the immune system into two broads, and at times overlapping, categories: the innate immune system and the adaptive immune system.

1.3 The innate immune system

The innate immune system is the most ancient form of immune defence. It is present in all classes of animals and plants in different forms and degrees of complexity. Even the smallest organisms have some form of mechanism to deal with possible intruders and to protect the organism from the external threats. As the word innate suggests, the defence mechanisms are already present and functional within the organism, before any threat is even encountered. Therefore, it can be immediately activated and begin the fight as soon as the intrusion is detected.

An important characteristic of the innate immune system is that, although the defence mechanism is already present and ready to respond to an external threat, the response to a threat is generally similar to previous ones, without providing a better or faster reaction to a subsequent encounter. This is the major point of difference between the innate and adaptive immune systems, as we will see in the next section.

Within the innate system, the easiest and most effective instrument to avoid infections is to prevent the intrusion of pathogens in the first place. This is performed using physical barriers to separate the organism from the outside world. This is the primary function of the skin that, when intact, is impermeable to most infectious agents. The skin can also produce sebum that, with its low pH, acts as an additional barrier against bacteria and viruses.

The defensive lines of our body are not only made by physical barriers but also chemical barriers, such as the mucus, secreted by the membrane in the internal surfaces, or other humoral fluid containing bactericidal substances including tears, saliva and nasal secretions. These are usually connected to mechanical systems such as blinking, coughing and sneezing. These mechanisms provide a first valid protection; however, many infectious agents can enter the body through wounds, gastrointestinal and urogenital tracts, or the respiratory epithelium, which is the case for the common cold.

When a pathogen overcomes the epithelial barriers and infects the body, many mechanisms of the innate immune system are immediately activated. For example, the blood contains several classes of soluble molecules that can kill or weaken

pathogens. These include antimicrobial enzymes, such as lysozyme, which can digest bacterial cell walls; antimicrobial peptides such as the defences that lyse bacterial cell membranes; and a system of plasmatic proteins known as the complement system, which can target pathogens both for lysis and for phagocytosis by employing cells of the innate immune system such as macrophages. These mechanisms are part of what is called humoral immunity, the part of the immune system that resides in the tissue fluids.

The complement system is one of the major defence systems in the body. It is a complex and intricate system, with a large number of functions. The complement system can trigger and amplify inflammation reaction, direct microbial killing, attract phagocytes, and help the development of an antibody response (one of the many links between the innate and adaptive immune systems).

The innate immune system is not only composed of humoral immunity but also a complex system of immune cells floating in the blood and tissue, forming what is called cellular immunity.

1.4 The myeloid lineage: Cells of the innate immune system

Both innate and adaptive immune responses are based on the activities of white blood cells, also known as leukocytes [18]. All cells of the immune system originate in the bone marrow and, with a few exceptions, also develop and mature there. Leukocytes migrate to the peripheral tissues, circulating in the bloodstream or in a specialized system called the lymphatic system, which carries the lymph, a fluid containing white cells, around the body. Within the lymphatic system are present the lymphatic organs such as the spleen, which produces an immune response through the production of antibodies, and the thymus, in which the maturation of T cells occurs.

All cells of the blood, including red blood cells for oxygen transportation, platelets that repair damaged tissues, and white blood cells of the immune system, derive from the hematopoietic stem cells of the bone marrow, known as pluripotent

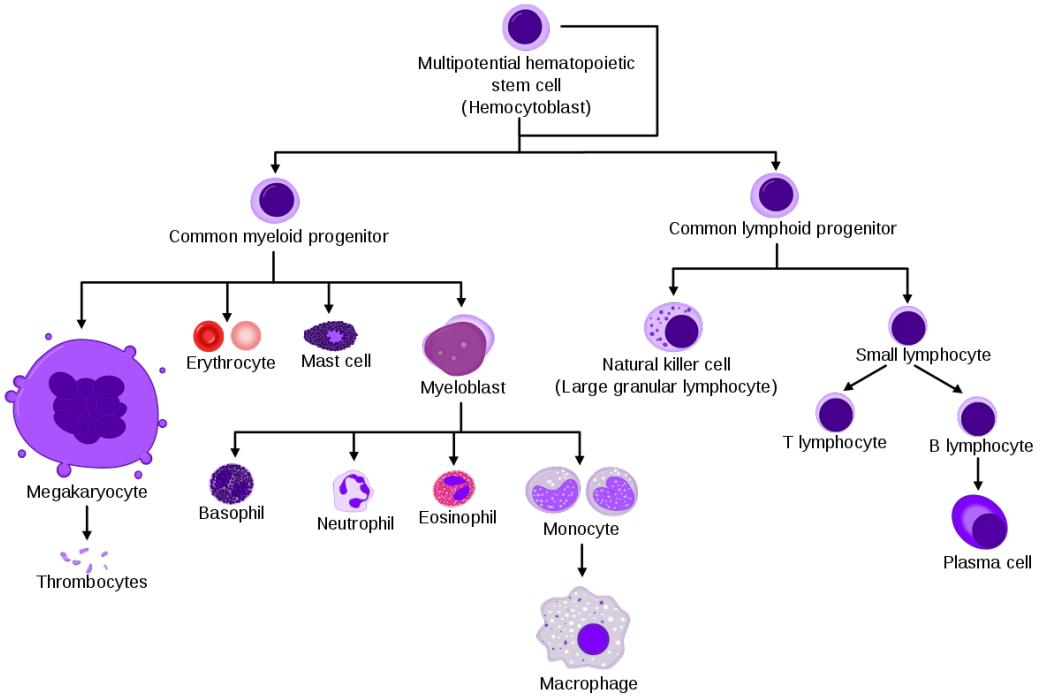


Figure 1.1: Diagram of human haematopoiesis from stem to mature cells: In humans, the haematopoietic stem cells are located in the medulla of the bone marrow and have the ability to differentiate in all types of mature blood cells and tissues [22].
In the first stage of haematopoiesis the cell differentiates into either the common myeloid progenitor or the common lymphoid progenitor. The first (present on the left-hand side of the tree) produces two lineages: the cells of the myeloid lineage known as granulocytes, megakaryocytes and macrophages that are involved as innate immunity and the erythrocyte, or red blood cells. The second (on the right-hand side) produce the adaptive immune system cells, T cells, B cells and Natural killer cells. Figure source [23].

hematopoietic stem cells. They differentiate into stem cells of less developmental potential that are the immediate progenitors of red blood cells, platelets, and the two main categories of white blood cells, the lymphoid lineages and myeloid lineages. See Figure 1.1.

The common myeloid progenitor is the precursor of the macrophages, granulocytes, mast cells and dendritic cells of the innate immune system, and of megakaryocytes and red blood cells, which will not be explained in this thesis.

Macrophages are relatively long-lived cells, present in almost all tissue. They perform several functions of the innate immune response and are also connected with the adaptive immune response. Their main task is to engulf and kill invad-

ing microorganisms by what is called phagocytosis. They literally surround, engulf and digest the pathogen and the infected cells targeted by an adaptive immune response. They also coordinate the immune response by inducing inflammation, which secretes proteins that activate other immune-system cells, such as cytokines, and recruit other cells to an immune response site.

The granulocytes are so called because of the presence of several dense and colourful granules in the cytoplasm; they are also called polymorphonuclear leukocytes because of their oddly shaped nuclei. There are three types of granulocytes —neutrophils, eosinophils, and basophils —which are distinguished by the different colour properties of the granules. They are all relatively short-lived, typically for a few days, and are produced during immune responses, when they leave the blood to migrate to sites of infection or inflammation, where pathogens and infected cells are to be found.

The dendritic cells (DC) have long finger-like protuberances, like the dendrites of nerve cells, which give them their name (from the Greek dendron, meaning "tree" [24]). Immature DC migrate through the peripheral blood from the bone marrow to the tissues. As for macrophages and neutrophils, DC also engulf the pathogens. However, their main purpose is not to eliminate the infectious pathogen by direct phagocytosis, but to present chunks of the digested pathogen to a class of lymphocyte cells called the T lymphocytes. The pieces of pathogen exposed on the DC surface are called antigens.

Therefore, the DCs and other cells, such as macrophages and B cells, are also called "antigen presenting cells" or APCs. The DC is the most important cell type in this class, the act of presentation being the main function of the DC. This is one of the most important links between the innate and the adaptive immune system, supplying clear clues to the pathogen kind and its characteristics, allowing the adaptive system to develop an effective and precise response.

Chapter 2

The adaptive immune system

The adaptive immune system is the second typology of immune system present in humans, as well as all jawed vertebrates. The adaptive system is not an alternative or unrelated type of immune system in respect to the innate. On the contrary, both systems work together for the common goal of protecting the host. The feature which distinguishes the adaptive system is its ability to create specific and targeted responses to new threats, discerning what belongs to the host, thus "self", from what does not, "not-self", and to remember those threats already encountered (immunological memory).

If the innate response is fast, immediate, available and gives a generic response, the adaptive response is instead much slower, needing to develop a specific reaction. However, this delay occurs only on the first encounter. After that, the system can remember the event, even for the entire lifespan of the host, and give a rapid response at a second encounter. This capacity, for instance, is what accounts for the success of vaccination [25]. Therefore, the power of the adaptive system is reliant on its ability to fight new pathogens, and to recognise and remember old threats.

2.1 The lymphoid lineage: Cells of the adaptive immune system

As we have already seen (Figure 1.1), the pluripotent hematopoietic stem cell differentiates into a myeloid and lymphoid lineage, the latter being much simpler compared to the myeloid, and presenting fewer types of cells: the Natural Killer cells

and the cells of the adaptive immune system T and B.

The Natural Killer (NK) cells form 15% of all lymphocytes present in the blood, and it has a peculiar granular cytoplasm, reminiscent of that present in the granulocytes. Although deriving from the same common progenitor of T and B cells, the NK are considered part of the innate immune system. NK are principally involved in the elimination of intracellular pathogens, host cells infected by viruses, and some abnormal cells; for example, some tumour cells.

The last group of cells are the lymphocytes T and B. These two kinds of cells are also known as antigen-specific lymphocytes for their peculiar characteristic of being able to recognise and bind a large diversity of antigens, using the receptors on their cellular surface. These receptors are the special tools that allow these lymphocytes to recognise, by direct physical contact, what belongs to the host and what does not. When recognition occurs, the lymphocytes become activated, and differentiate further into fully functional lymphocytes, known as effector lymphocytes.

Each lymphocyte carries only a single type of receptor and its effectiveness is limited to a few epitopes (the part of the antigen recognised by the receptor). Therefore, the adaptive system relies on the great number and variability of lymphocytes circulating at any given moment in the blood, presenting a valid defence to any possible threat to the host.

There are two types of lymphocytes: B lymphocytes (B cells) and T lymphocytes (T cells), with different roles and maturation place, and distinct types of antigen receptors.

B cells develop in the bone marrow, its name coming from the bursa of Fabricius [26], a lymphoid organ present in birds where these cells were originally discovered.

The functions of the B cell are the production and secretion of antibodies (Ab), one of the main components of the humoral immunity component of the adaptive system, the release of cytokines and the presentation of antigens; B cells are considered professional APCs.

On the surface of the B cell is present a receptor called a B-cell receptor (BCR).

The B cell circulates in the body scanning the surrounding environment with its BCR, looking for not-self antigens. Once the BCR bonds with an antigen the B cell will proliferate and differentiate into a plasma cell, the smaller part into memory B cells. Plasma cells are the effector form of the B cells, attacking the pathogen by secreting antibodies.

Antibodies are Y-shaped molecules with two branches and an antigen specificity sites on the tips of each branch. These sites are identical to the B-cell receptor. Therefore, the antigen that activates a given B cell becomes the target of the antibodies produced by the plasma cell.

A major role of the antibodies in adaptive immunity is to help the other cells of the immune system to see and engulf the microbe or the infected cell. Antibodies binding to their antigens can flag these cells and make their elimination much easier.

Chapter 3

The T Cell

3.1 T cells, functions and typologies

The T cell, together with the B cell, is the other of the two cells of the adaptive immune system. Thanks to their wide range of functions, the T cells can be considered one of the most important cells inside the immune system. Indeed, the T cell is one of the main protagonists in the activation, regulation, memory and killing process of the adaptive immune system.

Once the T cell is activated by an antigen recognition, it differentiates into several different subtypes of cells: we can gather them into four broad categories:

- **The Helper T cells (T_H).** This is a large subgroup of T cells, containing many subtypes such as T_H1, T_H2, T_H3, T_H17, T_H9, or T_{FH}. As their name suggests, the function of the cells is to "help", to sustain and guide the immune reaction. After T cells have been activated by the APCs, the cells differentiate into one of the many subtypes of T_H, which rapidly divide and secrete a large number of cytokines. Doing that, they can mediate the cytotoxicity and inflammation, and stimulate B cells to proliferate and produce antibodies.

  The most common subtypes of T_H are the T_H1 and the T_H2. The first is responsible for defence against intracellular viral and bacterial pathogens, the second against large extracellular pathogens and allergic responses.

- **Regulatory T cells (T_{Reg}).** If T_H has the role of sustaining the immune response, T_{Reg} has the role of suppressing it, by inhibiting the proinflammatory

T cells, suppresses autoreactive T cells that escaped the process of negative selection, preventing autoimmune disease.

- **The Cytotoxic T lymphocytes (CTLs), or T Killer cell.** These cells are responsible for the direct destruction of virus-infected cells, damaged or tumour cells. Host cells infected by intracellular pathogens would express a molecular named MHC class I (we will it see later), that is recognised by the CTLs. This is crucial for the body; indeed, intracellular pathogens are generally overlooked by the innate immune system.

- **The memory T cells.** These are the T cells that develop after an encounter, and persist for a long time inside the body, after the infection is concluded. The function of this cell is to give a faster and stronger immune response in a second encounter with the same antigen. This is the mechanism at the core of the vaccination process [27].

3.2 T-cell development

Both T cells and B cells derive from the common lymphoid progenitor, as seen in Figure 1.1, but unlike the B cells that mature in the bone marrow, the T cells migrate from there to mature in the thymus.

The thymus is one of the primary lymphoid organs present in the human body: where the lymphocytes would differentiate, proliferate, be selected and then mature into functional cells.

The thymus is a bilobed organ, present in the thoracic cavity behind the sternum, just above the heart. Each of the two lobes is organised into lobules, divided by connective tissue, and each lobule is divided into an outer cortex and an inner medulla. The two areas of the lobules reflect the maturation of the T cells: in the outer part are present the immature T cell, and the inner part the mature cells.

The process that leads the hematopoietic progenitor of the T cells (or thymocytes) to mature into functional T cells is called thymopoiesis. This starts as soon as the thymocytes migrate from the bone marrow and enter the thymus.

The maturation of thymocytes requires progression through the different stages of thymopoiesis. To mark these stages, it is possible to use the presence and absence of different kinds of surface glycoproteins, known as clusters of designation or classification determinant (CD). For this thesis, we will focus primarily on CD4 and CD8, which have a role as co-receptors during the antigen presentation event, and would determine the nature and function of the cell.

The thymopoiesis can be divided into three broad stages, depending on the CD present on the surface of the cell. At the earliest stage, the thymocytes express neither CD4 nor CD8. For this reason, it is called the **double-negative stage** (DN) ($CD4^{-}CD8^{-}$). Later, the T cell develops a second stage, called the **double-positive stage** (DP)($CD4^{+}CD8^{+}$), where both surface proteins are present. At the end, the cell passes to a **single-positive stage** where only one of the proteins is present ($CD4^{+}CD8^{-}$ or $CD4^{-}CD8^{+}$). The cell is now mature and is released from the thymus to peripheral tissues [28].

CD4 and CD8 are important for the stage classification and future nature of the mature T cell. However, the significant event during T-cell development is the production of a functional and non-self-reactive T-cell receptor (TCR). We have already seen that the TCR plays a fundamental role in the immune system, recognising what is considered to not be part of the body, and kick-starting the immune response. Given this great importance, the production and selection of TCR is a highly organised and regulated event.

The DN stage can be divided into four different stages, during which the thymocyte proliferates, loses the B-cell and myeloid potential, and rearranges the β chain (one of the two chain of the T cell) loci to produce a functional β chain. If the β chain can successfully pair with an invariant form of α chain, the cell can pass to the DP stage, otherwise the T cell is eliminated by apoptosis (β-selection stage). The following DP stage is focused, instead, on the production of a functional α chain. This chain will be tested to pair with the β chain, leading to apoptosis if an error occurs and forming a TCR otherwise.

The TCR is now formed and must be tested to prove its functionality. The

TCR passes through two series of selection that will determine if the cell can be released into the body or eliminated. These selection processes are named positive and negative selection.

During **positive selection**, the DP thymocytes interact with different Major Histocompatibility Complex, a set of surface protein that carries antigens, present in two classes: class I (MHC class I) or class II (MHC class II) and that are present on the surface of the thymic cortical epithelial cells. If the thymocytes interact with an appropriate intermediate affinity (not too strongly or too weakly) they are considered suitable and not eliminated.

During this stage, it will be defined the glycoprotein that will be present on the surface of the thymocyte. If it has interacted well with the MHC class II, the T cell will become a $CD4^+$ cell. Otherwise, it will mature into a $CD8^+$ T cell.

During **negative selection**, the TCR are now tested not based on their affinity to the antigen but based on which typology of antigen they are attracted to. The negative selection must eliminate all the T cells capable of being activated by self-peptides that will be naturally present in the body.

The negative selection happens in the medulla region of the thymus: here, the medullary thymic epithelial cells present the MHCs carrying self-antigens. The thymocytes that interact too strongly with the antigen will receive an apoptotic signal leading to the death of the cell. Those interacting weakly are spared and will become Regulatory T cells. With these two selections, the body ensures that the TCR has been bound to the antigen with an adequate strength (positive selection) but do not recognise an antigen belonging to the host (negative selection).

The thymopoiesis is, therefore, a very well organised process, where the TCR is tested in all its aspects, with several checkpoints and control systems. It has been calculated that only 2% of all thymocytes will survive to the thymopoiesis and leave the thymus as mature T cells [19] .

3.3 Antigen presentation

I briefly mentioned that the T cells are activated by the recognition of a not-self antigen, and this antigen is presented to the TCR by the APC cells (dendritic cells, macrophages, etc.).

Let us now explore the process in greater detail.

The event in which the APCs present the antigen to the T cell is called an antigen presentation event. The elements involved are: the T-cell receptor (which will be analysed in the next section); the major histocompatibility complex (MHC), which is a surface protein present on the APCs that physically carries the antigen; and the antigen itself.

Generally speaking, an antigen is defined as a molecule inducing an immune reaction. The part of the antigen recognised by the TCR is called the epitope (or antigenic determinant), and the part of the TCR/BCR recognising it is called the paratope. An antigen can have different epitopes, and the same epitope can be recognised by various T-cell receptors: this phenomenon is known as cross-reactivity.

Antigens can derive from the external environment and internalised (exogenous antigens) or can be generated inside the host cells due to viral or intracellular bacterial infection (endogenous antigens). The antigen presented to the TCR by the MHC is usually a short piece of protein: 8-10 amino acid for MHC class I and 13-24 for MHC class II.

The MHC molecules are a group of cell surface proteins essential for the immune system. They are expressed by the APC cells for the antigen presentation, and from the virus-infected cells. The principal function of the MHC molecules is to bind the antigens derived from the degradation of the pathogens and display them on the cell surface.

In humans, the MHC, are also known as human leukocyte antigens (HLA), because of their first identification as histocompatibility (transplantation) antigens on the surface of leukocytes [30]. The most relevant class of MHC are the MHC class I and II. These have a dissimilar structure and functions. MHC class I carries en-

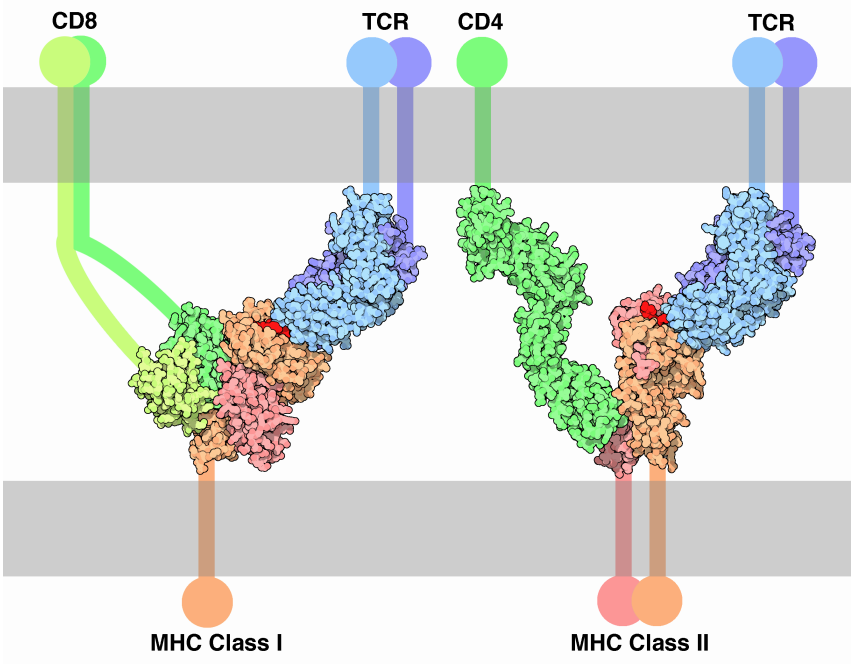


Figure 3.1: The interaction between CD4$^+$ and CD8$^+$ TCR and, MHC class I and II: In this picture we can see the interaction between the TCRs, the clusters of differentiation (CD) and the MHC class I and II. Despite the TCR being essentially identical in both CD4$^+$ and CD8$^+$ T cells, a successful interaction is aided when the CDs on their surface interact with the MHC.
On the surface of the T cells can be present only one kind of CD therefore, the T cell can recognise only a type of MHC. This will decide the "fate" of the T cell. Indeed, the MHC class I (left-hand side) can only bind endogenous antigens and it can only be recognised by CD8 co-receptors (Cytotoxic T cells). While MHC class II (right-hand side) carries exogenous antigens and it can only be recognised by CD4 co-receptors (Helper and Regulatory T cells).
The CD4 is formed by four consecutive immunoglobulin domains, from D1 the most external, to D4 which is anchored to the surface. The CD8 is formed by two chains: α and β, belonging to the immunoglobulin superfamily, and an intracellular tail. MHC class I is formed by two polypeptide chains, α and $\beta 2$-microglobulin. While the MHC class II consists of two chain α and β, both divided in two region $\alpha 1$, $\alpha 2$ and $\beta 1$, $\beta 2$. All these proteins are anchored to the membranes by flexible chains and hinges, so that they can move around and form this complex interaction. Source [29]

dogenous (or intrinsic) antigens, those antigens derived from viruses and pathogens inhabiting the cell, while MHC class II carries antigens which are derived primarily from exogenous sources, like extracellular pathogens (exogenous or extrinsic antigens).

MHC class I is formed by three different extracellular domains ($\alpha 1$, $\alpha 2$, $\alpha 3$), a transmembrane region and a cytoplasmic tail. MHC I can only bind endoge-

nous antigens and can only be recognised by CD8 co-receptors, (Cytotoxic T cells). MHC class II is instead formed by a α chain (heavy) and a β chain (light). It carries exogenous antigens. It can only be recognised by CD4 co-receptors, (Helper and Regulatory T cells). See Figure 3.1

The activation of TCR happens when it recognises the MHC-peptide complex, together with other co-stimulatory signals.

3.4 TCR Structure

T-cell receptor is a heterodimer surface protein, formed by two polypeptides connected by disulphuric bonds and anchored to the membrane by a small membrane domain, and a very short cytoplasmic tail (Figure 3.2 b). It is structurally and functionally similar to the BCR and antibodies belonging to the same family of immunoglobulin (Figure 3.2 a).

It is usually formed by a combination of α and β chains. $\alpha\beta$ T cells make up 90-95% of all T cells of the peripheral blood in humans. The alternative form of T cell is made by two similar polypeptides, the γ and δ chains, therefore called, $\gamma\delta$ T cell [18][31].

Both kinds of receptors are associated with a set of five polypeptides which form the CD3 complex, all together forming the TCR-CD3 complex (Figure 3.2 b). [18][32].

Each chain of the TCR is composed of a constant (C) region on the intracellular side, and a variable (V) region on the extracellular face of the receptor. A key characteristic feature of the V region of TCRs (and BCRs) is that their genes are not functionally encoded in the germline. Instead each chain derives from several non-contiguous DNA segments that are selected and recombined. However, TCR gene selection is highly biased and not all combinations of genes are equally likely [33].

The locus for the α chain contains two categories, or types, of segments: the variable (V) and joining (J) segments. While for the β chain a third type is present in addition to these two: the diversity (D) segment (Figure 3.2 c).

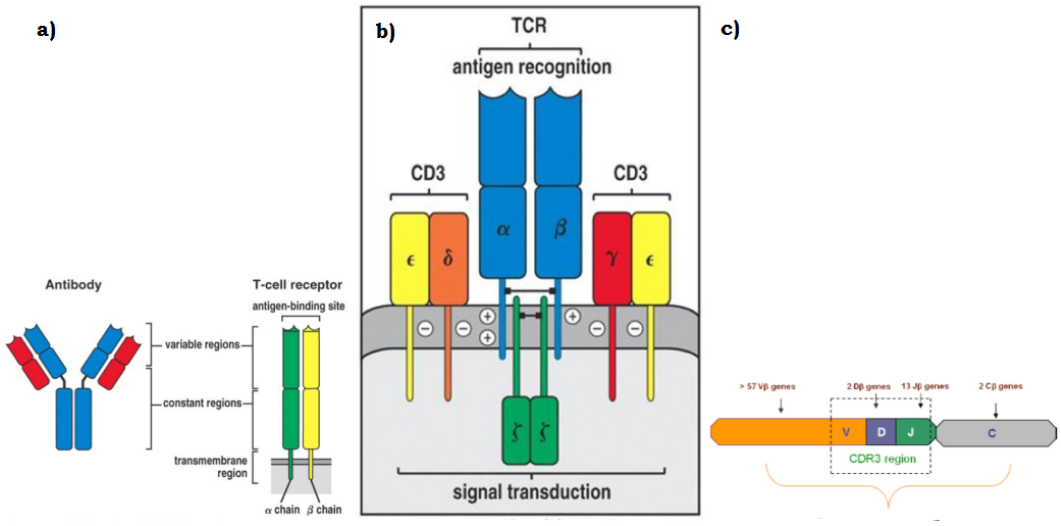


Figure 3.2: The TCR structure, comparison and the CD3 complex: In this picture we can see the structure of the TCR and the TCR-CD3 complex. In **Sub-fig a:** the structure of the TCR and BCR are compared, both belong to the same family of immunoglobulin, with similar structures made of a constant and variable region. They also have a similar function, recognising antigens through their binding site made by two CDR3s. Given the similarity of the latter, any understanding in the antigen-CDR3s interaction in one would help the study of the other molecule. In **Sub-fig b:** the TCR-CD3 complex, the CD3s is reported. The complex consists of a CD3 γ chain, a CD3 δ chain, and two CD3 ε chains. These chains are associated with the ζ chain (zeta chain). Together those are considered relevant for the transmission of the activation signals after peptide binding to generate an activation signal in T lymphocytes. The TCR, ζ chain, and CD3 molecules together constitute the TCR complex. **In Sub-fig. c:** the structure of a β chain of the TCR is reported. In this chain the variable domain is formed by one more region (D), absent in the α chain [18].

On the top of the variable domain on both chains three hypervariable [34] regions are present, better known as complementarity-determining regions (CDR). These six regions are considered very important for the recognition of, and the interaction with, the pMHC. We will analyse these regions in greater detail in the following sections.

As suggested by the presence of the CDRs, the variable region of the TCR is more oriented to interaction with the MHC, while the constant region has more a structural role, with the presence of disulphuric bonds between the two chains.

Because the TCR does not have a cytoplasmic domain capable of transmitting a signal after interaction with the MHC, this role is entrusted to the CD3 proteins in

the TCR complex. The cytoplasmic portions of the CD3 contain sequences called ITAMs that are targets for protein kinases. These segments become phosphorylated briefly after the TCR-MHC interaction, which produces physical changes in the TCR structure, which in turn can propagate the signal [35].

3.5 V(D)J recombination

As mentioned, the function of the TCR is to recognise the not-self antigens present in the host. However, it would be impossible for the host to encode in its genome a different TCR sequence for each possible antigen.

In this scenario, the space needed in the genome would be enormous, while the host would be vulnerable to any new mutation occurring in a pathogen. To avoid this, evolution has provided a system that needs only a relatively small number of genetic loci, that can be combined and rearranged in a way to produce a wide and effective variation of TCRs within the repertoire.

This system is called V(D)J recombination. A schematic for TCR is represented in Figure 3.3.

The V(D)J recombination occurs to both B and T lymphocytes during their development in the primary lymphoid organs: for the B cells in the bone marrow, and for the T cells, during the thymopoiesis, as we have already seen. For both lymphocytes, the system of V(D)J recombination is similar, producing a great range of immunoglobulins and antibodies.

3.6 The number of TCR and T cells

The total number of T cells and TCR diversity in the human body and other animals, especially in mice, has been one of the main debates regarding T cells in the last two decades [36][37]. It has been noted that the value of clonotypes in the peripheral blood changes with factors like ageing, pathogens and viral infection, immunisation and transplantations [37][38][39][40][41]. For these reasons, a precise over-time description of repertoire diversity would help in understanding how any of these factors can influence our life and the fight against diseases.

The most common form of T cell in humans is the $\alpha\beta$ T-cell receptor. The

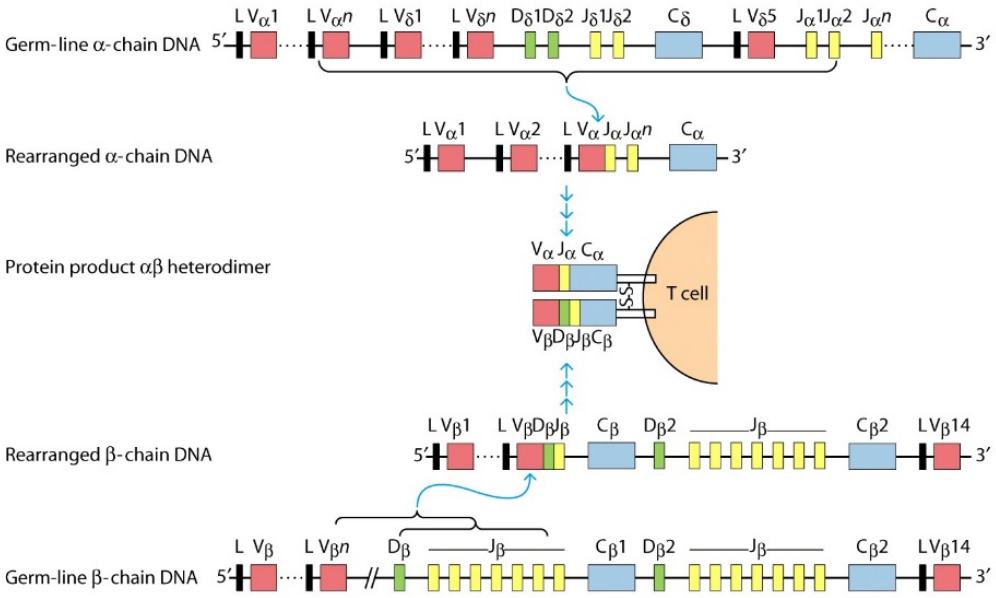


Figure 3.3: V(D)J recombination: Here is reported a schematic representation of the V(D)J recombination: both α and β-chain genes are composed of discrete sections that are joined by somatic recombination during the T-cell development.
For the α chain (upper part of figure), a V_α gene segment rearranges to a J_α gene segment to create a functional V region exon. Transcription and splicing of the VJ_α exon to C_α generates the mRNA that is translated to yield the T-cell receptor α-chain protein.
For the β chain (lower part), the variable domain is encoded in three gene segments, V, D and J. Rearrangement of these gene segments generates a functional VDJ_β. V region is transcribed and spliced to join the C_β; the resulting mRNA is translated to yield the T-cell receptor β chain.
The α and β chains pair soon after their biosynthesis to yield the α:β T-cell receptor heterodimer. Source [18].

β-chain (TRB) locus is formed by 54 TRBV genes, 2 TRBD genes, 13 TRBJ genes and 2 TRBC genes (see Table 3.1). The first step of beta recombination is the recombination of one of the two D genes, Dβ1 and Dβ2, recombining respectively with one of the six Jβ1 segment, or with one of seven Jβ2 segments. The DJ recombination is followed by the rearrangement of one of the circa 50 Vβ gene segments, with the DβJβ already rearranged. All of that it is combined with one of the two segments of the constant domain genes (Vβ-Dβ-Jβ-Cβ).

The α-chain locus is composed of 47 V genes, 57 J genes and a single TRAC (See Table 3.1). This resembles a V-to-J rearrangement with a lack of D segments, but a prodigious number of J segment [42].

Number of segments		
Regions	**α Chain**	**β Chain**
Variable (V)	47	54
Diversity (D)	-	1,1(2)
Joining (J)	57	6,7(13)
Constant (C)	1	2
Possible Combinations		
Segments combinations	2,679	2,808
Merging α and β	7,522,632	
P-N insertions	$10^{15}/10^{20}$	

Table 3.1: Theoretical number of $\alpha\beta$ TCR by the V(D)J recombination: In humans, the TRA locus comprises 47 TRA (T-cell receptor alpha) V genes, 57 TRA-J genes and a single TRA-C gene. VJ recombination can rearrange these 104 genes into 2,679 unique α-chain VJC gene combinations.
The T-cell receptor beta (TRB) locus contains 54 TRB-V genes, 2 TRB-D genes, 13 TRB-J genes and 2 TRB-C genes. VDJ recombination can rearrange these 71 genes into 2,808 unique β-chain VDJC gene combinations.
Merging all α and β chain would produce an impressive 7 million combinations. Subsequent deletions and insertions of nucleotides at the junction section would result in a theoretical repertoire of $10^{15}/10^{20}$ different TCRs in humans. Data source [42].

The entire process of the V(D)J recombination is mediated by a class of enzymes called VDJ recombinase, of which the most important are the recombination activating genes (RAG) 1 and 2, the terminal deoxynucleotidyl transferase (TdT).

To proceed with the recombination, the recombinase makes use of special regions flanking the gene segments for all V, D and J regions. Such regions are called recombination signal sequences (RSSs), and they are formed by three elements: a heptamer, that is, seven conserved nucleotides (CACAGTG); a nonamer, nine conserved nucleotides (ACAAAAACC); and a region of 12 or 23 base pairs in length.

The nucleic acid composition of the 12/23 base pair is poorly conserved, but 12 and 23 base pair are approximately equal to one or two turns of a DNA helix. This space is used by the recombinase enzymes and therefore it is referred to as 12/23 rule.

If we take the values in Table 3.1 and compute the possible combinations with all segments we will find that for the α chains, 2,679 unique combinations of the 104 genes of VJ recombination are possible. For the β chain 2,808 unique sequences

are possible. Finally, all the possible combinations of α and β chains can give rise to a value of $7.5 \cdot 10^6$ TCRs [43].

The V(D)J recombination is a biased event [33]. Therefore, the body would not produce each TCR with the same frequency. This could risk decreasing the possible number of TCRs and the overall repertoire of variability. However, during the process of recombination, the RAG enzymes break the DNA, making a single-strand cut at the $3'$-$5'$ end of the heptamer: DNA repair enzymes would add and/or remove various nucleotides (insertion/deletion event), called palindromic (P) nucleotides, while the TdT will add/remove non-templated (N) nucleotides to the $5'$-$3'$ direction.

This event can boost the value of the theoretical repertoire to 10^{15}-10^{18} different TCRs [41][42][43]. This large value exceeds even the number of cells in a human body, and so, naturally, only a smaller number of combinations is present at any given time.

Thanks to new modern sequencing techniques, such as high-throughput sequencing (HTS), the possibility of gaining a clearer insight into the number of different T cells in the blood has increased.

The latest value for T cells in the peripheral blood is around 10^{11} [37]. Included in this value are all T-cell clonotypes, thus all the T cells that share the same TCR. Therefore, the total number of T cells is a value closer to 10^{10}. In mice, the total number of naïve T cells in peripheral blood is 10^7, with the number of single clonotypes about half the size [42].

3.7 The complementarity determining regions and the MHC interaction

The variable region of the TCR is the part of the receptor used for the recognition of the MHC carrying the peptide. Thanks to studies on the multiple sequence alignments of TCRs and antibodies [44], it has been possible to identify six specific regions that interact directly with the MHC and the peptide.

These regions are poorly conserved, and for this reason they were originally named hypervariable regions. However, structural studies and X-ray

crystallography-imaging showed that these regions are loops on the top of the TCR, and they interact directly with the MHC and the peptide. Because of this, they have been renamed Complementarity Determining Regions (CDRs) [45]. Although they have been extensively studied, the precise mechanisms by which the CDRs can bind the MHC and recognise the peptide are still largely unknown.

On each chain of the TCR are present three of these regions: CDR1, CDR2 and, CDR3 (Figure 3.4).

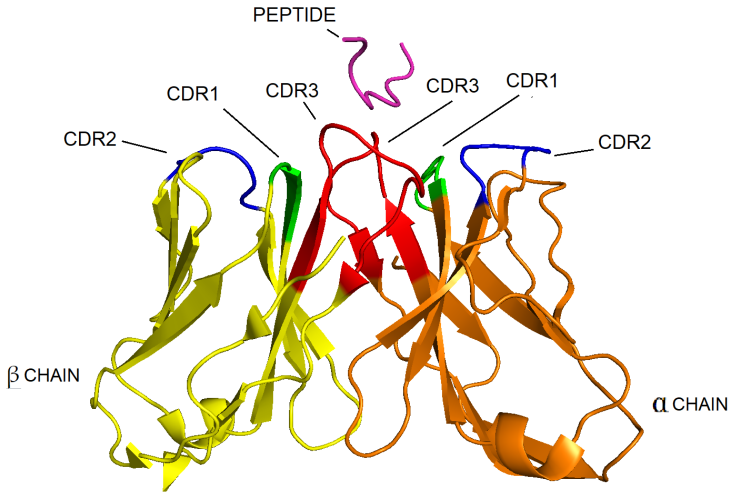


Figure 3.4: Particular of TCR V region and peptide: Above is illustrated the TCR V region interacting with a peptide (magenta). The three pairs of CDRs are pointed out and we can see that the CDR3s (in red) are in direct contact with the peptide, suggesting that these play a major role in the interaction with the peptides while the CDR1 (in green) and CDR2 (in blue) are more involved with the interaction with MHC. In yellow the β chain, in orange the α chain. Picture produced with PyMol, PDB crystal structure ID: 4MNQ [46].

The three pairs of CDRs loops play different roles during the interaction with the p:MHC. Their loops are arranged in different ways (see Figure 3.4) and have a different role in the interaction.

The CDR1 and 2 are mainly involved with the binding of the MHC [12]. Both loops work together to allow a correct orientation of the MHC and TCR, and then to arrange the physical contact between the two molecules [47]. The CDR2s have an exclusive contact with the central parts of the MHC, and are probably highly relevant in the recognition of the MHC [48], while the CDR1s loops have a smaller contact with the MHC, and a contact with the terminal part of the peptides [49]. The

CDR1 of the α chain interacts directly with the N-terminal of the antigen, while the CDR1 in the β chain can interact with the C-terminal part of the peptide [47].

CDR1 and CDR2 loops are very rigid loops, showing little or no rearrangement for the binding process [48]. The binding between TCR and MHCs is mediated by specific "contact points" present on specific positions on both molecules. It has also been suggested that there are different "contact points" between the TCR binding the MHC class I, and the TCR binding the MHC class II [12]. This and other clues suggest that there could be two different binding mechanisms for CD4 and CD8 T-cell interactions and recognitions [48].

The CDR1 and 2 are considered fundamental for the docking of the two proteins, but they only have a minor contribution for the recognition of the peptide. What determines the stability of binding is the interaction between the peptide and the last CDR, the CDR3 [48].

The CDR3s are defined as the piece of the TCR sequence between the last conserved Cysteine at the end of the V segment, and the conserved FG[X]G motive in the J segment [44][50]. This means that the D region of the TCR β chain is present completely inside the CDR3 β segment [51]. The two CDR3s are positioned on the centre of the top of the TCR molecule, and the two loops form a pocket where the antigen can be allocated and interact directly with the TCR [52][53][54].

Formerly, because of the presence of the D region within the CDR3 beta, the CDR3 repertoire was considered to be the greatest source of variability. However, thanks to new crystal structures, this idea has been reconsidered, and the contribution to recognition is viewed as a product of equally-important and complex interactions between the TCR α and β chains [51].

While the CDR1 and 2 are two rigid loops, the two CDR3 have great flexibility.

They are able to create different conformations and to "adapt" to different antigens. This great ability makes it possible to extend the number of peptides that a TCR can potentially recognise [47][51][55][56].

However, if the flexibility of the loops increases the number of peptides recognised by a single TCR, what makes the entire TCR repertoire an effective system

for peptide recognition is the great variability of the CDR3 repertoire.

Because of this, it has been possible to conduct studies where the same peptide has been recognised by different CDR3s. These studies suggest that the interaction between CDR3 and peptides is mediated by a few hydrogen bonds (H-bonds), by the side-chain of CDR3 peptides, interacting with the backbone of the peptide, and one (or rarely more) charge–charge interactions with the peptide [56].

These observations lead to the conclusion that within the binding surface there are different parts with distinct contributions for the association and stabilization of the protein complex. Such hotspot-like positions and interactions can explain how the TCR can cross-react with many different types of peptides [57].

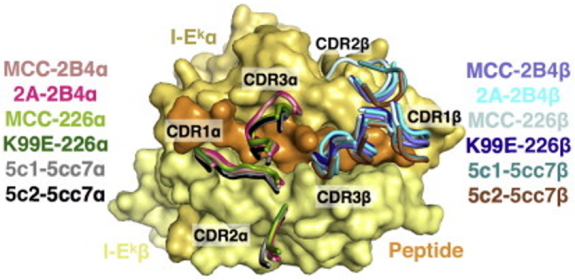
3.8 Cross reactivity

By cross-reactivity is meant the ability of a TCR to recognise different types of peptides and MHC combinations (cognate ligands). This is similar to the cross-reactivity that may happen when antibodies bind to different antigens [19][58].

The degree of cross-reactivity of a TCR, that is, how many different peptides can the receptor recognise, can vary greatly, and in the TCR it is generally very high. Each TCR can recognise hundreds of peptides, some estimations suggesting even thousands. It has also been suggested that this degree of cross-reactivity might be different between class I and class II MHC-specific TCRs [47].

The TCR cross-reactivity can be caused by the structural adjustment of TCR-MHC docking. For example, flexible CDRs loops can accommodate different peptides without altering the docking orientation [56][59]; the same TCR binds different peptide-MHC ligands using different docking orientations [60] or, alternatively, by molecular mimicry. Different peptide-MHC ligands can form very similar interfaces with partially identical amino acid sequences or structural similarities such as size, charge, or hydrophobicity at certain positions [49][61].

Cross-reactivity has been explained as a way of expanding the effectiveness of the TCR repertoire [62]. And because only by this phenomenon can the number of peptides being recognised increase by a large margin, this is viewed as a necessary



CDR3:

2B4	α	CAALRATGGNNKLTFGQG		2B4	β	CASSLNWSQDTQYFGPG	
226	α	CAAEPSSG--QKLVFGQG		5CC7	β	CASSLNNANSDYTFGSG	
5CC7	α	CAAE-AS-NTNKVVFGTG		226	B	-Identical beta chain-	

Figure 3.5: Three TCRs recognising the same peptide: This figure (originated by [47], top part from Figure 5C, sequences from supplementary material) illustrates the interaction between a singular peptide and three different TCRs and their CDR3s.
In the paper the authors investigate the TCR cross-reactivity to give a first-time measurement of it by identifying hundreds of peptides reactive with five different murine and human TCR.
In the picture above three human TCRs interact with the same peptide despite not identical CDR3s, with few different amino acid in the position directly interacting with the peptide (in red). This result reinforces the idea that the peptide recognition is driven by a small set of amino acids rather that the entire CDR3 sequence.

technique for the body, almost a "biological imperative" [47].

Recent studies have shown that cross-reactivity relies on a lock-and-key-like mechanism, where a minimal binding mechanism allows tolerance of a great variability of peptides, as long as the peptides conserve specific amino acids named "hotspots" [47][57]. A similar concept is present in [61], where it has been found that the peptides binding the same MHC need to share five amino acids to cross-react to the same TCR. For instance, for the I-Ab MHC class II molecule, the binding peptide has to conserve the positions (P2, P3, P5, P7, P8), with the P5 considered the most important because it interacts directly with the CDR3s [12].

Similar to the concept of hotspots on the peptides, the presence of conserved amino acid on the CDR3s has been observed for TCRs reacting to the same peptide. This is the case in [47], where a study concerning the crystal structure of different TCRs recognising the same peptide has seen the presence of conserved amino acids on the CDR3s of both chains (Figure 3.5).

The presence of hotspots on the TCR has led to the idea that motifs can exist within the CDR3s' pair structural sequence [47][57]. Searching for these motifs is a recurring theme in this thesis on the premise that by investigating these can understand the TCR interaction mechanisms.

Despite cross-reactivity being a necessity for the immune system, this has also been correlated to the induction of autoimmune diseases [19][56][57][58][61]. An autoimmune disease occurs when the adaptive immune system recognises as "non-self" an epitope that actually belongs to the host.

3.9 Public and private sequences

Another important phenomenon observed among TCR repertoires is the presence of shared (or "public") sequences among many individuals of the same population, while other TCRs are unique to single individuals ("private") [41].

The frequency with which private sequences appears is not correlated with the relative abundance within the populations. Indeed, widely spread sequences in a population can simultaneously be rare in each individual. However, it has been seen that identical clonotypes present in more individuals often respond to the same pMHC antigen epitope [63].

The causes of the presence of public sequences are not very well understood. And it is not clear if there are evolutionary advantages to this phenomenon. However, we do know that the presence of public or private sequences is not influenced by the environment, but is largely determined by the characteristics of the naïve repertoire [64].

The uniqueness of the TCR sequences can be explained by the propensity of the V(D)J recombination to develop a TCR repertoire that is as variable as possible, while the causes of the shared sequences are harder to uncover. In the literature, two models have been proposed: the recombinational biases and convergent recombination [63][65][66].

The recombinational biases consider that V(D)J recombination is not totally random, and that some combinations of regions are preferred to others. Considering

the V(D)J recombination as a pseudo-random event, therefore, means that over time the same sequences are more likely to be present, and consequently to be shared.

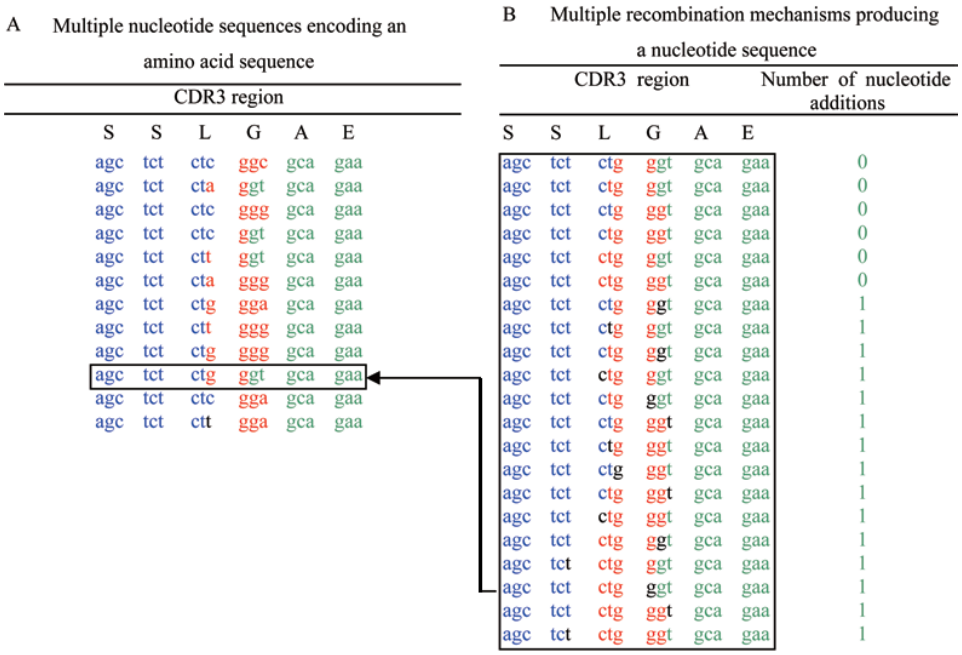


Figure 3.6: Convergent recombination: Picture originated by [65] (Figure 1). In this paper the authors investigated the role of public TCR in the population.
They explain it as coused by the convergent recombinational in which multiple recombination events "converge" to produce the same nucleotide sequence and/or multiple nucleotide sequences encode the same amino acid sequence.
Indeed, we can see this from the above picture, where multiple recombination mechanisms (**sub-fig. B**) can produce equal nucleotide sequence and small difference can still produce the same CDR3 (**sub-fig. A**).

Multiple rearrangements can converge to produce different nucleotide sequences, but these nucleotide sequences converge to encode the same amino acid sequences [63]. In other words, the redundancy of the nucleotide triplets encoding the same amino acid is considered to be one of the leading forces in the production of sharing sequences [41][64]. In Figure 3.6 (source [65]), it is seen how equal sequences of CDR3 are originated by different nucleotide sequences, showing how the convergent recombination provides the mechanistic basis for public TCR between individuals [67].

These two models will be useful later in this thesis, when we will analyse the common features expressed by the CDR3s.

The concept of private/unique sequences can be also applied to the CDR3 se-

quences. Because in this thesis I am focusing on CDR3 β sequences, future references to this concept are solely related to these sequences.

Part II

Results

Chapter 4

Quantitative analysis of the CDR3 dataset

4.1 Overview of the repertoires

The CDR3 database derives from two experiments with a similar protocol performed by Professor Friedman and colleagues in the Department of Immunology, Weizmann Institute, Rehovot, Israel. The sequence files are available at `http://www.ncbi.nlm.nih.gov/sra/?term=SRP075893`.

A total of 37 different C57BL/6 mice were sacrificed at different time points, of which 28 were immunised with one or two different antigens, while the remaining nine were left unimmunised and used as the control group. Details of immunisation are present in [68] and also in [16] [69].

The C57BL/6 mice also known as "C57 black 6", "C57" or "Black 6", is one of the most common inbred strain of laboratory rodent [70]. These are common in different areas of research including cardiovascular biology, developmental biology, diabetes and obesity, genetics, immunology, neurobiology, and sensorineural research [71]. Within inbred strains of mice like C57BL/6, all individuals of the line are nearly genetically equal to each other.

This is a great advantage for our experiment because we can test the effect of different antigens on "copies" of the same mouse, with the assumption that the different responses would not be affected by genetic factors.

All 28 immunised mice were immunised with freeze-dried Mycobacterium tuberculosis H37RA in water/oil emulsion, named Complete Freund's Adjuvant (CFA). Of these 28, 15 were also immunised in combination with another antigen, ovalbumin (OVA) [72]. For this experiments the peptide used from ovalbumin was "OVA 323-339" `ISQAVHAAHAEINEAGR`.

Freund's Adjuvant (named after Jules T. Freund) is a solution of antigen emulsified in mineral oil. It exists in two forms, complete and incomplete. The complete form is composed of heat-killed and dried Mycobacterium tuberculosis in non-metabolizable oils (paraffin oil and mannide monooleate), whereas the incomplete form (IFA or FIA) lacks the mycobacterial components (hence just the water in oil emulsion) [72][73].

Ovalbumin (OVA) is the main protein found in egg white, making up 54% of the total protein content. Ovalbumin and albumin were some of the very first proteins to be studied, the first crystallization of OVA having been recorded in 1890 [74]. The ovalbumin protein of chickens consists of 385 amino acids, and its relative molecular mass is 45 kDa [75].

Of the entire database of 37 mice CDR3 repertoires, 24 mice come from a first experiment composed as follows: six immunised mice were sacrificed five days after the infection event, six immunised were sacrificed at day 14 and finally six immunised and were sacrificed at day 60, with six unimmunised mice as a control. For each time point group of mice, half were immunised with only CFA and the other half with CFA plus OVA.

The second experiment involved: five mice sacrificed at day 7 and five mice sacrificed at day 60. For each time point, three mice were immunised with CFA plus OVA and two with only CFA, plus three unimmunized mice, for a total of 13 mice.

From all the mice, the spleen was isolated, and the TCR β chain from $CD4^+$ T cells sequenced and then analysed with the software Decombinator [76]. Decombinator is a five-item identifier that uniquely and unambiguously identifies any Illumina short-read sequence data for individual TCRs, identifying their regions and

CDR3s.

In summary, our CDR3 database is composed of sequences from 37 different mice.

1. Nine control mice, of which: six from the first experiment (1^{st}) and three from the second (2^{nd}).

2. Six day 5 (1^{st}), of which: three CFAs, three CFA+OVAs.

3. Five day 7 (2^{nd}), of which: two CFAs, three CFA+OVAs.

4. Six day 14 (1^{st}), of which: three CFAs, three CFA+OVAs.

5. Eleven day 60, of which: six (1^{st}), five (2^{nd}), five CFAs, six CFA+OVAs.

For a total of nine controls, thirteen mice sacrificed at an "early" stage, and eleven at a "late" stage. Thirteen immunised with only CFA, fifteen with CFA plus OVA. See Table 4.1.

		First Experiment		**Second Experiments**	
Control		6		3	
Immunised with		CFA	CFA+OVA	CFA	CFA+OVA
Sacrificed at day	5	3	3		
	7			2	3
	14	3	3		
	60	3	3	2	2

Table 4.1: The CDR3 repertoire database used in this thesis. 37 mice have been sacrificed and here is reported their distribution according to the antigen used and the time they have been sacrificed. From left to right: There are 9 control mice (6+3), 6 from Day 5, 5 Day 7, 6 Day 14, 11 Day 60. Of which 13 immunised with CFA, 15 with CFA+OVA. Abbreviations: CFA = Complete Freund's Adjuvant; OVA = Ovalbumin.

Information about the sequencing procedure are present in [68] and its Supplementary Material. Extensive pre-processing and error correction analysis of the raw reads can be found in [77].

4.2 Terminology Adopted

In this thesis are the reports of many different experiments performed with different combinations of mice repertoire, based on the time they were sacrificed, or on the antigen they were immunised with.

The repertoires from the first experiment are referred to as "Group A", while the second group of mice are called "Group B". The mice immunised with CFA are referred as CFA mice. The mice immunised with CFA plus OVA are referred to simply as OVA.

4.3 Analysis of repertoires

In the following sections, there is an analysis of and a discussion about the CDR3 database used in this thesis concerning its principal features. These include number and length of sequences, and compositions of the different mice repertoire. Because all of the experiments were performed on this database, I considered it worth spending some time thoroughly exploring its features, evaluating it, and comparing our findings with what it is reported in the literature.

4.3.1 Number of sequences

As I stated in the introduction, the number of sequences present in the body is one of the main subjects of discussion in the literature regarding T-cell and CDR3 repertoire. Here, I compare my findings with what is present in the literature regarding CDR3 in murine peripheral blood.

4.3.1.1 Total number of sequences

The total number of sequences in our database is $8 \cdot 10^7$, with a mean value of $2.1 \cdot 10^6$ and standard deviation of $2.6 \cdot 10^6$.

From Figure 4.1 and Table 4.2 we can also see that the values can vary from different mice.

In general, we can see that the average number of sequences per repertoire from the mice from the second experiment outnumbered 10-fold the number of sequences from the first experiment.

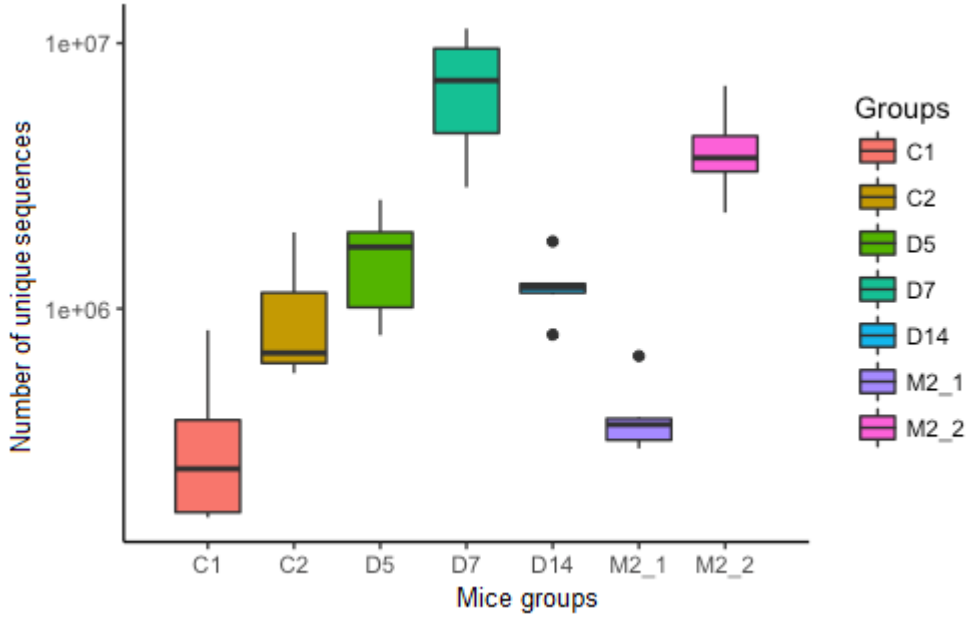


Figure 4.1: Number of unique sequences per mouse group: This box plot graph shows the different presence of sequences with each group of mice. As we can see the number of sequences can vary greatly, from half a million of control 1 (C1) to several millions of Day 7 (D7). In general, it is considered that a value of one million sequences or higher can give a good representation of the internal variability of the mice of origin [36]. In our case we have a low number of sequences for C1 and M2_1. These might not be considered enough for a good representation of the mice CDR3 repertoire. Legend: C1 = Sequence from Control mice from the first experiment; C2=Control mice second experiment; D5= Day 5; D7= Day 7; D14= Day 14; M2_1= Two months, first experiment; M2_2= Two months, second experiment.

Indeed, despite the first experiment having more mice, the actual number of sequences is 2-fold lower: in the first experiment are present $2.1 \cdot 10^7$ sequences, in the second experiment $5.9 \cdot 10^7$.

The number of sequences per mouse is shown in Figure 4.2 and Figure 4.3. In Figure 4.2, each bar represents the number of sequences present in each mouse, while the colour represents whether the mouse repertoire comes from the first or second experiment.

From Figure 4.3, we can see the number of total sequences among the mice, coloured per the antigen infection. The total number of sequences for the control mice is $5.2 \cdot 10^6$, for the OVAs $4.3 \cdot 10^7$ and CFAs $3.2 \cdot 10^7$.

From the previous figures and tables, it is clear that the sequence originating

	Whole	C1	C2	D5	D7
Mean	$2.1 \cdot 10^6$	$3.4 \cdot 10^5$	$1 \cdot 10^6$	$1 \cdot 10^6$	$7 \cdot 10^6$
SD	$2.6 \cdot 10^6$	$2.5 \cdot 10^5$	$7.5 \cdot 10^5$	$6.8 \cdot 10^5$	$3.4 \cdot 10^6$
	D14	**M2_1**	**M2_1**	**1st**	**2nd**
Mean	$1 \cdot 10^6$	$3.9 \cdot 10^5$	$4.1 \cdot 10^6$	$8.9 \cdot 10^5$	$4.5 \cdot 10^6$
SD	$3.2 \cdot 10^6$	$1.3 \cdot 10^5$	$1.7 \cdot 10^6$	$6.6 \cdot 10^5$	$3.3 \cdot 10^6$

Table 4.2: Number of CDR3 sequences per group of mice: Summary table of the mean and standard deviation of sequences present in all mice per mice groups and per experiment-origin. From this table we can see that the amount of sequences present in the entire database (Whole) can vary by one order of magnitude depending from which time point the mice originate. We can see that from the second experiment we have a higher number of sequences while from the first this value drops to few hundred thousand of sequences. Label: Mean: Average value; SD; Standard deviation. Groups legend: Whole = The entire CDR3 database; C1 = Sequence from Control mice from the first experiment; C2=Control mice second experiment; D5= Day 5; D7= Day 7; D14= Day 14; M2_1= Two months, first experiment; M2_2= Two months, second experiment; 1st= All sequence from the first experiment; 2nd= All sequence from the second experiment.

from the second experiment is disproportionate. This is not an ideal scenario, but it can be solved by applying various methods of normalisation and standardisation.

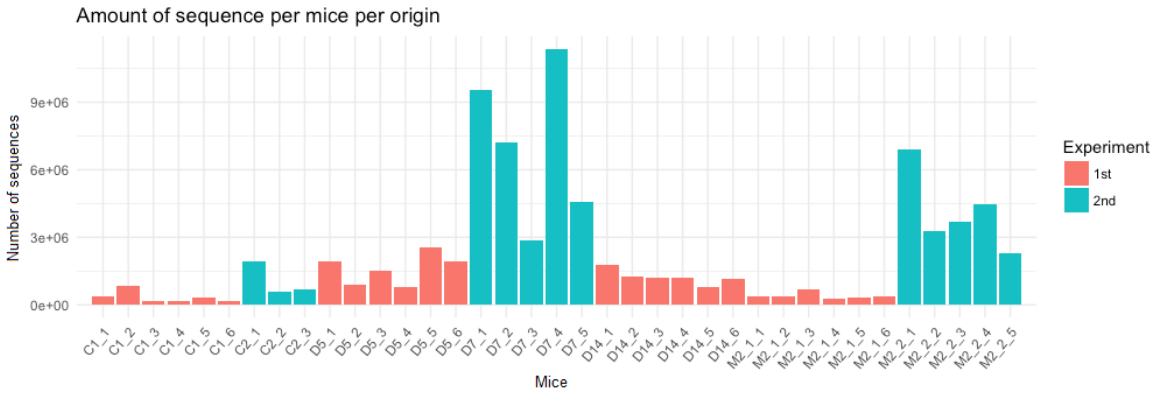


Figure 4.2: Amount of sequences per mice by origin: With this plot, I represent the great numerical difference between the number of sequences from the two different experiments. The second experiment has almost the same amount of sequences as all the mice from the first, despite fewer mouse repertoires. This gives an unbalanced representation for the two sets of mice. Interestingly, the amount of control sequences from the second experiment are much lower, increasing the overall amount of control sequences, but not by much. Legend: 1st= Sequences from the first experiment; 2nd= All sequence from the second experiment. On the x-axis all the mice of the database, labelled: C1 = Control mice from the first experiment; C2=Control mice second experiment; D5= Day 5; D7= Day 7; D14= Day 14; M2_1= Two months, first experiment; M2_2= Two months, second experiment. y-axis= Number of sequences.

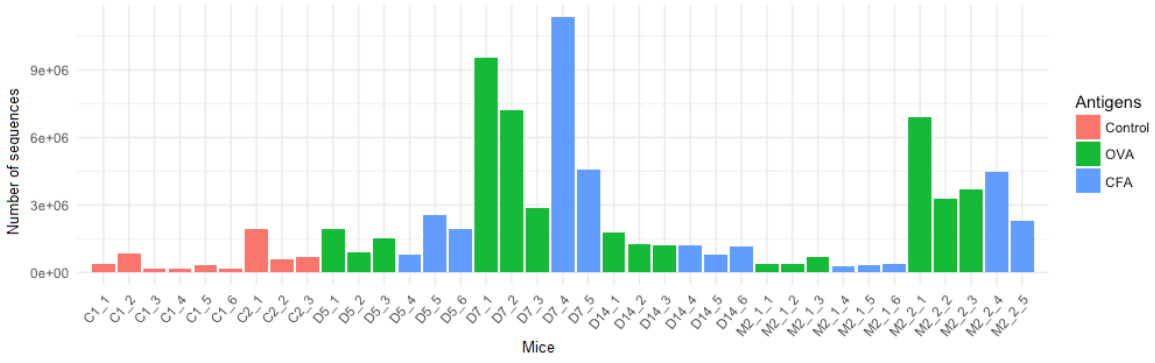


Figure 4.3: Amunt of sequences per mice per antigen: In this plot, similar to Figure 4.2, I represent the number of sequences per mouse in relation to the antigen used in the immunisation. Here the number of OVAs and CFAs sequences are more balanced, and the two datasets are more alike with the number of OVAs of $4.3 \cdot 10^7$ and CFAs $3.2 \cdot 10^7$. Groups as in Figure 4.2. Legend: Control=the control mice; OVA = Mice immunised with the Ovalbumin; CFA = Mice immunised with the Complete Freund's Adjuvant. x-axis and y-axis: as in Figure 4.2.

4.3.1.2 Number of unique sequences

In this section, we want to focus instead not on the total number of sequences, but on the number of unique sequences.

In Table 4.3, Figure 4.4 and Figure 4.5, the value of unique sequences per mouse, by group of mice are shown.

We can still see that the sequences are disproportionate, but the level of disproportion is less pronounced, and the average values are in line with the literature.

	Whole	**C1**	**C2**	**D5**	**D7**
Mean	$2.4 \cdot 10^5$	$8.2 \cdot 10^4$	$2.1 \cdot 10^5$	$3.2 \cdot 10^5$	$3.9 \cdot 10^5$
SD	$1.4 \cdot 10^5$	$5.1 \cdot 10^4$	$4.6 \cdot 10^4$	$1.4 \cdot 10^5$	$1 \cdot 10^5$
	D14	**M2_1**	**M2_1**	**1st**	**2nd**
Mean	$2.6 \cdot 10^5$	$1 \cdot 10^5$	$3.5 \cdot 10^5$	$1.9 \cdot 10^5$	$3.3 \cdot 10^5$
SD	$5.4 \cdot 10^4$	$2.2 \cdot 10^4$	$1.3 \cdot 10^5$	$1.3 \cdot 10^5$	$1.2 \cdot 10^5$

Table 4.3: Number of unique CDR3 sequences per groups of mice: Mean and standard deviation of the unique sequences present in all mice, per different mice group and per experiment-origin. This table is similar to Table 4.2, but while in Table 4.2 we saw a great variation in the total number of sequences, here the number of unique sequences is more uniform with all values around 10^5. This implies that a great number of sequences in the database are copies, and the number of unique sequences is more uniform among our repertoires. Legend as in Table 4.2.

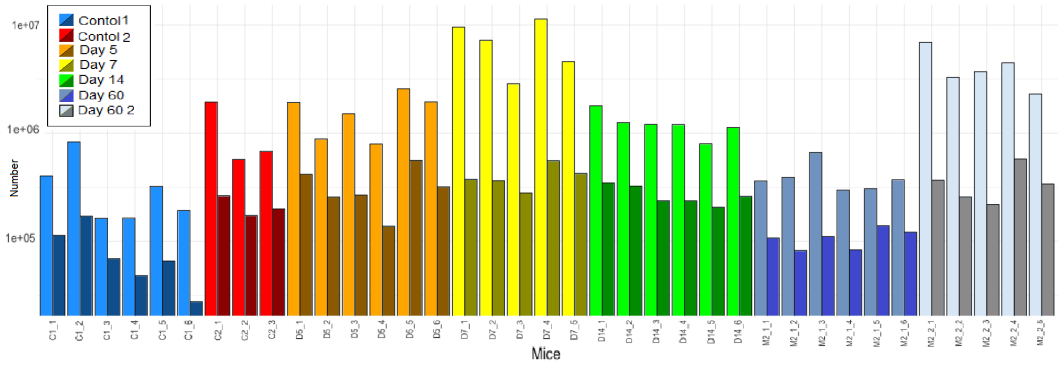


Figure 4.4: Number of sequences for each CD R3 repertoire: Here, I represent the number of total sequences (lighter colour bars) side by side with the number of unique ones (darker colour) in a *y*-axis log scale. We can see that the number of sequences varies profoundly from mouse to mouse, while the unique sequences are more balanced. Groups colour coding explained in the legend inset in the top left corner.

In Figure 4.4 the value per mouse is presented, with the total number of sequences and the number of unique ones. We can see that the variation is smaller, and all mice tend to have a more similar number of sequences.

In Figure 4.6, the size of the repertoire versus the unique sequences is plotted. As we can see, an increase in the number of sequences is related to a higher number of unique sequences, until they reach a plateau. To reach such a plateau at least a million of such sequences is required, as already claimed in [36]. After this value, the number of unique sequences grows much more slowly.

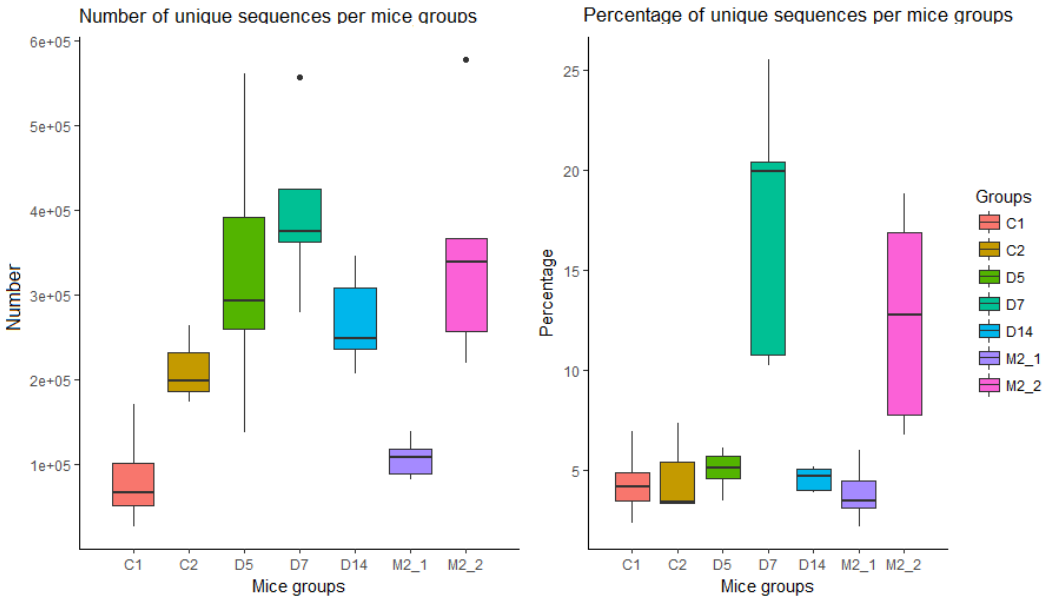


Figure 4.5: Amount of unique sequences and percentage per mice group: In the left-hand side box plot are reported the of unique sequences present in all mice. We can see that the number of sequences can vary greatly from group to group, from few hundred thousand in C1 to several million in D7 or M2_2. However, the percentage of unique sequences is alike in all repertoires as reported on the right-side box plot. In detail: the mean of the percentage of unique sequences is C1=4.3%, C2=4.6%, D14=5%, D5=17.3%, D7=4.5%, M2_1=3.8%, M2_2=12.6%. Groups legend as in Figure 4.1.

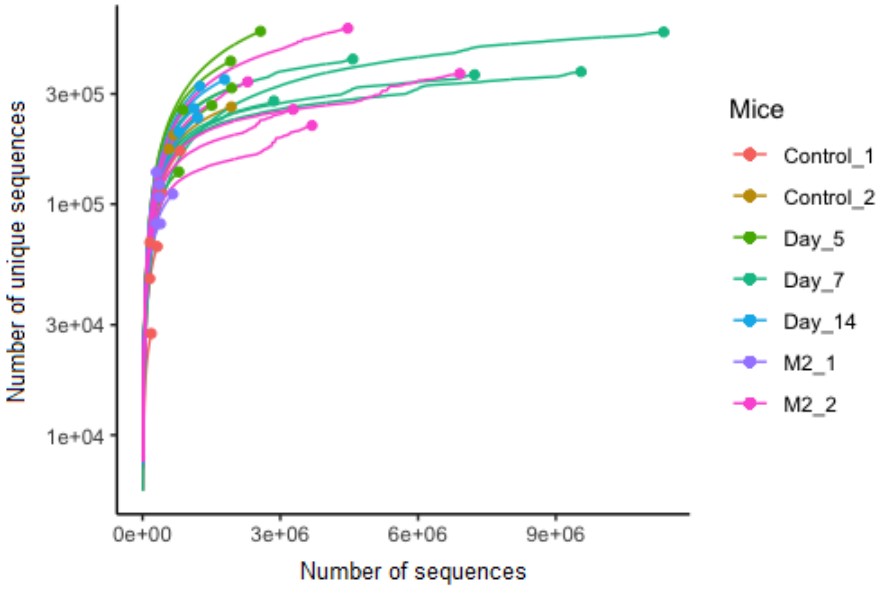


Figure 4.6: Size of repertoires versus number of unique sequences: From the analysis of the repertoires we noticed that the percentage of unique sequences does not vary greatly, and (except for D7 and M2_2) it is around 4-5% of the overall sequenced. To better illustrate this behaviour, I created the above plot: from each mouse, I take a fix number of sequences, and I check the progression of unique sequences versus the total number, and I repeat it until the end of the repertoire. With this system we can see the increase of unique sequences at the increase of the repertoire. The number of unique sequences (y-axis) grows fast until reaching a plateau of half a million sequences when the repertoire size reaches 1-2 million of sequences (x-axis). Groups colour coding explained in the legend inset in the bottom right corner.

4.3.2 Length of CDR3 sequence

The average length of the entire CDR3 dataset is 16.31 (see Figure 4.7), with a variance of 3.55 and a standard deviation of 1.88.

In Table 4.4 the percentage of sequences with the most abundant length is reported.

Length	$\leq$ **14**	**15**	**16**	**17**	**18**	$\geq$**19**
Percentage	11.15	13.85	23.31	27.88	17.21	6.6
		82.25				

Table 4.4: Percentage of sequences length: Percentage of the most abundant sequence lengths expressed in nucleotides (nt). In this table, I report the distribution of CDR3 sequence length. We can see that the 82% of all sequences are distributed in four values from 15 to 18 nt, with the relative majority (median) of sequences having length 17 nt, but a mean value of 16.31 nt (not reported in the table).

Results are consistent with many other works [6][78][79][80][81][82][83].

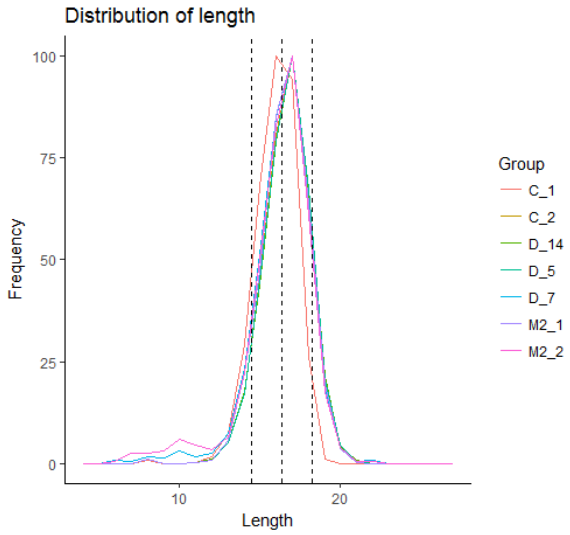


Figure 4.7: Distribution of lengths of CDR3 repertoires: In this graph is reported the distribution of sequences' length through the database. The distribution is a thin bell shape with practically all sequences inside one standard deviation far from the mean. The central dotted line is the mean of all length of sequences (16.31), and the side lines are one standard deviation. Also, the overall distribution is left-skewed (Negative Skewness) with mode and median equal to 17. Groups colour coding explained in the legend inset in the right hand side.

4.3.3 Jaccard index

The Jaccard index (JI) [84] is a coefficient measuring similarity of finite sample sets. It is defined as the size of the intersection divided by the size of the union of the sets, see Equation 4.1:

$$J(A,B) = \frac{|A \bigcap B|}{|A \bigcup B|};\ 0 \geq J(A,B) \geq 1 \tag{4.1}$$

The JI ranges from 0 to 1, in which the higher the value, the more similar the two sample sets.

This formula computes the similarity among repertoires. The dissimilarity between the sample sets (also known as Jaccard distance) is the complementary of the JI.

In this case, I computed the proportion of shared sequences among all mice. This is reported in Figure 4.8.

All the results are consistently low, not higher than 0.268, and with an average value of 0.135.

We could expect that nearly genetically identical mice, immunised with the same antigen would develop similar immune repertoires. But if we look at the JI per antigen, the results are still low: JI per Controls is: 0.126; OVAs: 0.137; CFAs: 0.139.

Therefore, as already claimed in [16] and [77] the immunization event alters the repertoire states, but it does not drive a repertoire convergence.

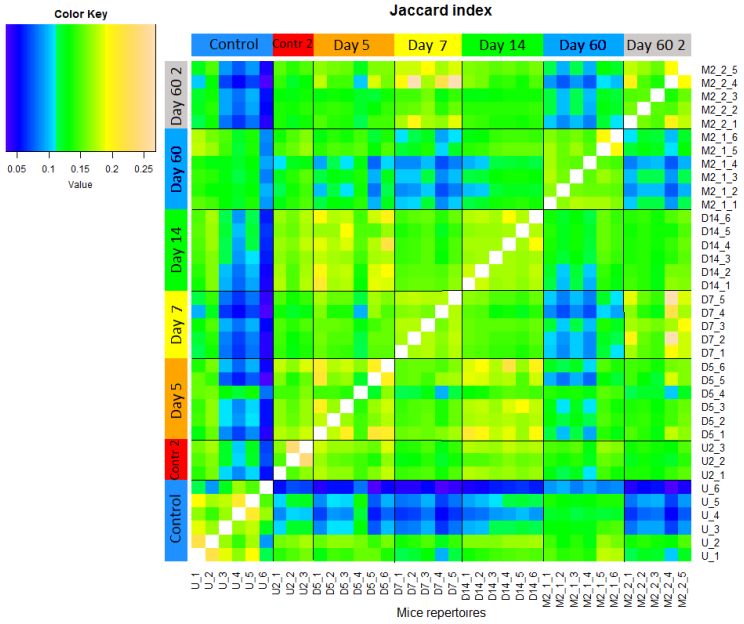


Figure 4.8: Heat-map of the Jaccard index of all repertoires: In this heat-map plot I represented the JI of each mouse against everyone else. The JI is a measure of similarity between repertoires and with this plot we can actually see that: 1) the group Control 1 has the lowest average JI against all other groups. However, this characteristic is not present in the Control 2 group. 2) The highest values are between the group D5-D14 and D7-D60_2. In theory these groups should have very little in common and worryingly the only thing that they share is the experiment of origin (second experiment). 3) Low values also for D60 and D60_2, while we would have expected high values, considering the same type of experiment and time point. The diagonal values (identity values) are excluded. Blue, green and yellow indicate progressively higher Jaccard index, i.e. higher dissimilarity between sample pairs.

4.3.4 Gini coefficient

The Gini coefficient [85] is a measure of statistical dispersion of a frequency distribution. The value of the Gini coefficient falls within a range between 0 and 1, where 1 is a perfect inequality and 0 is a uniform distribution. It is used primarily to study population, in multiple fields from economics to immunology [86].

The Gini coefficient is computed as half of the relative mean difference, see Equation 4.2:

$$G = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right|}{2\sum_{i=1}^{n}\sum_{j=1}^{n}x_j} = \frac{\sum_{i=1}^{n}\sum_{j=1}^{n}\left|x_i - x_j\right|}{2n\sum_{i=1}^{n}x_i} \tag{4.2}$$

In Equation 4.2 the numerator is composed of the sum of the absolute value of the difference of all elements x of a set X. The denominator is the sum of X times double the number of elements in X.

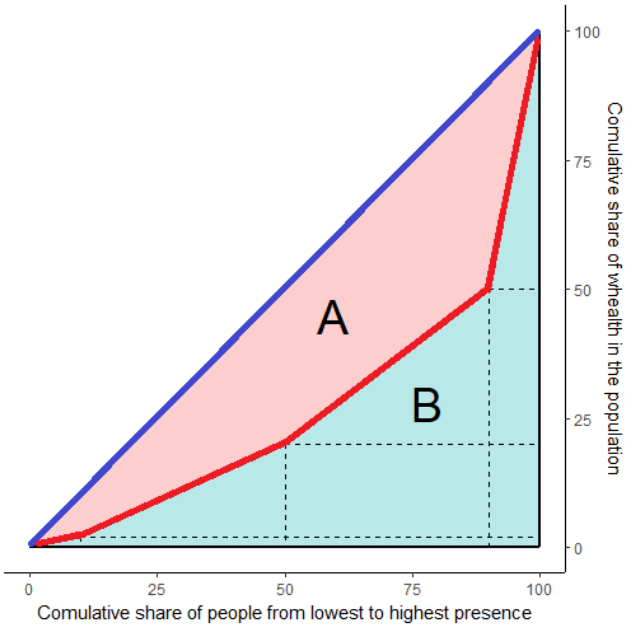


Figure 4.9: Graphical representation of the Gini coefficient: In this graph, we can see that the cumulative share of population creates a line in red, called the Lorenz Curve. Such a curve can only be equal to or lower than the uniform distribution line in blue. The further the Lorenz curve is from the uniform distribution line, the lower the Gini coefficient. The Gini coefficient can be also computed as $A/(A+B)$. The area A is the area between the uniform distribution line and the Lorenz curve. B is the area under the uniform distribution line minus the area A.

Let us take for example a population in which the percentage of total wealth is distributed as follows: 2% of wealth for the poorest 10%, 18% for the 40%, 30% for another 40%, and 50% for the remaining richest 10%. The Gini Coefficient applied Equation 4.2 would be 0.25. This low value is not a surprise considering that half of all the wealth belongs to 10% of the population.

A representation of this example is in Figure 4.9: we can see that the Gini coefficient can be also computed as the difference of the area between the uniform distribution line and Lorenz curve and all the area under the uniform distribution line.

$$Gini\ Coefficient = \frac{A}{A+B} \tag{4.3}$$

For $A = 0$ the Lorenz Curve is equal to the uniform distribution line.

My aim with this coefficient is to see if, in each mouse repertoire, there are sequences with a high frequency, with the assumption that an over-represented sequence might be relevant to the immune response.

I computed the coefficient using the `R` package `ineq`.

The average Gini coefficient of our repertoires is 0.65, with a standard deviation of 0.08. This means that there is a small number of sequences with a very high frequency. However, the number is not overwhelming. Moreover, the Gini coefficients of the control groups are no different from the immunised mice. Therefore, we can claim that the immunisation event is not a driving force pushing the immune response towards an over-representation of one or a few sequences.

In Figure 4.10 the Lorenz Curve for all mice is reported, coloured according to their group. Each line is somewhat far from the uniform distribution line, indicating a high Gini Coefficient value as a result. Indeed, if we look at the bottom green line, this mouse is a D7 mouse and in Figure 4.10 we see the same outline in Group D7 (yellow).

From Figure 4.10 it is possible to regard the Gini Coefficient value represented as a box plot for all mice groups. We can see that all repertoires have a dissimilar value, which could vary greatly. Look at example groups D60, in which the first

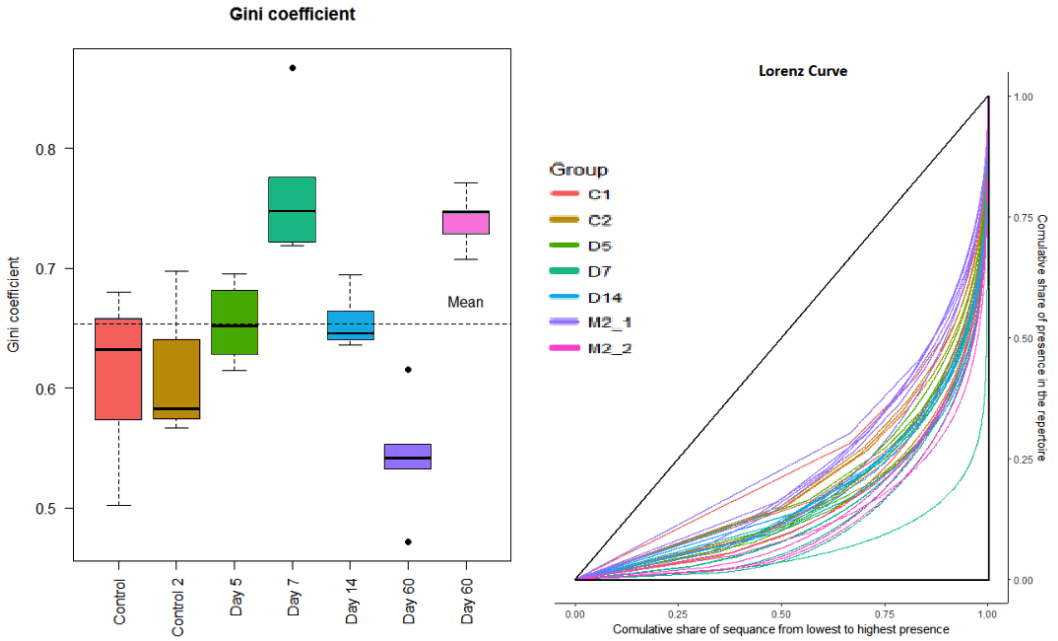


Figure 4.10: Gini coefficient of our repertoires: *Left-side plot*: Box plot for the Gini coefficient of our repertoires. In the plot each group of mice is reported in a different colour, the dash line being the mean Gini value of all repertoires together. We can see that the value is not constant, and it can change greatly. *Right-side plot*: Lorenz curve of our repertoires: here is represented the Lorenz curve for all our repertoires. Each line represents a different mouse and is coloured according to its group. All curves are far from the uniform distribution line, varying and reflecting the Gini coefficient. High Gini coefficient value, as for D7 and Day60_2, do not imply a low level of variability but rather that a few numbers of sequences are overrepresented. It is not easy to determine if such overrepresentation is due to an issue in the sampling or sequencing process, or of an effective relevance in the mice immune response prior to the sample collection.

group has a level around 0.55 and the second around 0.75. It is hard to figure out the cause for this discrepancy.

Indeed, it seems that the Gini Coefficient is higher when the repertoires are bigger. This is confirmed by the following plot, where the Gini coefficient is put in relation to the repertoire size.

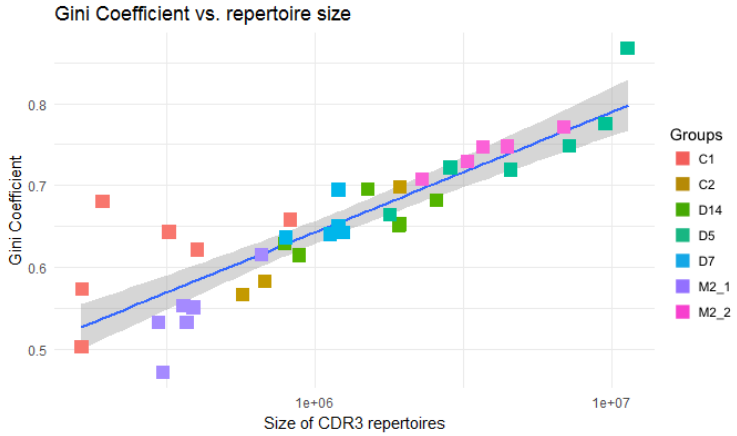


Figure 4.11: Gini coefficient in relation to repertoire size: With this plot I assessed if the Gini coefficient value is related to the size of each repertoire. We can see that the bigger the repertoire, the higher is the Gini coefficient. Because the number of unique sequences stop growing after one million of sequences, while the Gini coefficient keeps growing with the size of the repertoire, is my understanding that the high Gini value is not due to an overrepresentation of sequences inside the mice but to errors in the analysis process.

In Figure 4.11, we can see that the Gini coefficient value grows with the amount of sequences. Theoretically, the Gini coefficient should not change in relation to the amount of sequences sequenced. The sequencing process should give us the same percentage of type of sequences: for 1,000 or 1 million sequences we should have approximately the same value.

This is due to a few sequences with a lot of replication in those repertoires, probably caused by technical problems in the library production.

Support Vector Machines as classification method for CDR3 repertoires

Chapter 5

Support Vector Machine

5.1 Introduction

Support Vector Machine (SVM) is a supervised learning model [87] used for data classification and regression analysis. It is one of the main machine learning methods used in modern day artificial intelligence, and it has spread widely in many fields of research, not least in bioinformatics.

SVM can be used for regression and classification analysis. In this section, I will describe and show the use of the latter.

Programming languages like `R` or `Python` offer several libraries to compute and work with SVMs in a simple and flexible way. For these reasons, SVM is the principal classification method used in this thesis, and for the first two years of my PhD my projects and tests revolved around this method.

At the beginning of the first year of my doctorate, I started working on the research made by N. Thomas and presented in the paper [16] and in his thesis [88]. My initial task was to repeat his work, continue the investigation on triplets, and extend the classification to other CDR3 repertoires. Given the interesting results arising from this work, this soon became my main task, and the objective of the thesis you are now reading.

The application of SVM for the classification of CDR3 is not an immediate action, and its application required a series of steps. The prime issue is that the CDR3 repertoires are composed of strings of letters: therefore, we need to have

converted them into a suitable numerical format.

To make it possible, in [16] four different bioinformatics methods are used, those being:

1. The **bag of words and *k*-mers**: we create all possible combinations of amino acids of different length, such as duplets (2-mers) and triplets (3-mers), and by counting their presence within the repertoire we form a codeword for that repertoire.

2. **Numerical factors** with this, we convert the codeword into numerical vectors and with

3. ***k*-means algorithm** we gather them in a numerically smaller number of features.

4. And finally, the **Support Vector Machine** for the classification of CDR3 repertoire.

In this section I present my work with the core ideas in [16], my results, the pros and cons of the methods, and the extension of these to new repertories. Before describing my findings, however, it is worth providing an introduction to these methods, starting from the most important, the SVM, and proceeding with the others.

5.2 Support Vector Machine

5.2.1 History

The idea behind the Support Vector Machines (SVM) is to be able to find a separating line between two classes of elements as points in space. This concept was proposed and designed by Vapnik and Lerner in 1963 [89], wherein the idea of maximum-margin hyperplanes was first introduced.

The method was further developed over the following 30 years by Vapnik and other researchers, gaining greater consideration during the conference COLT 1992 [90]. The modern day SVM method was completed in the following five years, with

the application of the kernel trick to find the maximum margin hyperplane for the nonlinear separable space and the implementation of the soft margin in [91], later improved with [92][93][94].

5.2.2 Introduction

As mentioned in the previous section of this thesis, the SVM is a type of supervised machine learning, meaning the method needs a training set of labelled elements that will be used as a reference for assigning the class of unknown elements.

A supervised machine learning method needs to be "trained", and it goes without saying that the better the training set, the more accurate its prediction. The action of prediction means to assign a new, unknown element from what is called the test set to one of two classes, performing a binary classification.

In action, what the SVM really does is provide the optimal dividing line (or hyperplane) between the two classes of elements, mapped into an n-dimensional space (or hyperspace).

Any class of items, be they a house, a car, or a repertoire of CDR3 sequences, possess some set of features that can be sorted and measured. If we consider each feature as a dimension in a coordinate system, and the measures as coordinates, we can represent any item of our class as points in an n-dimensional space.

The optimal hyperplane is defined as the hyperplane most separating from the two classes: after it is found, it is possible to identify the class to whom a new element corresponds [95][96].

In summary, the SVMs can learn a decision function or hyperplane from a training set of examples (input) and perform a binary class recognition (output) of a new element or test set, using a non-probabilistic classification in a high dimensional feature space or hyperspace [97].

5.3 SVM: the concept

Let us consider two sets of elements with only two features.

For these examples, I choose to plot the data "iris" from R, for the flower type "setosa" and "versicolor".

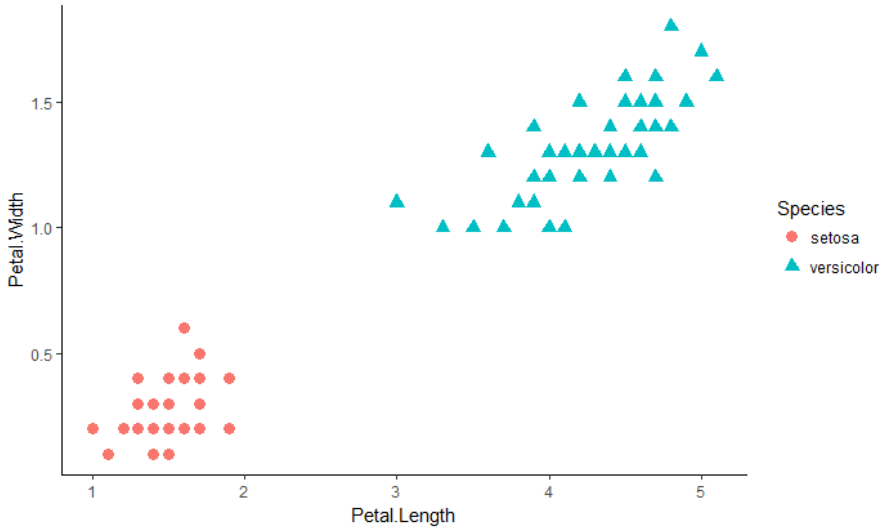


Figure 5.1: Example of two sets of data in two dimensions: Here are reported two species of the Iris flower data set, Setosa and Versicolor, plotted in relation to their length and the width of the petal. A large gap is present between the two groups and we can intuitively consider them as part of two groups.

The data are reported in Figure 5.1. We have got two different sets of flowers in relation to the length and width of their petals.

As you can see, there is a large gap of empty space between the two sets of data. We could easily draw an arbitrary line between the two sets and divide the space in two.

Now, in Figure 5.2, the plane is divided in two. Everything that is, or will be, present above the line would be classified as "versicolor", everything below the line as "setosa". Using this setup, we can easily classify the elements just by looking at where they would lie in the space.

The space between the points is called hyperspace, and the dividing line is a hyperplane. The hyperspace can be formed by any number of dimensions n, while the hyperplane has always one dimension less $n-1$. If the hyperspace is a line, the hyperplane would be a dot; for a plane, it would be a line (as for our example); for a 3D space, it would be a plane, and so on.

In Figure 5.2, we have now got a hyperplane between the two sets. However, this is not the only possible solution. In fact, an infinite number of hyperplanes can be drawn, as we can see in Figure 5.3.

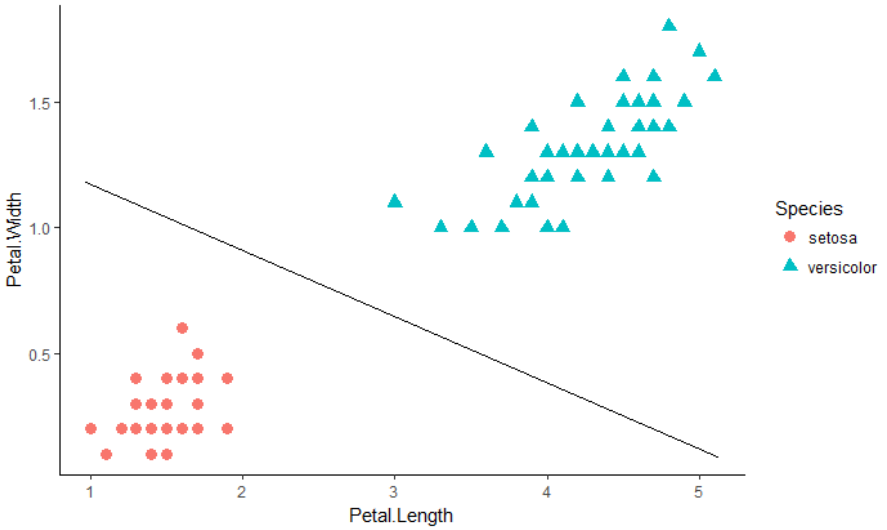


Figure 5.2: Arbitrary dividing line between two sets of data: As mentioned, we can intuitively divide the flowers into two distinct groups and we can choose to draw a separation line between them, aiming to divide the space between them into two equal spaces, so maximising the distance between the points and the line.

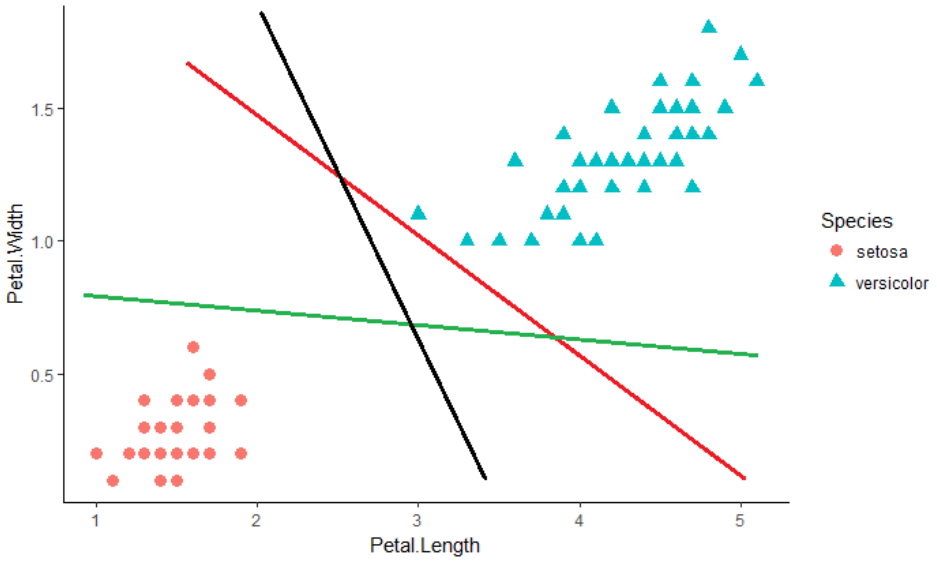


Figure 5.3: Infinite number of hyperplanes between two sets of data: The number of valid hyperplanes that can be drawn in a hyperspace is infinite. Any line, with any angulation and distance to the points counts as a hyperplane as long as the two set of points are in different hyperspaces.

An infinite number of lines are therefore possible, but we need only one of them, and possibly one that would give less misclassification and a better performance of any future prediction.

Therefore, which one is the best line? And how can we find it?

The answer to the first question is: the optimal hyperplane is defined by the maximum margin of separation between any training point and the hyperplane [96].

Therefore, among all infinite possible hyperplanes between two sets (Figure 5.4 A), the optimal hyperplane is the one with the highest distance to the margins (Figure 5.4 B).

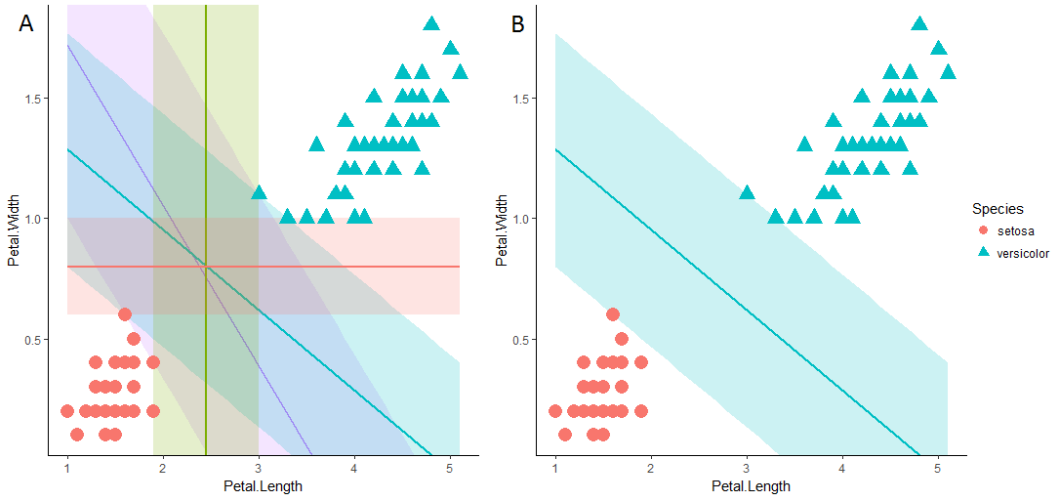


Figure 5.4: Examples of possible hyperplane: In sub-fig. A it is represented a selection of possible hyperplanes where their margins have been highlighted. From the vertical margins (light yellow) to the horizontal (light red), the possible hyperplanes are infinite. However, the optimal one has the largest margin between the points. **In sub-fig. B** I report the optimal hyperplane for this set of points. This is the hyperplane with the largest distance between the two margins, interacting only with the most external point of the classes, called the hard margin, in contraposition to the soft margin which we will see later.

Because the optimal hyperplane would be defined by its distance to the margins, and not by all data points, usually a small set of data points is enough to find the optimal hyperplane, as we can see in Figure 5.4. These relevant data points are referred to as support vectors.

As to the how, the answer is more complex, and we need to introduce some mathematics to this explanation.

5.3.1 Finding the best hyperplane

In our example, we have used two-dimensional data: length vs. width of flower petals. Because the hyperspace is two dimensional, the hyperplane must be a line.

The formula of a line is: $y = ax + b$, and this formula is not dissimilar to the general formula of the hyperplane, that is defined as:

$$w^t x = 0 \tag{5.1}$$

The left-hand side of Equation 5.1 can be considered as the inner product of two vectors. Indeed, when we are dealing with points in space, as in this case, it is useful to use the concept of vectors.

The introduction of the concept of vectors should not be a surprise, given the name of the subject. However, explaining the entire vector algebra and the mathematics would be needlessly long, and in this thesis, I am giving a minimal amount of information in order to understand the concept:

A vector is defined as any quantity with a magnitude and a direction.

In other words, a vector exists between the origin $O(0,0)$ and a point in the space.

The magnitude of a vector, x, is its length. This is usually written as $\|x\|$ and called the norm of the vector. For the point $P(3,4)$, the distance from $O(0,0)$ can be computed with Pythagoras' theorem and would be 5.

The direction of $P(3,4)$, is the vector $W\left(\frac{P_1}{\|P\|}, \frac{P_2}{\|P\|}\right)$.

In our case $\|P\| = 5$ therefore, the vector $W(\frac{3}{5}, \frac{4}{5}) = W(0.6, 0.8)$.

Using the definition of vector and the vector algebra we can measure the distance between our points and the origin, and from this compute the hyperplane.

When the two classes are linearly separable, it is possible to find two parallel hyperplanes with the distance between them as large as possible. These two hyperplanes are in contact with the elements of the two classes and they are called "margins". These types of margins are called hard margins

Because these two hyperplanes define the maximum margin possible, the hyperplane that lies halfway between them would be the best hyperplane.

These hard margins can be described by the equations 5.2:

$$\begin{aligned} w^t x + b &= 1 \\ w^t x + b &= -1 \end{aligned} \tag{5.2}$$

Consequently, we can write the hyperplane as $w^t x + b = 0$.

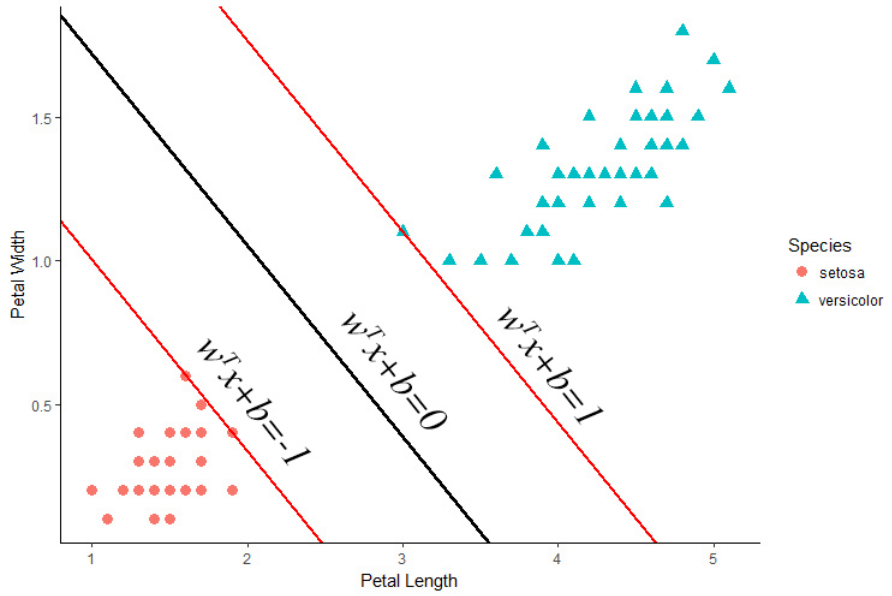


Figure 5.5: Formulas of the hyperplane: Here are reported an example of hyperplane line (black) alongside the two hard margins (red) and their formulas. The optimal hyperplane is the one with the highest distance between the margins, therefore the distance between the two margins and the hyperplane is the same (± 1) and the distance between the two margins is $\frac{2}{\|w\|}$.

The result is reported in Figure 5.5. There we have the hard margins in red and the hyperplane in black. Therefore, the best hyperplane is the one with the maximal distance m to the margins.

Because the distance from a point $p(x_0, y_0)$ to a line $ax + by + c = 0$ is:

$$\frac{|ax_0 + by_0 + c|}{\sqrt{a^2 + b^2}} \tag{5.3}$$

So, the distance between the line passing through the margin and the hyperplane is:

$$\frac{|w \cdot x + b|}{\|w\|} \rightarrow \frac{1}{\|w\|} \tag{5.4}$$

So, the distance between the two margins would be expressed as twice that, Equation 5.5:

$$m = \frac{2}{\|w\|} \tag{5.5}$$

To maximize the distance from the margins, we must minimize $\|w\|$.

If we add the constraint that no points should be present between the margins, we can rewrite Equation 5.2 as Equation 5.6:

$$\begin{matrix} w^T x + b \geq 1 \; when \; y = 1 \\ w^T x + b \leq 1 \; when \; y = -1 \end{matrix} \quad rewritten \; as \; y_i \; (w^T x + b) \geq 1, \; \forall \; i \; for \; y = \pm 1 \tag{5.6}$$

We can conclude saying that the best decision boundary would found by solving the following constrained optimization problem, Equation 5.7:

$$\begin{matrix} \max \frac{2}{\|w\|} \\ subject \; to \; y_i (w^T x + b) \geq 1 \end{matrix} \tag{5.7}$$

Coming back to our example seen in Figure 5.4, I isolate two possible hyperplanes, as reported in Figure 5.6:

In the case reported in Figure 5.6, the optimal hyperplane is the one on the right-hand side. Indeed, if we compute the distance between the hyperplane and margins using the formula for the distance of two parallel lines $\left(|b_2 - b_1| \sqrt{m^2 + 1} \right)$, the result is 1.03 for the left-hand side and 1.07 for the one on the right.

5.3.2 Soft Margin

The biggest limitation of this approach is that the method does not consider potential error due to outliers. In the previous example, a single outlier too close to the other class would have biased the margins. In addition, it is not always possible to find a clear line between two sets of points. For example, as we can see in Figure 5.7, it is not possible to draw a line that divides the two sets of data without assigning data

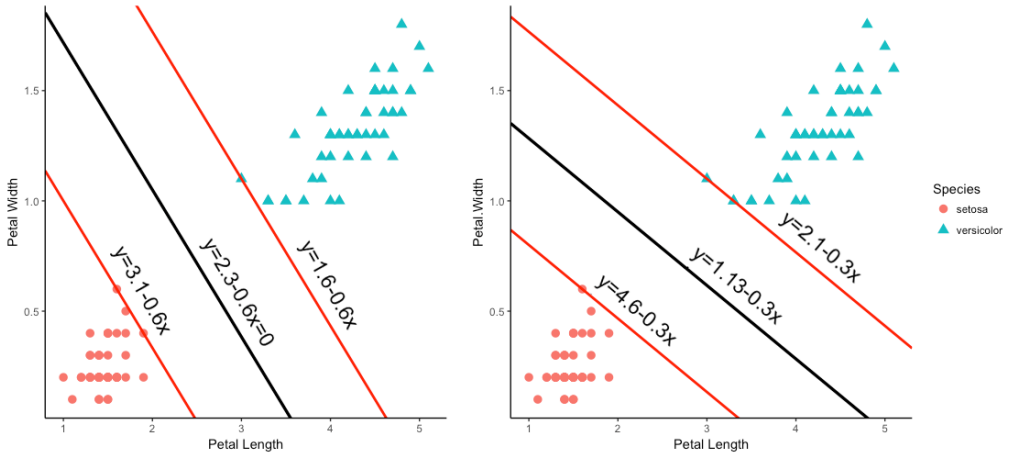


Figure 5.6: Example of two possible hyperplanes: Here two different hyperplane lines (black) and their margins (red) are reported. Also present is the formula of the line that they represent. Because in this example we work only with two dimensions, we can work out the distance between the margins and compute the optimal hyperplane. The distance between the margins of the left-hand side plot is 1.03 and on the right-hand side it is 1.07, therefore we can decree that the optimal hyperplane for this set of data is the one on the right-hand side.

points in the wrong hyperspace.

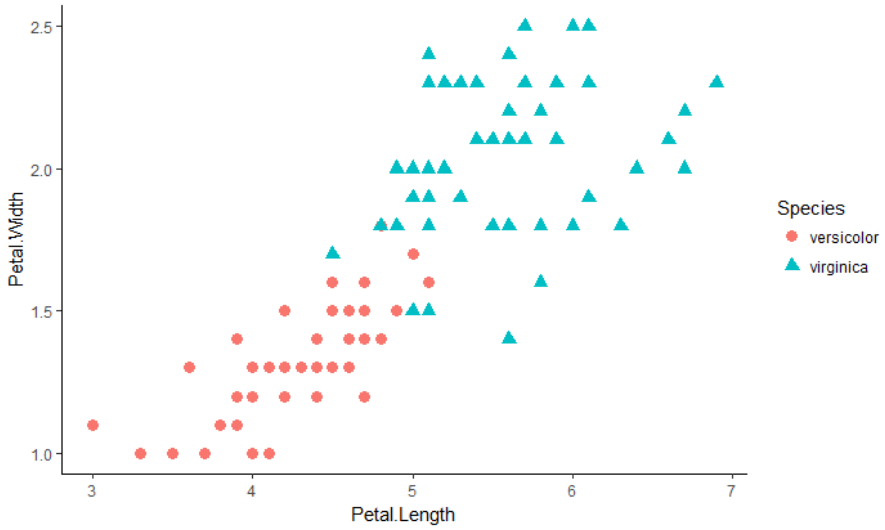


Figure 5.7: A non-linearly separable set of data points: In this plot I report another subset of the iris dataset, the group Versicolor and Virginica. Once plotted it is clear that is not possible to find a straight line that would separate the sets without leaving some points in the wrong side of the hyperplane.

For these reasons, these kinds of margin are called hard margins. To solve this problem, the concept of the soft margin has been introduced [98].

In order to find a soft margin for a linear kernel we have to introduce the hinge function: $\max(0, 1 - y_i(\vec{w} \cdot \vec{x_i} - b))$, of which y_i is the target and, $(\vec{w} \cdot \vec{x_i} - b)$ the value of the theoretical repertoire is the result. The function returns zero if the training point is on the correct side of the margin. This is inserted in the following function:

$$\left[\frac{1}{n}\sum_{i=1}^{n}\max(0, 1 - y_i(\vec{w} \cdot \vec{x_i} - b))\right] + \lambda \|\vec{w}\|^2 \tag{5.8}$$

Where the parameter λ is the trade-off parameters between increasing the margin and misclassified points.

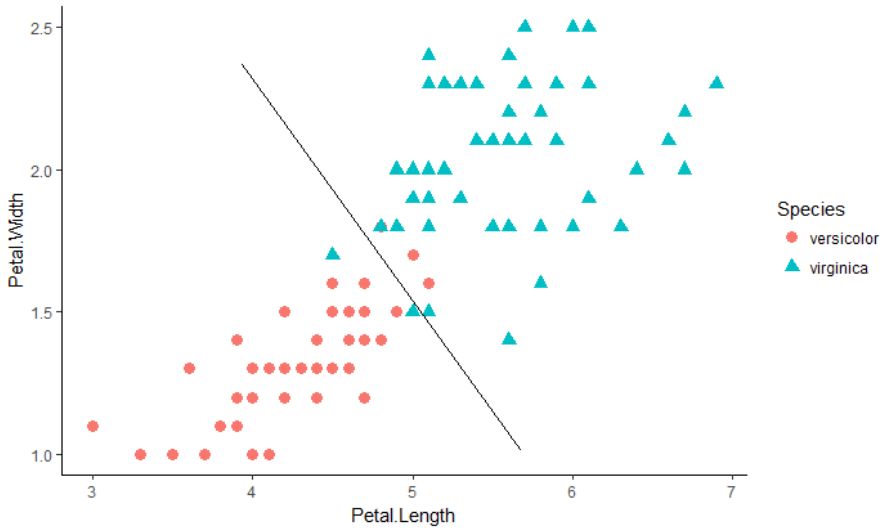


Figure 5.8: A hyperplane in a non-linear separable space: In this plot a possible linear hyperplane for this set of data is reported, using a soft margin approach. With the soft margin approach, misclassified points are weighed and computed to produce a hyperplane with the lowest value of misclassification.

5.3.3 Nonlinear classification: the Kernel trick

So far, our example has used a version of SVM that uses the dot product between the vectors to compute the maximum-margin hyperplanes, but in [98] a way to create nonlinear classifier by applying the concept of the kernel trick was introduced [99].

Equation 5.7 can be rewritten as:

$$\begin{gathered} \min \left(\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{n} \zeta_i \right) \\ subject\ to: \ y_i \left(w^t x + b \right) \geq 1 - \zeta_i \ with \ \zeta_i > 0 \end{gathered} \tag{5.9}$$

In which, the slack variable $\zeta_i = \max\left(0, 1 - y_i \left(\vec{w} \cdot \vec{x}_i - b\right)\right)$ for each $i \in \{1, \ldots, n\}$.

The variable ζ introduces a penalty for misclassified values. The optimal margins would be computed using a trade-off between the misclassifications and margin width. It defines how far the influence of a single training point reaches, in other words how many points far from the decision boundary are considered by the function. If it has a low value it means that an example has a far reach, close otherwise.

C is the regularisation parameter of the error term that controls the trade-off between maximizing the margin and minimizing the training error. Small C tends to ignore the outliers, while large C may tend to overfit the training data. These two parameters are not known a priori and one of the most common methods for tuning them is a combination of cross-validation and grid-search.

In the kernel trick each dot product is replaced by a nonlinear kernel function. This transforms the feature space, increasing the number of dimensions. One of the most common kernel functions is the Gaussian radial basis function:

$$k\left(\vec{x}_i, \vec{x}_j\right) = \exp\left(-\gamma \left\|\vec{x}_i - \vec{x}_j\right\|^2\right) \ for \ \gamma > 0 \tag{5.10}$$

In Figure 5.9, we can see that the first set of elements are points on a line (Figure 5.9 A) in which the two classes are clearly overlapping, so we increase the dimension to two (Figure 5.9 B). However, there is still too much overlap. By increasing once again to three dimensions, we can finally divide the classes (Figure

5.9 C).

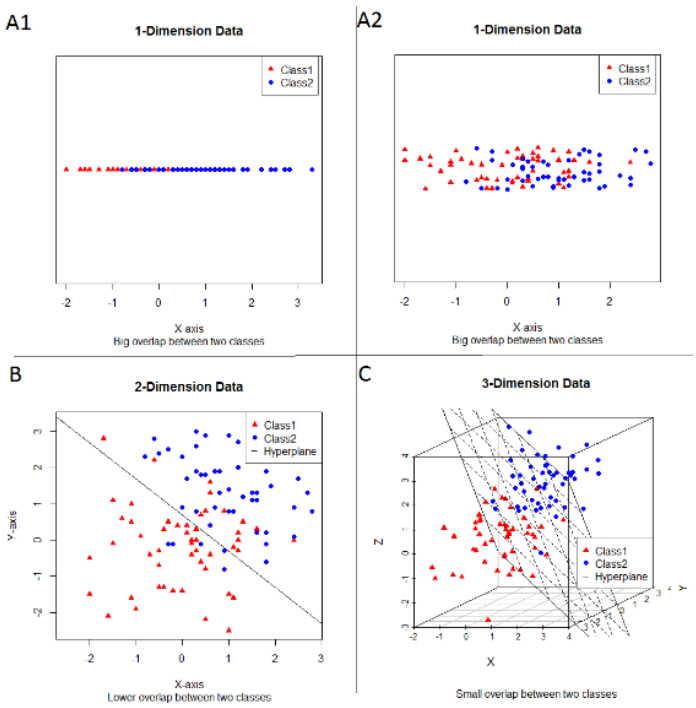


Figure 5.9: Increasing dimensions to find a separable space: In this plot I am giving a visual representation of how increasing the number of dimensions of a data set can help us to find a hyperplane that correctly predicts more and more data points. In **sub-figure A1** we have a set of one-dimensional data points. The two classes of points red and blue are distributed on a line (the hyperspace) in a way in which it is impossible to find a clear separation point, as we can see in **sub-figure A2** (the same data points of A1 but in a scatter form). However, if we increase by one dimension as in **sub-figure B**, and we draw a hyperplane, the number of mislabelled points decrease. This is even more the case if we increase by another dimension as in **sub-figure C**. Here there are no misclassified points. In conclusion, by increasing the number of dimensions, it is easier to find an optimal hyperplane that decreases the number of misclassified data points.

Using the kernel trick, it is possible to find a non-linear hyperplane without actually transforming the coordinates to a higher dimensional space.

The algorithm of the SVM is similar, but with the kernel trick every dot product between vectors is replaced with a nonlinear kernel function.

5.3.4 Multiclass SVM

As I have shown, SVM is inherently a two-class classifier. The best hyperplane is found between two classes and hence is optimal for binary classification problems.

Nonetheless, SVM can be applied to multiclass classification problems by reducing a multiclass problem to a series of binary-classification problems.

The most common strategies are the one-against-rest (OvR) and, the one-against-one (OvO). The latter is used in this thesis, see [100].

In the first strategy, during the training phase, each class's point is considered as one class, while all the other points from all other classes, are considered as a different class. In the testing phase, the samples are classified either as the single class or as the one formed by all other classes.

In the second strategy, for each class is applied a $K(K-1)/2$ binary classification for a K-way multiclass problem. Each binary classification receives samples from the training set of the pair of classes. During the testing phase, a voting system is applied, and the samples are classified as the class in which they are more often predicted into.

Chapter 6

Bag of Words

The bag of words (BOW) is a strategy usually adopted when we need to convert a document of strings, e.g., words in a book, into a vector of pure numbers that can be used for any sort of mathematical evaluation.

At first, this method was used in text [101] and image classification [102], but it has recently been introduced into bioinformatics, and can be successfully applied to CDR3 sequences' repertoires. For more general details of these approaches see [103][104][105].

With the BOW approach, we can redefine a highly complex document, such as a picture or a book, into a smaller set of low-level features, called codewords (also a codebook, dictionary or vocabulary).

The quality and origin of features is arbitrary, i.e. if we want to analyse a book or a series of books, we can choose as features all the words present inside the books, or the letters, or the combination of letters. As for the origin, the features can be all words present in the same books or all the words in the English dictionary, etc. As a result, the length of a codebook is defined by the number of features chosen.

6.1 Bag of words: Example

Item 1: Tom likes to go to the cinema on Sunday.

Item 2: Martin likes to go to the cinema on Tuesday.

Item 3: George prefers to visit science museums.

Item 4: Carl prefers to visit art museums.

First, we need to choose what type of feature we want to use. Because we are using four short sentences, we can use all the words present in the whole document.

Therefore, our code word is a list of all the words in the document without duplicates:

Codeword:

art, Carl, cinema, George, going, likes, Martin, museums, on, prefers, science, Sunday, the, to, Tom, too, Tuesday, visit

We can now represent each sentence as a vector, see Table 6.1:

Codeword	Item 1	Item 2	Item 3	Item 4
art	0	0	0	1
Carl	0	0	0	1
cinema	1	1	0	0
George	0	0	1	0
go	1	1	0	0
likes	1	1	0	0
Martin	0	1	0	0
museums	0	0	1	1
on	1	1	0	0
prefers	0	0	1	1
science	0	0	1	0
Sunday	1	0	0	0
the	1	1	0	0
to	2	2	1	1
Tom	1	0	0	0
Tuesday	0	1	0	0
visit	0	0	1	1

Table 6.1: Codeword for each sentence/item: In this table is represented each sentence as vector. In the first column there is the codeword with all single words present in the documents. In the following columns are listed the frequency of that word in each sentence. As result each sentence is now turned into a vector of numbers, in which any position of the vector represents a word and the numerical value its frequency in the sentence.

At this point, we can use these numerical vectors for any type of mathematical analysis.

For example, we can consider each vector as a point in an l-dimensional space, where l is the length of the codebook.

We can measure the distance between any points using the Euclidian distance [106]. In Table 6.2 we can see that items 1-2 and items 3-4 are closer to each other than the other combinations.

	Item 1	Item 2	Item 3
Item 2	2.0	-	-
Item 3	3.6	3.6	-
Item 4	3.6	3.6	2.0

Table 6.2: Euclidean distance between items: In this table is represented the Euclidian distance between each vector originated by Table 6.1. With this table we can see that vectors which originated from sentences with more words in common have a shorter distance.

6.2 Application to the CDR3 repertoires

The first problem we encounter when we apply the BOW model to the CDR3 repertoires is the choosing of the codeword features. If we use all elements of the CDR3 repertoires in the same way as we did for the example in the previous section, we would have an incredibly long codeword. Indeed, for a normal repertoire of around 1 million CDR3s, we can find more the 40,000 unique sequences.

This would lead to a codeword of an extremely high number of dimensions, posing several problems not just for the computational time required for its calculation, but also for the over-fitting issue that machine learning methods will incur.

The solution adopted here is to use all contiguous sub-strings of amino acids present in the repertoires. This approach is called k-mers [107][108].

6.2.1 k-mers

By the term k-mer is indicated all possible sub-strings of length k that are contained in a string. In our case we will refer to all possible sub-sequences of amino acids (of length k) present in the CDR3. Namely, single amino acids for $k = 1$, duplets for $k = 2$ and so on. k-mers are usually used for sequence analysis [109], motif identifications [110] and genome assembly algorithms [108][111].

The number of all possible k-mers present will form the codebook size that we will use. This size grows exponentially with the formula n^k, where n is the given

possibilities and k is the length of the sub-string, i.e.: A sub-string of two amino acids would have all the combinations of 20 amino acids, such as: AA, AC, AD etc., for a total of 400 possible combinations.

Once we have chosen the size of our string, we divide our sequences into all continuous subsets of that size. The number of k-mers for a given sequence of length L is: $L-k+1$. In other words, for a string size of 2 and a sequence of 10 amino acids we obtain 9 features, as from the first position of the sequence, we move our string one position at a time, until we reach the penultimate amino acid.

6.2.1.1 Example:

The codewords will all be triplet combinations of 20 amino acids, for a total of 8,000 features:

AAA (1), AAC (2), AAD (3), ... , YYV (7998), YYW (7999), YYY (8,000)

The sequences will then be parsed as in Figure 7.3:

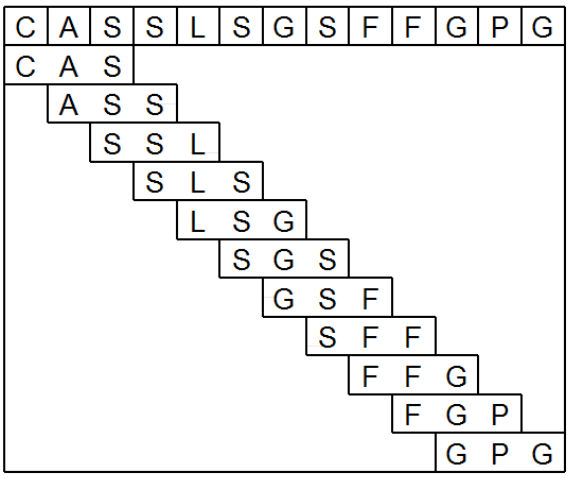


Figure 6.1: Continuous sub-strings of k-mer in a CDR3: In this plot I represented all the continuous sub-strings of k-mer of length 3, referred as triplets, present in a CDR3. On the top, is present an example of a CDR3 sequence: CASSLSGSF-FGPG, we proceed from the left-hand side of the sequence and we select the first sub-string of three amino acids (triplet), then, we slide one position to the right, selecting all sub string until the end of the sequence. All continuous sub string of triplets we select are listed below the sequence: CAS, ASS, etc. The total number of sub-strings is given by $L-k+1$, where L is the length of the string and k is the lenght of the sub-string, in this case the result is 11.

We repeat this process for all the sequences, then we just need to count the frequency of each triplet, the result being seen in Figure 6.2.

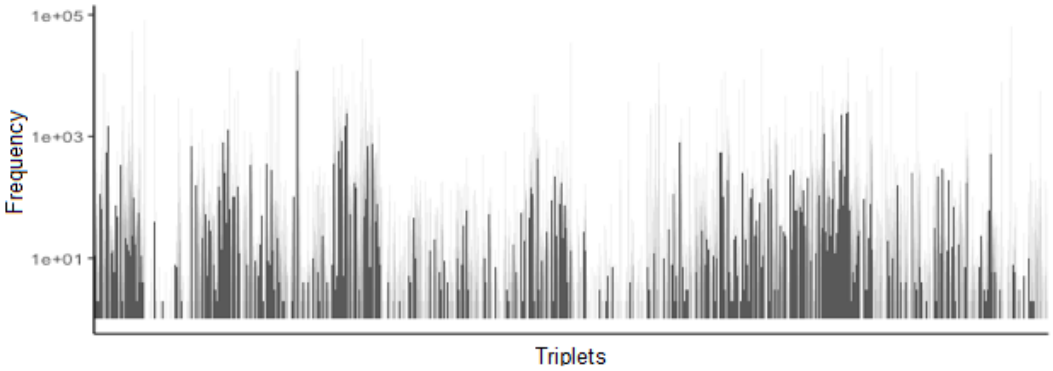


Figure 6.2: Count of all triplets in a CDR3 repertoire: In this plot I represent the CDR3 repertoire as a histogram with the frequency (y-axis) of each triplet (x-axis). We can see that some triplets are overwhelmingly overrepresented while others are almost if not totally absent.

Chapter 7

K-means

Using the bag of words and k-mers we are now able to convert our database of sequences into codewords formed by combinations of k-mers and their frequencies. However, for k-mers where k is greater than 2 the size of the codeword grows exponentially, and for triplets we already have 8,000 features. Such value has originally been considered too large in [16]. Clustering together the most similar triplets has been proposed as a way to reduce that number. It has been choosen to use the partitional clustering method of k-means.

The k-means algorithm performs better as compared to a hierarchical algorithm (HC) [112] and the execution takes less time, with a time complexity of $O(n^2)$ [113], lower than other HC methods that have a time complexity between $O(n^3)$ [114] and $O(n^2 \log n)$. On the other hand, HC provides good quality results in respect to k-means. In general, a k-means algorithm is good for a large dataset and HC is good for small datasets [113].

7.1 The algorithm

Given a set of points, where each point is an n-dimensional vector, the algorithm is able to separate the n points into k sets (with $K \leq n$), forming a number of clusters $S = \{S_1, S_2, \ldots, S_k\}$ with centers $(\mu_1, \ldots, \mu_k)$, the Within-Cluster Sum of Squares formula (WCSS) is defined as Equation 7.1.

$$WCSS = \min \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| x_j - \mu_i \right\|^2 \tag{7.1}$$

The algorithm is composed of four steps:

1. **Starting seeds:** allocate randomly a k number of starting points named "seeds". After the first iteration, these points are called centroids. See Figure 7.1.

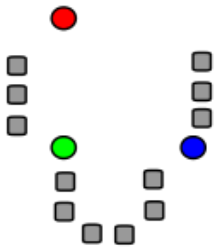


Figure 7.1: *K*-means algorithm, step 1: Three seeds are selected in a dataset space. Much freedom is given to the seed choosing; they can be randomly selected point in the space or following pre-determined formulas. Figure source [115]

2. **Assignment step:** Compute the distance between each of the points of the experiment and the seeds. Usually the Euclidean distance is used. The point considered closest to a seed is considered a part of that cluster. See Figure 7.2.

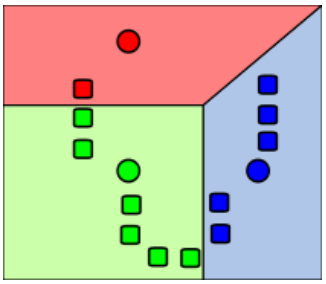


Figure 7.2: *K*-means algorithm, step 2: Once the seeds are chosen, the data points with the smallest distance to one seed are considered as part of a newly formed cluster. In the plot are formed three clusters, red, blue and green, covering all the space closest to each respective seed. Figure source [115]

3. **Update step:** Calculate centroids: inside the newly formed cluster, a new centroid is computed as the average points amongst all of the point forming the cluster. If the new centroids have the same position as before, proceed to the fourth step. If not, repeat the previous step. See Figure 7.3.

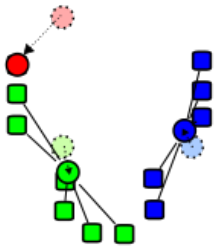


Figure 7.3: *K*-means algorithm, step 3: Now that a first cluster is formed we proceed by computing a new centroid for each cluster. If in the previous step the seeds where randomly chosen, now the centroids are computed as the middle point among all of each cluster. Figure source [115]

4. **Exit step:** When all centroids in the iteration i and in the previous iteration $i-1$ are equal, the clustering process is completed. This is also known as the convergence phase of k-means algorithm. See Figure 7.4

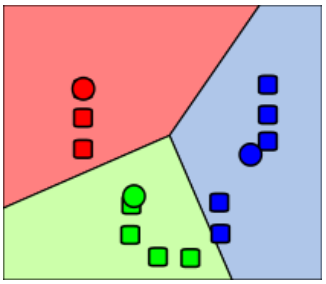


Figure 7.4: *K*-means algorithm, step 4: When the new centroids are computed the process restarts: all data points are reassigned to the closest centroid, a new centroid is computed and so on. The process exits when the position of all new centroids is equal to the previous ones, as this means that the process has reached a stable point and the data points will not be assigned to new clusters. Figure source [115]

7.2 Limitations

The k-means algorithm is an unsupervised algorithm, meaning that it does not need a training/test set. The algorithm is very sensitive to outliers and their presence can skew the cluster formation or in extreme cases create clusters of single elements.

It is important to comment on the choice of the initial seeds: seeds are usually chosen randomly and different runs of k-means on the same database could produce different cluster results.

We can choose how many clusters we want in our output, but we cannot be sure

if that is correct. Many methods have been proposed to compute the right number of clusters, such as the elbow method or some rule of thumb such as: $k \approx \sqrt{n/2}$, where n is number of points and k the optimal number of clusters. However, no method is definitive.

An example is the set of data "iris" used in the previous chapter. We have three sets and 150 points. If we apply the rule of thumb $k \approx \sqrt{n/2}$, the result is $\sqrt{150/2} \approx 8$. This result proves that also this method is inadequate to compute the correct number of clusters.

In Figure 7.5. I applied the k-means algorithm, with three clusters, and also with the correct number of cluster the final result has some minor errors.

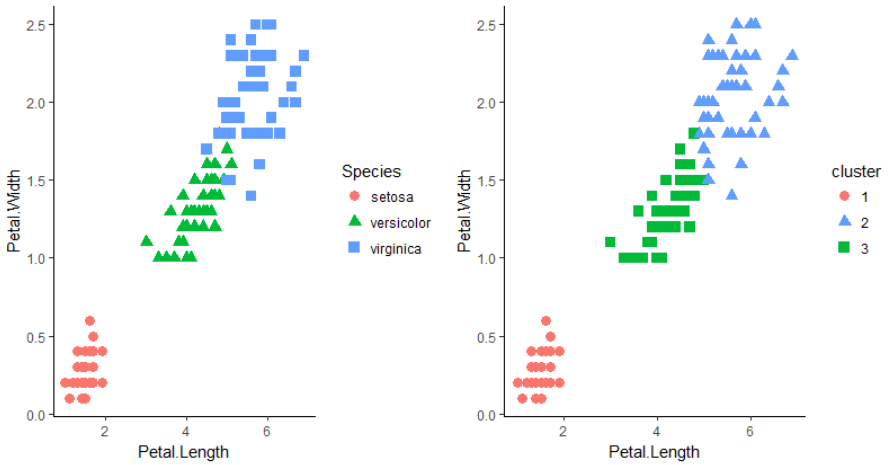


Figure 7.5: Real case application of k-means algorithm: In this plot I apply the k-means algorithm to the dataset used in the previous examples. On the left-hand side we have the iris dataset coloured according to their real class and on the right-hand side the data classified according to the k-means. The algorithm correctly divides group 3 from the rest but has produced some misclassification for groups 1 and 2.

Chapter 8

The SVM Experiments

In this section, we are now going to examine the results for the application of the SVM to the CDR3 repertoire classification problem. The section is composed of my analysis and redoing of [16] and the application of SVM method to different repertoires and the required new actions.

1 **Bag of Word (BOW)**

To get the n-dimensional point required for an SVM classification we first need to 'convert' our CDR3 sequence from strings to numerical vectors. We have already seen how to apply the BOW and k-mer to the CDR3 sequences in section 6.2, and that it is a clever and effective way to reduce the complexity of the repertoire, converting a huge set of strings to a simple vector of numbers by reducing the "word" of the BOW to single amino acids, duplets, triplets and so on.

Here, I have used duplets and triplets. These combinations of amino acids are considered an optimal compromise between the use of single amino acids that might lead to a loss of information, and the use of longer p-tuples, that would be very computationally expensive and hard to compute.

2 ***K*-means algorithm**

As we have already seen, the k-means algorithm works with a point in space, and we have only strings of letters (AA, AC; AAA, AAC, etc.). Somehow, we need to convert these strings into numbers in order to have some suitable measure of distance between points.

Thus, any amino acid must be replaced by numbers. In [16] the Atchley numerical factors are used. Let us see what they are.

3 **Numerical factors**

For each amino acid in the available literature there are tables of values listing the different features of the element, such as polarity, hydrophobicity, etc [116].

Parallel to the studies of the single amino acids features, other authors have focused their attention on what is called the "sequence metric problem" [117]. In this, a great number of chemical/physical amino acid properties are gathered together, and each amino acid is given a synthetic and representative numerical value. Converting the alphabetic letters into an array of numerical values allows us to summarise all the underlying properties of each amino acid.

In my analysis, however, I have extended the work to three other numerical factors, including two from the literature: the Sandberg [118] and Kidera [119], and one to use as a control and without biological meaning. More details for each numerical factor are reported in Table 8.1.

Factors	Len	Features
Atchley	5	In [117] the authors used multivariate statistical analyses to reduce 494 amino acid attributes into a smaller set of highly interpretable numeric patterns. These are summarised by five multidimensional patterns of attributes that reflect polarity, secondary structure, molecular volume, codnaon diversity and the electrostatic charge of each amino acid. Using an alphabetic letter code to represent the amino acid would result in a loss of information on the physiochemical properties of amino acids. For example, the amino acid leucine (L) is more similar to valine (V) than leucine is to alanine (A). However, the alphabetic "distance" between these letters in the alphabet does not reflect these relationships. Therefore, the use of the small set of numerical features that they propose, can help to understand relations and similarities between different amino acids.
Sandberg	5	In [118] the authors analyse 26 different amino acids, properties from 87 amino acids, and using the principal component analysis they have reduced these properties into 5 purely numerical factors.

Kidera	10	In [119] the author worked on 188 physical and conformational properties of the 20-natural occurring amino acids in order to reduce this number to a smaller value (10) without losing the information contained in their physical properties. In the paper they first use hierarchical cluster analysis to cluster similar amino acid properties together before using partial correlation factor analysis to produce a single numerical value for each cluster. The result is 10 values, one for each of the following properties: 1. Alpha-helix (bend-structure preference-related). 2. Bulk-related. 3. Beta-structure preference-related. 4. Hydrophobicity-related. 5. Normalized frequency of double bend. 6. Average value of average composition. 7. Average relative fractional occurrence. 8. Normalized frequency of alpha-region. 9. pK-C 10. Surrounding hydrophobicity in β-structure.

Arbitrary	20	Last, I used numerical factors without any relation to any amino acid properties. I choose to use a vector of 20 numbers, with a single positive number corresponding to the alphabetic order of the amino acid name. Ala: 1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; Cys: 0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0; etc. This way to represent amino acids is to be preferred to a simpler system, like giving each amino acid an increasing number of 1,2,3 etc. This because the cardinal number may underlie a 'distance' between amino acids. This is the same problem we face when using an alphabetic letter code.

Table 8.1: List of Numerical Factors: In this table is found the numerical factors used to convert the string of amino acid letters into numerical values. Atchley, Sandberg and Kidera factors, all are present in the literature, and are all derived from measurements of the amino acids physical/chemical properties. In contrast, the arbitrary factor is a pure invention and has no biological background. In the first column of the table is the factor name, in the second how many values compose the numerical factor, and last is a list of what those value represent and how they have been generated.

Once the single, duplets and triplets are gathered into the resulting k-means clusters, the value for each cluster will be given by the sum of the values of the original duplets/triplets now forming the new cluster. We can now pass to the last step.

4 **SVM classification**

In order to measure the outcome of the classification, I compute for each experiment the overall success rate (OSR). As described in [120], the OSR is defined by the trace of the confusion matrix divided by the total number of classified instances:

$$OSR = \frac{1}{n} \sum_{i,j=1}^{x,y} n_{i,j} \tag{8.1}$$

Where n is the number of classified instances, x the number of elements within

the class and y the number of the class.

The greatest advantage of the OSR is that it is multiclass, symmetrical, ranges from 0 (perfect misclassification) to 1 (perfect classification), and the result corresponds to the observed proportion of correctly classified instances.

In this thesis, I choose to rappresent the OSR as a value between 0 and 100 and refer to results as percentage of the overall success rate. See Table 8.2.

Example:

For a three class experiment, we test six elements:

Tests	Blue	Green	Red
Blue 1	100		
Blue 2	80	10	10
Green 1		100	
Green 2			100
Red 1			100
Red 2			100

Table 8.2: Overall success rate —Example: Reported is an example calculation of the overall success rate used in this thesis. We have a three-class system: Blue, Green and Red classes and six test subjects, formed of 100 samples each. We can see that Blue 2 has 20 samples misclassified as Green or Red. Green 2 is completely misclassified. The OSR would therefore compute as: Class Blue outcome $100 + 80$, plus class Green outcome 100, plus class Red outcome 100+100, divided by the total number*n*of classified instances, 600, results in: $480/600 = 0.8$. In other words, 80% of samples are correctly classified.

8.1 Experiments and Results

The experiment with the SVM has been designed in the following way:

From each of the murine CDR3 repertoire, I randomly selected 10,000 sequences, then counted the number of times each 2-mers (duplets) and 3-mers (triplets) is present in all the sequences, producing a vector of frequencies for the 400 duplets or 8,000 triplets (bag of words).

This process has been repeated 100 times for each repertoire.

I converted each duplet and triplet into the four numerical factor described is Table 8.1, and used the k-means algorithm to reduce the 400 duplets and 8,000 triplets into 100 features.

Therefore, if before we had 37 mice repertoires with different number of sequences, we now have for each repertoire two sets of 100 vectors of 100 features, one set originated by the use of 2-mers, a.k.a. duplets, and another one by 3-mers, a.k.a. triplets.

As seen in *k*-means algorithm Chapter 7, the *k*-means algorithm would exit the algorithm after reaching the convergence phase. However, the number of iterations needed for our dataset is too high, and the function `kmeans` of the `R` package `stats` ends after 10^9 iterations.

The immediate consequence is that, because the initial cluster seeds are chosen randomly, every time we run the program we will obtain different results. In other words, the final clusters will be composed of different *p*-tuples every time.

In the original paper, this aspect had not been noticed. However, as I experienced by running numerous SVM experiments repeating the *k*-means step, the final SVM OSR does not actually change.

Despite the *k*-means results having no effect on the SVM classification results, I consider this to largely undermine the use of a *k*-means algorithm to produce clusters with a minimum biological significance.

After noticing this first issue I decided to run several more tests altering all main parameters. I can summarise my tests into four experiments:

1. Run the experiment similarly to the one performed in [16].

2. Test the previous experiment applying different numerical factors.

3. Classifying by data point and antigens and, extending the experiment to the new set of data.

4. Test different values for the number of *k*-means clusters.

8.1.0.1 Exp.1: SVM classification of repertoire group A

In [16], as described above, 100 numerical vectors of 100 features-long has been originated by 100 random sub-samplings of a total of 10,000 sequences from each

repertoire. With those the following experiment has been performed: a leave-one-out cross validation using Support Vector Machine classification, in which one repertoire is the test and the others are the training set.

In [16] only the repertoires from the first experiment has been used (Group A, see Section 4.2), divided into four classes of elements: six control mice, six mice sacrificed at day 5, day 7 and day 60.

The results that I obtained are in Table 8.3:

100 clusters per 10k sequences	**Duplets**	**Triplets**
Atchley	83.33	62

Table 8.3: Results of experiment present in [16]: For this experiment 10,000 sequences from each murine repertoire has been converted into 400 duplets and 8,000 triplets and gathered into two 100 features-long vectors. These two vectors have been, previously, created by converting the sub-string of amino acid into numerical values using Atchley factors and gathered into clusters using the k-means algorithm. With this data has been performed: a leave-one-out cross validation using Support Vector Machine classification method. Each repertoire is considered correctly classified when the major part of 100 vectors of triplets are correctly classified, using the linear SVM classification function. We can see that the result for duplets is much higher than for triplets, however, still higher than the random classification (25%).

The result for this linear classification for duplets 83% and triplets 62% is higher than the random classification 25%.

When I extend this analysis using the other numerical factors, I obtain similar values, as reported in Table 8.4:

100 clusters per 10k sequences	**Duplets**	**Triplets**
Sandberg	70.83	66.66
Kidera	75	66.66
Arbitrary	70.83	62.5

Table 8.4: Results for SVM classification using linear function and originating the data point from different numerical factors: For this experiment I use the same methodology used in Table 8.3, but using Sandeberg, Kidera and an arbitrary numerical factor instead of the Atchley. From the table we can see that changing the numerical factors does not produce a difference in terms of classification result, and the results are similar with what obtained in table Table 8.3.

We can immediately see that the results are not very different, as using numerical factors with or without biological significance does not lead to significant changes.

This new information suggests that the use of different numerical factors, being they derived from physical-chemical analysis or completely arbitrary, is not contributing to the overall success rate for both duplets and triplets.

8.2 SVM classification of repertoire group A and B

In this new test, I performed the same experiment as before, but added the new repertoires coming from the second experiment, Group B.

Because Group B repertoires have only three-time points (Control, Day 7 and Day 60) and those do not match those in Group A, I chose to reduce the classes to three: Controls (Control mice from A and B), Early (Day 5, Day 7 and Day 14) and Late (Day 60 from A and B).

The results for the experiments with the new dataset are present in Table 8.5:

100 clusters per 10k sequences	**Duplets**	**Triplets**
Atchley	83.78	67.56
Sandberg	81.08	75.67
Kidera	83.78	72.97
Arbitrary	86.48	75.67

Table 8.5: Experiment with group A and B mice divided in three classes: Here I performed an SVM classification using a linear function and as in Table 8.4 here we see that the results do not seem to be influenced by the numerical factors.

As we can see, the results are higher than the random classification (33%) and in general higher than the results for Table 8.4. This is because in the previous experiment there were four classes and Day 5 and 14 were often misclassified one for the other, while now these two time-points are in the same class.

For this experiment as for the one above different types of numerical factors do not influence the outcome of the experiment.

8.3 OVA vs. CFA mice

So far, we have tried to classify the repertoires on the basis of time after immunisation. However, the mice used in the experiments have also been immunised with two different antigens, CFA and OVA. Accordingly, I also tried to classify the repertoires on the basis of the antigens.

To do so I performed two experiments: OVA vs. CFA with Group A, and OVA vs. CFA with Group A and B. This was replicated for duplets and triplets.

When Group A was used, we have 18 repertoires, 9 OVAs and 9 CFAs, while for Group A and B there are 28 mice, 15 OVA and 13 CFA.

The results are presented in Table 8.6:

100 clusters per 10k sequences	**Duplets Group A**	**Duplets Group A & B**	**Triplets Group A**	**Triplets Group A** & B
Atchley	61.11	53.57	55.55	57.14
Sandberg	72.22	55.55	55.55	57.14
Kidera	61.11	50	50	60.71
Arbitrary	66.66	61.11	61.11	39.28

Table 8.6: OVA vs. CFA mice classification per duplets and triplets: In this table are reported the results of the classification for OVA vs. CFA mice repertoire from Group A and Group A+B, repeated for duplets and triplets and for four different numerical factors. We can see that classification results do not vary on the basis of numerical factors and p-tuples used.

From Table 8.6 we can see that there is not variation in terms of duplets vs. triplets or using different numerical factors, plus the results in terms of OSR are not much higher that a random classification (50%). This implies that the SVM classification method as previously used is not a valid method for the classification of OVA vs. CFA mice.

8.4 Different number of clusters k-means

So far, all experiments have been performed using 100 clusters for the k-means. However, choosing 100 clusters was an arbitrary choice.

Hence, in the following tables (Table 8.7 and Table 8.8) I have re-run the very first experiment of Group A mice with four classes with an increasing number of

clusters, until using the entire number of features, 400 and 8,000, and thus no clustering at all.

Naturally, for the experiments without clustering, I avoided the use of numerical factors and *k*-means algorithm, and used the values directly coming from the BOW.

In Table 8.7, the results for duplets, classified with 15, 23, 100, 200 and 400, can be seen.

Duplets	**15**	**25**	**100**	**200**	**400**
Atchley	62.5	83.33	83.33	70.83	79.16
Sandberg	66.66	79.16	70.83	70.83	
Kidera	66.66	79.16	70.83	70.83	
Arbitrary	70.83	79.16	70.83	66.66	

Table 8.7: Different number of clusters for *k*-means for duplets: For this experiment I used an increasing number of *k*-means cluster result values, with the aim of finding whether the difference in such numbers will improve the final OSR. As we can see from the table, changing the number of clusters used do not lead to advantages.

As we can see, the results do not show any consistent relationship between number of cluster and OSR. Rather, if I use all possible 400 amino acid duplets, I can completely avoid using numerical factors and the *k*-means algorithm without losing OSR.

In Table 8.8, the results for triplets, classified with 25, 64, 100, 200, 500 and 8,000, are reported.

Triplets	**25**	**64**	**100**	**200**	**500**	**8,000**
Atchley	66.6	62.5	62.5	70.83	62.5	70.83
Sandberg	70.83	58.33	66.66	62.5	58.33	
Kidera	70.83	62.5	62.5	70.83	62.5	
Arbitrary	70.83	58.33	62.5	75	75	

Table 8.8: Different number of clusters for *k*-means for triplets: Here I repeated what I tried in Table 8.7 but with triplets. As ever for triplets there is no difference in the use of different numbers of clusters.

As with the results for duplets, the results for triplets do not show any consistent difference between number of clusters and OSR.

8.5 Test without numerical factors and *k*-means

This test is performed using all duplets and triplets, avoiding using the numerical factors and the *k*-means algorithm entirely.

I can perform two types of tests per duplet and triplet: classification by sacrificed date, a four class SVM classification of the Group A, a three class SVM with Group A and B mice repertoires; and, classification by antigens, Group A, OVA vs. CFA repertoires and Group A and B, OVA vs. CFA repertoires. The results for duplets and triplets are presented in Table 8.9 and Table 8.10.

Duplets	**Group A 4 classes (data points)**	**Group A & B 3 classes (data points)**	**Group A OVA vs. CFA (Antigens)**	**Group A & B OVA vs. CFA (Antigens)**
	79.16	81	66.66	60.71

Table 8.9: SVM tests using 400 duplets: In this experiment I repeated the test seen in the previous section but using the total number of duplets skipping the *k*-means/numerical factor steps. As we can see the results are as good as before.

Triplets	**Group A 4 classes (data points)**	**Group A & B 3 classes (data points)**	**Group A OVA vs. CFA (Antigens)**	**Group A & B OVA vs. CFA (Antigens)**
	70.83	86.48	61.11	64.28

Table 8.10: SVM tests using 8,000 triplets: In this experiment, as for Table 8.9 I repeated the test seen but using triplets. Also here the results are as good as before.

In Figure 8.1, there is a representation of whether the mice repertoires are classified correctly or not. In blue are the correctly classified and in white are the misclassified repertoires. The outcome of all four SVM tests previously discussed and the outcome for duplets and triplets is reported.

In Figure 8.1, Group A four classes, we can see that Day 5 and Day 14 and Day 60 all have some degree of misclassification, but if for Day 5/14 the repertoires need some time to differentiate properly and change while fighting the infection, the reason why D60 also has some misclassification does not appear clear. The control mice are, instead, correctly classified. In our experience, the classification of unimmunised controls vs. immunised repertoires is usually easy, and the results correct.

However, this is not the case for our second experiment, Group A and B, three classes. Here, two control mice from the Group B are misclassified. This specific group of control repertoires are often misclassified in almost all my classification experiments, not only in this SVM test. This fact, along with the fact that also Day 7 and Day 60 from Group B are a source of misclassification, in duplets and triplets, led me to consider that there is some sort of batch effect between the two groups of mice repertoires that is affecting my tests. However, we have seen above that reducing the number of classes, merging D5/D7/D14 to one class, leads to less misclassification and a higher classification performance.

The second set of experiments is OVA vs. CFA repertoires. Here, the OSR is lower and the number of misclassified mice higher. Furthermore, the occurrence of misclassification is spread in all classes of mice (Day 5, 7 etc.).

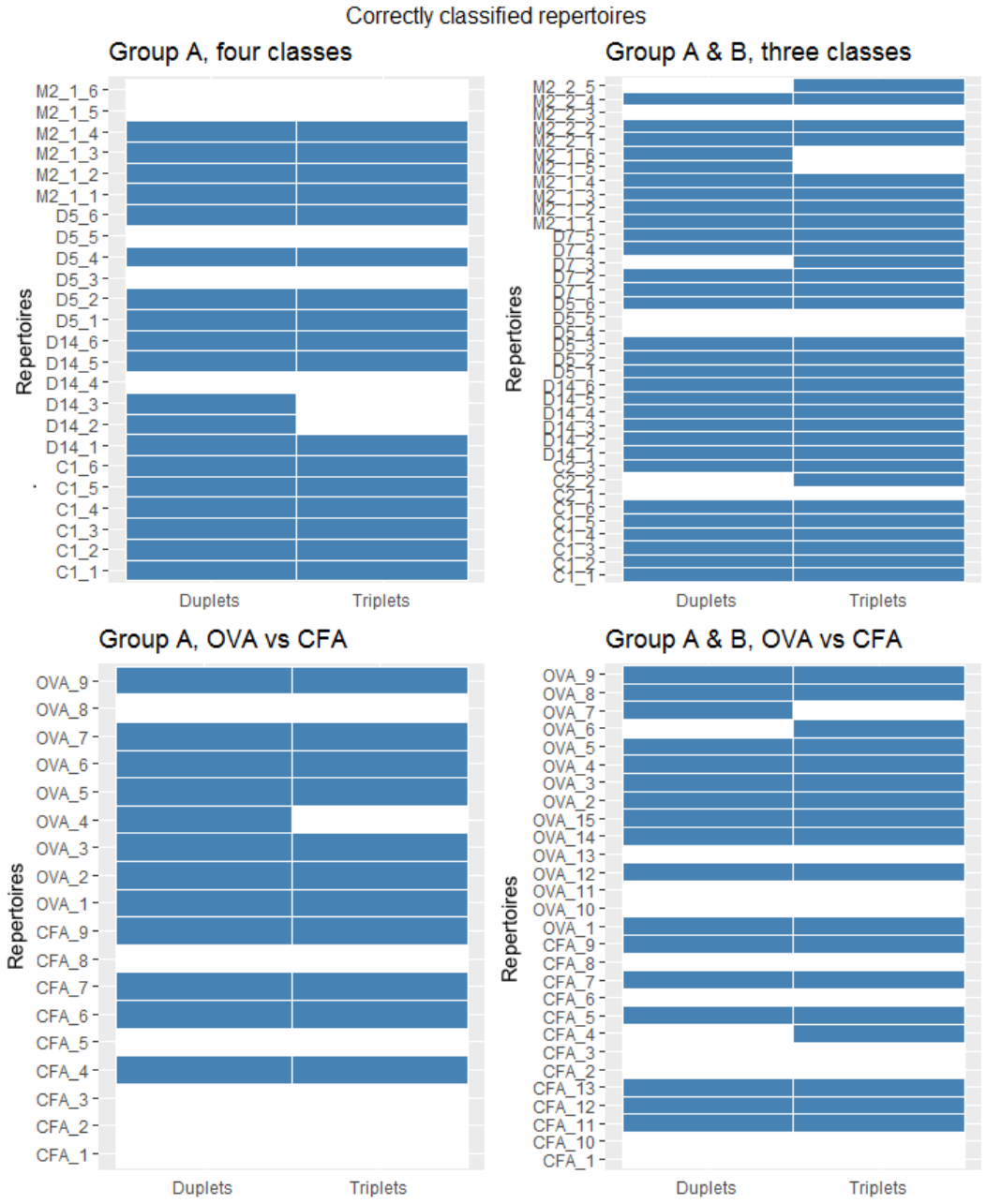


Figure 8.1: Results of mice classification: In this plot I show which ones are the particular mice repertoires that are correctly classified (in blue) and which ones are not (white) for duplets and triplets and for all four kinds of experiments. On the top is reported the classification by immunisation date, on the left-hand side only the group A mice are divided in four classes (Control, D5, D14, M2) on the right, group A and B are divided in three classes (Controls, Early and Late). On the bottom are the OVA vs. CFA classifications, on the left-hand side only the group A mice (mice groups: D5, D14, M2) on the right, group A and B (mice groups: D5, D7, D14, M2_1, M2_2). It is also interesting to see that the results between duplets and triplets are often the same, with one or two variations per experiment.

8.6 Discussion

8.6.1 *K*-means is not a valid clustering method

The *k*-means algorithm is a popular algorithm for unsupervised classification. However, given the high number of features and the limitations of the package's function, to perform a classification of the amino acid combinations (duplets and triplets) would be challenging and computationally expensive. We might still want to use it, but this would not bring us any considerable advantages.

Finally, it is very important to know how many clusters to use before performing the *k*-means algorithm. Without this information, the only option is to test as large a range as possible, and best find, depending on the situation, a method not dissimilar to that employed in Table 8.8.

8.6.2 Numerical factors do not affect the SVM OSR

Using a numerical factor to convert the strings of letters into numbers is, no doubt, a clever idea, and it is the obligatory path for whoever would want to research the biological property of *p*-tuples. We have seen in the experiments in Table 8.4, Table 8.5 and Table 8.8 that the use of one numerical factor with respect to another does not lead to a great variation in the results. Furthermore, the use of numerical factors with or without the biological background has no relevance whatsoever in this analysis.

These considerations, and those done with the *k*-means, lead me to conclude that the use of numerical factors and *k*-means are two steps of the method that can be avoided. Indeed, neither would accomplish the task they were intended for. They do not reduce the computational work and reducing the feature space does not improve classification OSR.

Bayes' theorem and its application in SVM feature selection

Chapter 9

Bayes' theorem

9.1 Introduction

After my work with the SVM classification, I started considering that the use of a numerical vectors originated from all duplets and triplets might not be necessary. Perhaps a better approach would be using a smaller subset of features: this could in turn increase the overall success rate.

By doing SVM classification runs with a random number of random subsets of triplets, I sometimes got an OSR much higher than ever before (See Figure 9.1), therefore it was clear that a higher success rate was possible. The question was, how to select a non-arbitrary subset of features that would increase my success rate, and possibly be considered having biological relevance.

My idea was to try to classify one feature at a time, rate and sort them from the highest to the lowest rate and use a progressively larger subset of features until I had the best CDR3 success rate. I reasoned that if a single feature could be correctly classified, perhaps a small collection of features with a high classification rate would lead to greater success rate in the overall classification of the CDR3 repertoires.

My idea was to use one feature SVM; however, after some research, I found the formula of the 1-Dimensional Bayesian Function (1-DBF). This function can evaluate the likelihood of an element to be part of one of two classes, using mean and standard deviation of the two samples.

The work presented here was published in [17].

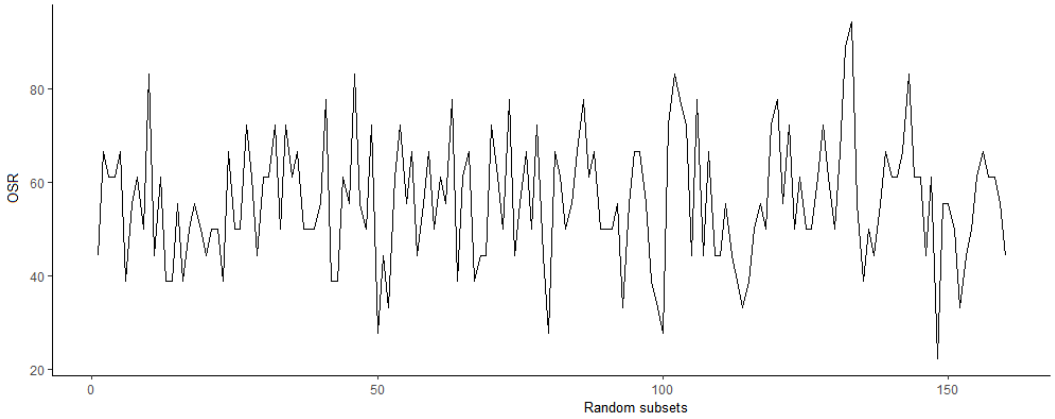


Figure 9.1: Random subset of 50 triplets: I tested if I could obtain high level of classification using smaller set of triplets. In this plot I report 160 tests using 50 triplets, rather than the normal 8,000. I randomly divided the 8,000 triplets into 160 sets and performed the OVA vs. CFA SVM classification with Group A mice. The results are interesting: we can clearly see a great fluctuation of results from a minimum of 22% to a maximum of 94% of OSR. This proves, at least in theory, that a smaller subset of features can produce higher classification results with respect to the whole set. This result made me consider the idea to use such smaller set and moreover stat researching a model to select a meaningful and not random subset of triplets.

Before focusing on the 1-DBF, let us briefly examine Bayes' theorem.

9.2 Bayes'Theorem

9.2.1 Overview

Bayes' theorem is today considered one of the main theorems in statistics, and one of the most applied formulae in science. Rather than being immediately celebrated by the statisticians at the time of its publication, its importance grew steadily until the middle of the last century. It is now considered essential in all statistics courses, and applied in almost every field of research —not least in Bioinformatics, where it has been applied extensively to the biological system analysis [121].

At first glance, Bayes' theorem can seem confusing, counterintuitive, and hard to grasp. We know that, for many, statistics is not intuitive, as with other aspects of mathematics [122]. However, if we analyse the thought processes leading Bayes to his theorem, we see that these are natural and logical ways of thinking.

9.2.2 History of Bayes' theorem

Bayes theory is named after the Reverend Thomas Bayes. Around 1740 Bayes performed a thought experiment: he imagined putting his back against a flat, square table, and launching a ball onto the table without knowing where it would land. Then, he thought of launching a second ball, this time asking his assistant if the ball landed to the left, to the right, in front of, or behind the first ball. Using this system, he was able to narrow the position of the first ball with each new launch. It is not possible to know exactly where the first ball landed using this system; but it created a method, or a way of thinking, whereby each new piece of evidence improved the estimate.

Bayes never published his idea. After his death in 1761, his friend Richard Price found his notes, re-edited them, extended them (at some points), and eventually published them [123]. He contributed greatly to Bayes' theorem —by modern standards, we would normally refer to the theorem as the Bayes-Price theorem, giving the appropriate credit to Price. Despite Price's work and the publication, the theorem remained unknown until it was rediscovered, reinterpreted, and brought to modern formulation in 1774 by Pierre-Simon Laplace in [124]. Here, it was stated that "The probability of a cause (given an event) is proportional to the probability of the event (given its cause)". In more modern words: Bayes' theorem describes the likelihood of an event, based on prior knowledge of conditions that might be related to the event.

9.2.3 Classical Representation and Examples

The most classical representation of the formula is the following:

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \tag{9.1}$$

Where A and B are two events that we want to analyse.

$P(A)$ and $P(B)$ are the distinct and independent probabilities of the event A and event B (prior probabilities).

$P(A|B)$ (read: P of A such that B) is the conditional probability, the probability

of the event A, given that the event B has occurred (posterior probability).

Vice versa, $P(B|A)$ is the probability of B given A.

Therefore, this formula relates the prior probabilities to the posterior probabilities, with the possibility to integrate new observations to an established model based on previous observations.

Let us now move on from this formal terminology and look at some real examples.

9.2.3.1 Example 1:

In this example, let us consider a school composed of 60% boys and 40% girls, in which all boys have a short haircut while the percentages of girls with long hair and girls with short hair are equal.

We meet a student with a short haircut. What is the probability that the student is a girl?

We can use the Bayes formula to answer this question. In order to do so, let us identify the elements of the formula.

In this case, four elements are involved: the gender of the students, boys (B) and girls (G); and the hairstyle, long (L) or short (S). The hairstyle events are connected and dependent on the gender of the student and those are therefore our dependent variables. The gender of the students is the independent variable.

From the composition of the school, we can compute that the independent probability the student is a girl (event G) is 40% $P(G)$. Conversely the independent probability the student is a boy (event B) is 60% $P(B)$.

The probability that a student has a short haircut (event S) considering the entire school $P(S)$ is: all males plus half of females $60\% + (40\%/2) = 80\%$ of all students.

The probability that a student has a long haircut (event L) is: $P(L) : 1 - P(S) = 20\%$.

The dependent probability of a student with short hair, given that the student is a girl $P(S \mid G)$, is 50%. This is because the girls are equally divided between long and short haircuts.

The dependent probability of a student with short hair, given that the student is a boy $P(S \mid B)$, is 100%, given that all boys have short hair.

Having established this, we can now apply the formula to compute the conditional probability:

$$\begin{aligned} P(G|S) &= \frac{P(G)P(S|G)}{p(S)} = \frac{0.5\times0.4}{0.8} = 0.25 \rightarrow 25\% \\ P(B|S) &= \frac{P(B)P(B|G)}{p(S)} = \frac{0.6\times1}{0.8} = 0.70 \rightarrow 75\% \end{aligned} \tag{9.2}$$

In conclusion, the probability that the student with short hair is a girl is 25%.

Interestingly, we could have computed $P(G \mid S)$ even without knowing $P(S)$ by doing:

$$\begin{aligned} P(G|S) &= \frac{P(G)P(S|G)}{P(G)P(S|G)+P(-G)P(S|-G)} = \\ \\ \frac{0.4\times0.5}{0.4\times0.5+0.6\times1} &= \frac{0.2}{0.8} = 0.25 \rightarrow 25\% \end{aligned} \tag{9.3}$$

9.2.3.2 Example 2:

Let us assume that a test for a disease is positive. We know that the test performed has an accuracy of 99%, and that the disease it tests for affects one person out of a thousand. What is the likelihood of having the disease?

One might say 99% because of the accuracy of the test. This is not correct, because the accuracy of the test actually means the percentage of testing positive (event T), given that you have the disease (event D) is $P(T \mid D)$. What we really want to know is the probability of having the disease, given that the result of the test is positive: $P(D \mid T)$.

If the $P(D)$ is equal at the percentage of the population 0.001%, we can now apply Bayes' theorem, so that we have:

$$\begin{aligned} P(D \mid T) &= \frac{P(T|D)P(D)}{P(T)} = \frac{P(T|D)P(D)}{P(T|D)P(D)+P(T|-D)P(-D)} = \\ \\ \frac{0.99\times0.001}{0.99\times0.001+0.01\times0.999} &= 0.09 \rightarrow 9\% \end{aligned} \tag{9.4}$$

9% is less scary than 99%. But why is this value so low?

We know that the disease affects only one person in one thousand and that the test has a 99% accuracy, therefore in a sample of one thousand, only one person would be a true positive, but ten people would results as false positive. See Figure 9.2:

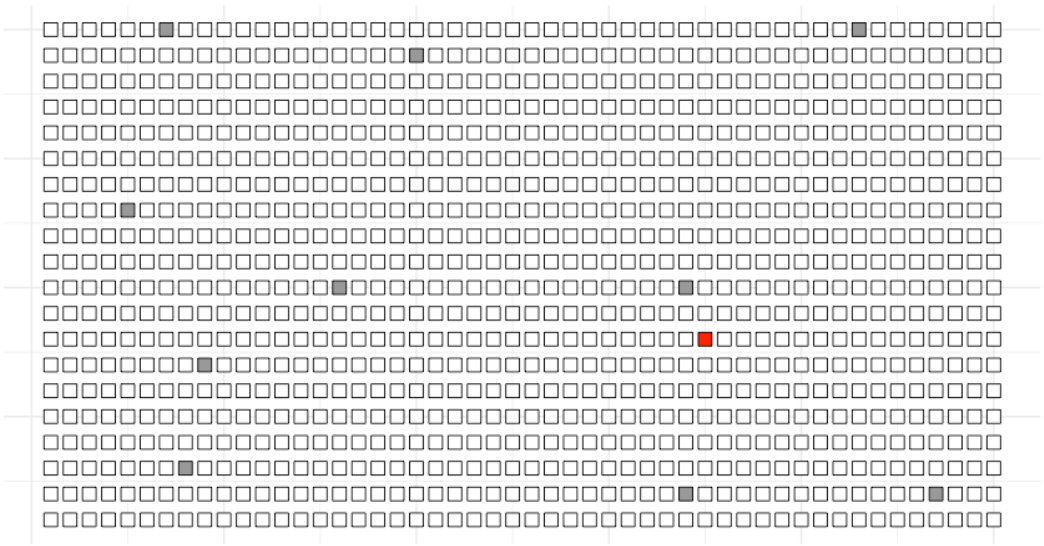


Figure 9.2: Positive and false positive results of the test: In this picture, we see a sample of 1,000 people tested with a test which has 99% accuracy. A priori, we know that only one individual in 1,000 is positive for the test (in red), and all the other results are negative. However, because the test is only 99% accurate, we are going to have 10 individuals who will test positive, even though they do not actually have the disease, thus false positive (in grey). Therefore, according to the Bayesian theory, the probability of a true positive result for a 99% accuracy test for an incidence of 1 over 1,000 is not 99%, as one could think, but only 1 over 11 (all the person considered positive, true and false combined) thus, 9%.

Therefore, an individual resulting positive has a 9% possibility to be a true positive.

$$1/11 \ = 0.09 \rightarrow 9\% \tag{9.5}$$

Now, if we repeat the test and the result is still positive, how much is $P(D|T)$? Again, we can use Bayes, using the previous result as our new independent variable $P(D)$, therefore:

$$P(D \mid T) = \frac{P(T|D)P(D)}{P(T)} = \frac{P(T|D)P(D)}{P(T|D)P(D)+P(T|-D)P(-D)} = \tag{9.6}$$

$$\frac{0.99\times 0.9}{0.99\times 0.9+0.01\times 0.91} = 0.9073 \rightarrow 90.73\%$$

A third time would be 99.89% and so on.

Chapter 10

Bayes' theorem as classification method

10.1 Overview

The 1-Dimensional Bayesian Function (1-DBF) formula has been reported and described in [97]. Here, I will discuss the formulation of the formula and analyse its output in two examples.

Given two classes C_1, C_2, different and dividable by a hyperplane (see SVM for definition). The posterior probability that a new observed element x belongs to a class k, is given by:

$$P(C_k \mid x) = \frac{P(C_k)p(x \mid C_k)}{p(x)} \tag{10.1}$$

Where $P(C_k)$ is the a priori probability of class k. $p(x \mid C_k)$ is the probability density function when x belonging to class k, and $p(x)$ is a probability density function given by:

$$p(x) = \sum_{k=1}^{2} P(C_k)p(x \mid C_k), \int p(x)dx = 1 \tag{10.2}$$

We assume that the one-dimensional data x, which belongs to class k, obey the normal distribution rules, with the mean μ_k and σ_k^2 (the variance) given by:

$$p(x \mid C_k) = \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left(-\frac{1}{2\sigma_k^2}(x-\mu)^2\right) \tag{10.3}$$

According to Bayes' theory, the optimal classification of x is given to the class with the maximum posteriori probability $p(x \mid C1) > p(x \mid C2)$ or vice versa.

Instead of comparing the posteriori probabilities, [97] compares the logarithms of the posteriori probabilities. Simplifying for one class, the result is:

$$g_k(x) = \log P(C_k) - \frac{1}{2}\left(\log(2\pi\sigma_k^2) + \frac{1}{\sigma_k^2}(x-\mu_k)^2\right) \tag{10.4}$$

Thus, the Bayesian decision function for the two classes is given by:

$$D_{Bayes}(x) = g_1(x) - g_2(x) = \log\frac{P(C_1)}{P(C_2)} - \frac{1}{2}\left(\log\frac{\sigma_1^2}{\sigma_2^2} + \frac{1}{\sigma_1^2}(x-\mu)^2 - \frac{1}{\sigma_2^2}(x-\mu)^2\right) \tag{10.5}$$

If $D_{Bayes}(x) > 0$, x is classified into class C_1, and if $D_{Bayes}(x) < 0$, class C2.

10.2 Example 1:

Let us now see how the formula can be applied to an everyday life example.

In this example, we want to identify the sex of an unknown person using only their height.

To do so, first we have to collect the data from the population. The results indicate that the heights of a group of women (h_f) are normally distributed between 145 cm and 185 cm, and the heights of a group of men (h_m) are normally distributed between 165 cm and 205 cm, with an overlap of 20 cm between hf and hm (165-185 cm). See Figure 10.1.

After this, I created a test set formed of 10 individuals, and predicted the sex of each individual in the test set using Equation 10.5. The results can be seen in Table 10.1.

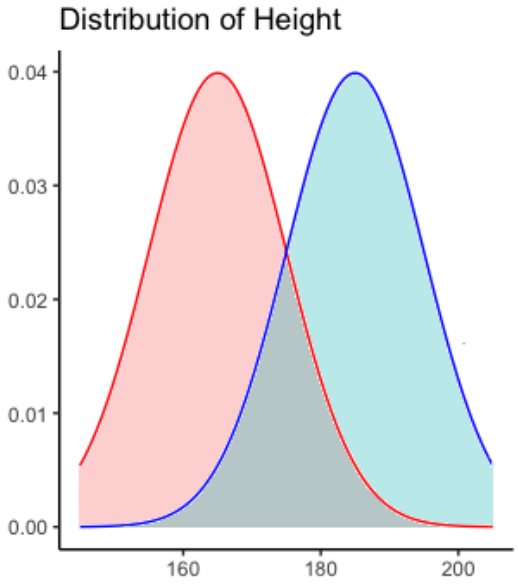


Figure 10.1: Example of two classes within a population: In this plot I report the distribution of height of a hypothetical population divided by sex. On the left-hand side, in red, there is the distribution of height for the female class and on the right-hand side the male distribution. We can see that on average the men are taller than women and that there is an overlapping are between the two classes. Given this data we could use the height of individual of unknown sex as predictor of his/her sex. And the more the height is far from the mean between the two classes and the surer we could be.

Known heights	1-DBF Results	Predicted Sex
145	4.36	F
151	3.39	F
158	2.42	F
165	1.45	F
171	0.48	F
178	-0.48	M
185	-1.45	M
191	-2.42	M
198	-3.39	M
205	4.36	M

Table 10.1: Sex prediction using an arbitrary height distribution: Continuing with the example see above, let us see how using male and female human population distributed in base of their height would work as predictor of unknown individuals' sex. In the left column we have the heights of 10 individual of unknown sex, in the central column the result of the 1-DBF formula and in the right column the predicted sex for each individual, those are predicted as male if his result is lower than zero and female vice versa.

When the result is positive, the individual is assigned as female (F). Otherwise, the individual is assigned as male (M).

From the table, we see that the further a result is from zero, the more certain we can be about the prediction.

10.3 Example 2:

Let us now analyse the results of Equation 10.5 by modifying the two elements considered in the formula, mean μ and standard deviation σ.

By setting different values for μ and σ, let us see the properties and the behaviour of 1-DBF.

We see four different scenarios described below:

$$\begin{aligned} &\mu_1 = \mu_2, \quad \sigma_1 = \sigma_2 \\ &\mu_1 = \mu_2, \quad \sigma_1 \neq \sigma_2 \\ &\mu_1 \neq \mu_2, \quad \sigma_1 = \sigma_2 \\ &\mu_1 \neq \mu_2, \quad \sigma_1 \neq \sigma_2 \end{aligned} \tag{10.6}$$

If we plot these four scenarios (as in Figure 10.2), we can suggest that the smaller the overlap between the areas, the more confident we can be in the result of 1-DBF.

From these four scenarios (Figure 10.2), we can see that if the more two populations have a different mean the less is the area shared between them, Therefore, we would have a higher success rate by applying the 1-DBF.

Let us now perform these four hypothetical scenarios and see the results.

Table 10.2 represents 48 different experiments, representing the four different scenarios with different parameters. These experiments were made with the leave-one-out method with 10,000 elements.

Scenario 1, top left: If mean and standard deviation are equal, the populations are equal as well. There is a complete overlap between the populations, and we cannot be sure of the classification. This method has the same reliability as flipping a coin.

Scenario 2, bottom left: The two populations lie on the same mean. The only

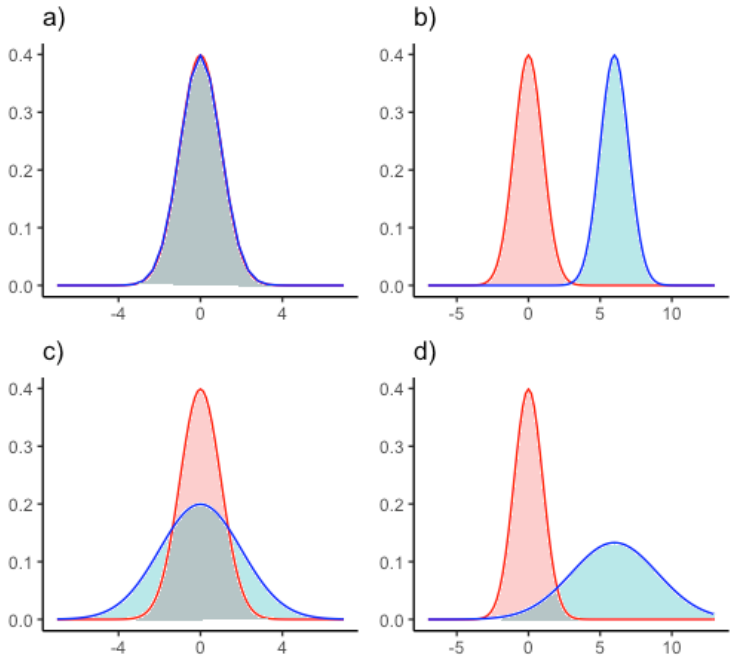


Figure 10.2: Four scenarios for two populations: Within this plot are represented four cases of two population with equal or different standard deviation and mean. The four cases are: **sub-fig. a)** Same mean and standard deviations. In this case the two populations are equal, and the overlapping area between the two is equal to the area under the curves; **sub-fig. b)** Different mean and equals standard deviation. In this scenario the two populations have no overlapping area, and no individual could belong to both of them; **sub-fig. c)** Equal mean and different standard deviations. The two populations have an overlapping area equal to half of each single population area; **sub-fig. d)** Different mean and standard deviation. Similarly, to sub-fig. b, a difference in mean produces a decrease in the area shared between the two populations.

improvement is when we increase the difference of the standard deviation. This is due to a decrease in the overlap between the two populations. In any case, the success of this process increases slowly, and it could never lead to a perfect classification.

Scenario 3, top right: The results increase, moving from left to right and from bottom to top. Gradually, there are narrower standard deviation and a higher distance of means, until there is no overlap at all. We can see immediately that small changes in the means lead to greater results. The mean has a greater impact on the formula, but its contribution is still not enough for all scenarios.

Scenario 4, bottom right: This last scenario confirms the previous one. Moving from left to right (thus increasing the distance of means) the results increase. From

		Mean of population 1 and population 2					
		1, 1	5, 5	10, 10	1, 2.5	1, 5	1,10
SD	1, 1	50	48.32	49.66	77.41	97.64	100
	5, 5	49.75	49.47	49.79	55.39	65.22	81.33
	50, 50	48.95	49.49	49.76	50.74	50.75	53.73
	10^2, 10^2	49.38	50.48	49.95	50.11	50.62	51.76
	1, 5	81.97	82.35	82.16	82.8	86.36	95.55
	1, 25	95.46	95.45	95.41	95.37	95.47	95.57
	1, 50	97.49	97.53	97.61	97.48	97.67	97.43
	1, 100	98.62	98.71	98.71	98.71	98.69	98.57

Table 10.2: 1-DBF Test experiments: This table reports the classification results of 48 experiments of two populations with different means of standard deviation, trying to reproduce what we saw in Figure 10.2. For each experiment, the populations are composed of 100 points, of which distribution has a set mean and standard deviation. The results can be divided into four groups. Top left (in yellow), populations with same standard deviation (SD) and mean: here the classification results are all around 50%, thus equal to random classification. If we recall Figure 10.2, on the top left scenario the two populations were identical; therefore, the formula assigns the point with the same accuracy of randomness. The average of all these experiments is 50%. Top right (in red), **same SD and different means:** for this group of experiments we can see that the success rate grows the smaller the SD and the larger the distance between the means. In Figure 10.2, we saw that this scenario was the one with smallest shard area. The average of all these experiments is 77.41%; but the highest accuracy value is also present here (100%). Bottom left (in green), **different SD and equal mean:** for this group the success rate grows, with a particular increase in one of the SD. However, even with small variations of SD and mean the success rate is already high. Average of all these experiments is 81.97%. Bottom right (in blue), **different SD and mean:** in this scenario we can see that the success rate grows when one of the SD grows and, in this group, we have the highest average success rate of 82.8%. From this table we can understand that the further the mean is between the populations and the smaller the standard deviation, the higher the success rate.

top to bottom, the increase of one standard deviation decreases the overlap, leading to better results.

In conclusion, the best results occur with the use of great differences in means and small standard deviations.

Chapter 11

Feature selection using one dimensional naïve Bayes' classifier increases the OSR of support vector machine classification of CDR3 repertoires

In this chapter I am reporting my work made with the 1 Dimensional Bayesian Function (1-DBF) for the classification of the murine CDR3 repertoires. The work reported herein has been published in Bioinformatics [17].

11.1 Overview

As we saw in the previous chapter, I can reach an overall success rate of circa 60-70% for the classification of OVA and CFA mice repertoires by applying the SVM classification. In this experiment, I wanted to improve such success rate: to do so, I thought to introduce a 'selection step' between the BOW and the SVM in which all features are evaluated, sorted, and filtered out, and by doing this the number of features is decreased.

11.2 Analysis

From each mouse repertoire, 10,000 sequences have been selected, divided into k-mers and counted, forming one numerical vector. This process has been repeated to include single amino acids (singles), duplets, triplets, and quadruplets.

In the previous experiment, we continued by classifying the repertoires using the SVM classification system. However, here my intention was to apply a filtering system that would rank the feature, before being downsized and used in the SVM.

Therefore, adjustments needed to be made to the process.

The system chosen to evaluate each feature is the 1-DBF explained in the previous section. Similarly to the SVM, this method requires a training and test set in order to work properly. If we would, train/test the 1-DBF and use its results to train/test the SVM we will fall into an overfitting error.

Therefore, the right approach is to leave one mouse repertoire as a test set for the SVM, then train/test the 1-DBF, then train the SVM and finally test it with the test set left out at the beginning. In this way, this became a double leave-one-out experiment.

In further detail:

- A first mouse repertoire is left aside to be used as the SVM test sample. This mouse is called the “outer test set”.
- The remaining repertoires are the “outer training set”. This is used for the 1-DBF process. Here again, one repertoire is left out as an “inner test set”, while all the others form the “inner training set”.
- Within the 1-DBF each “inner test set” is composed of a single feature of the codeword (a.k.a. one p-tuples) and the same feature in all other repertoires form the “inner training set”. In this way we get a classification value for each single p-tuple.

Left to do, then, is to sort the p-tuples in decreasing order. One result is presented in Figure 11.1, showing all duplets features ranked by their classification value by the 1-DBF.

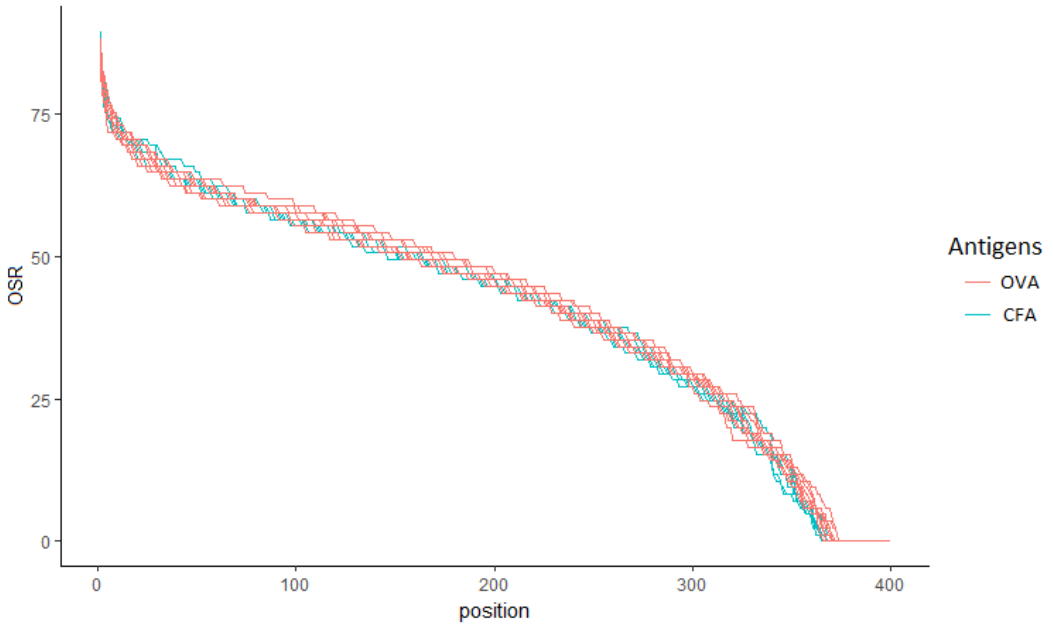


Figure 11.1: Example of duplets sorted by 1-DBF: In this plot is reported the OSR value for each duplet for each mouse repertoire sorted in decreasing order. Each line in the plot is a different repertoire. We can see that a handful of duplets have a very high OSR value which drops quickly to the 50% area, and one handed circa of duplets have values lower than 25%. The duplets with zero values are the duplets absent in the repertoire hence the value cannot be computed.

As we can see from Figure 11.1, the features with a very high classification rate comprises of a small group, then it decreases rapidly to low values. Null values occur due to the fact that many features are partially or totally absent in each repertoire.

Once all features are ranked and sorted, we can test the original "outer test set" of the SVM.

Here, contrary to the 1-DBF stage, the train/test is not formed by one single feature, but by a subset of all features. We start with a subset of two highest p-tuples and add one feature at time until all are considered. Figure 11.2 shows the results of the SVM classification for single amino acids codeword and the first 100 duplets, triplets and quadruplets.

From Figure 11.2, we can see that success rate usually starts from a low value, reaches a maximum pick quickly, and then decreases when the number of features increases.

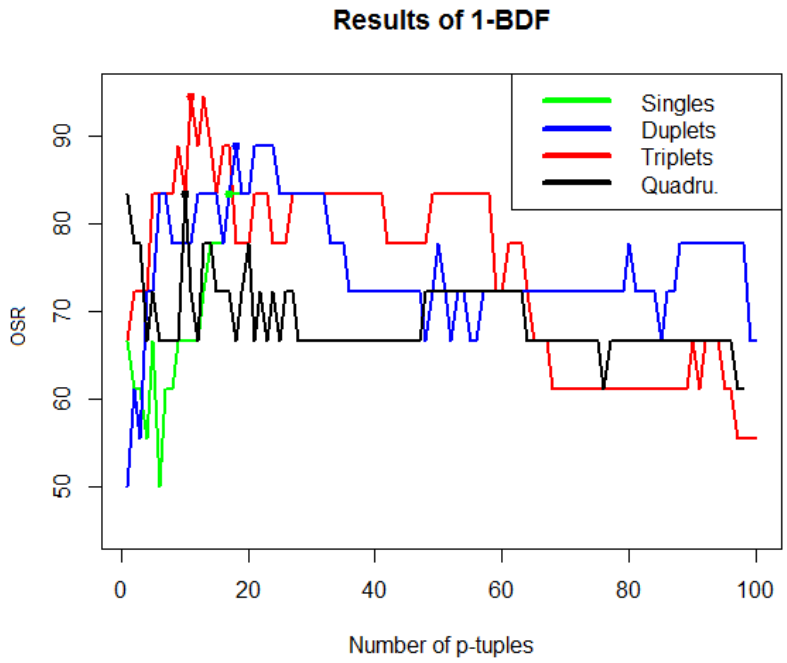


Figure 11.2: Success rate for singles to quadruplets for the first 100 *p*-tuples: In this plot, I report the success rate for single amino acids and the first 100 duplets, triplets and quadruplets. After I had ordered the *p*-tuples using the 1-DBF, I computed the SVM classification score using an increasing number of tuples (starting from the first two, then proceeding with the first 100 —or, in the case of singles, to 20). We can see that: for single, the highest score is reached when all features are used with a score higher than 80%; for duplets, the score floats between the 70-80% value; for triplets, after a spike around 12-14 features with the highest OSR score, the classification value drops around 60%; and for quadruplets, the classification scores range over 70-80% with smaller variation after the thirtieth position.

The minimum number of features needed to obtain the highest SVM classification value seen in Figure 11.2 is computed post-facto, as it is not possible to determine it a priori.

The highest result for each *p*-tuples is reported in Table 11.1:

From Table 11.1 we can see the result for each of the *p*-tuples. The highest value is for the triplets, where only one mouse is misclassified (and which is, incidentally, always misclassified).

	Singles	**Duplets**	**Triplets**	**Quadruplets**	
Day5_1	100	100	100	100	OVA_1
Day5_2	100	100	100	100	OVA_2
Day5_3	90.9	100	100	100	OVA_3
Day14_1	100	100	100	100	OVA_4
Day14_2	100	100	100	100	OVA_5
Day14_3	100	100	63.6	60	OVA_6
Day60_1_1	100	100	100	0	OVA_7
Day60_1_2	81.81	100	100	100	OVA_8
Day60_1_3	100	100	100	100	OVA_9
Day5_4	100	100	100	60	CFA_1
Day5_5	100	18.1	100	100	CFA_2
Day5_6	100	100	100	60	CFA_3
Day14_4	0	0	0	20	CFA_4
Day14_5	100	100	100	100	CFA_5
Day14_6	90.9	100	100	60	CFA_6
Day60_1_4	0	0	72.7	0	CFA_7
Day60_1_5	100	100	100	100	CFA_8
Day60_1_6	0	90.9	100	100	CFA_9
OSE	83%	89%	94%	83%	
# of features	17	17	12	11	

Table 11.1: Maximal SVM success rate with minimal number of *p*-tuples: In this table, I report the maximal SVM success rate generated by the minimal number of *p*-tuples ordered by the 1-DBF. In each row is present the mouse repertoire label as timepoint (first column) and antigen-infected (last column), and the classification score for each different *p*-tuples (central columns). In the last two rows is present the success rate and the number of features used for this classification. All misclassified mice are highlighted in red. All classification score range between 83% and 94% with the highest score produced by the first 12 triplets.

To prove that this improvement in the success rate is not driven only by a smaller number of features, but is due also to a higher informative feature, I repeated the SVM experiment 100 times. I used an increasing size of random triplets: 25, 50, 75, 100, 125 and 150. The result is reported in Figure 11.3.

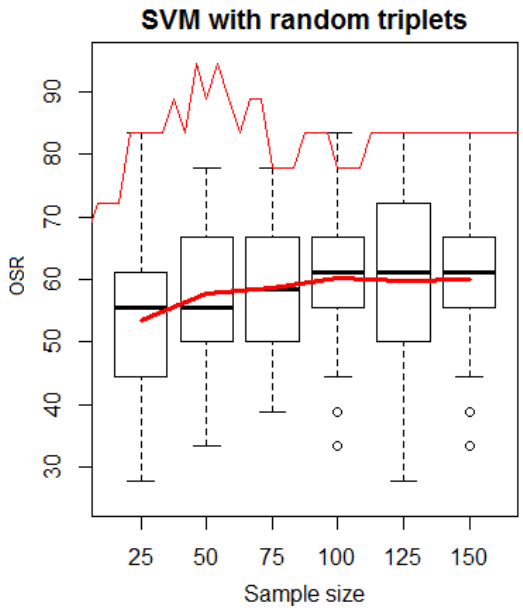


Figure 11.3: Success rate by 1-DBF vs. random subsets: In this plot I tested if the classification score of 94% with 12 triplets could be reproduced using random subset of triplets of increasing size. In the plot, the thin red line is the behaviour of the success rate (as seen in Figure 11.2), the boxplot is the results of 100 SVM test with 25, 50, 75, 100, 125, 150 random triplets. The solid red line is the mean of each boxplot. We can see that all results are lower than the one performed with our ranked triplets. This plot proves that the results obtained with the combination of 1-DBF-SVM cannot be reproduced or originated stochastically.

All the results for the 100 SVM experiment with random triplets are reported in Figure 11.3. We can now see how the average results (solid red line) is much lower than our results (thin red line).

Our highest value is 94%, which was never reached by the random tests; ranked triplets kept outperforming the random test, even after 150 features.

Considering the very high results with the 12 triplets, I wanted to see where these triplets are present on the CDR3. The result is presented in Figure 11.4.

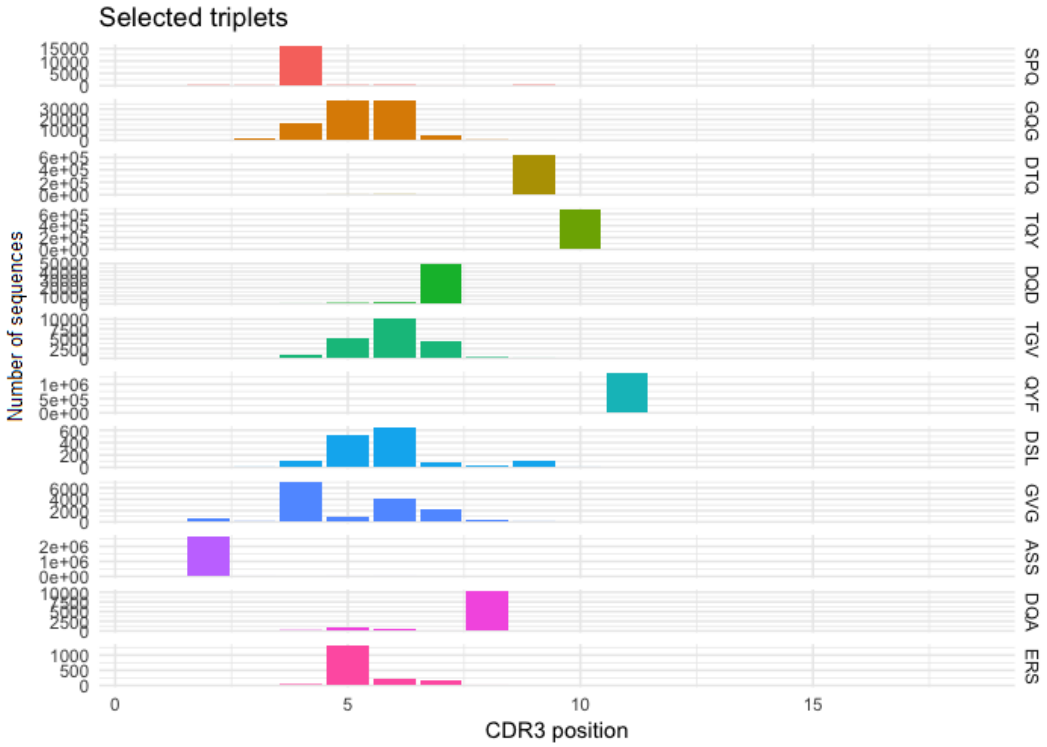


Figure 11.4: Position of the 12 selected triplets on the CDR3: In this plot is reported the position of each of the selected 12 triplets on the CDR3. For each plot I counted the starting position of each triplets for all CDR3 of length 16 in the entire repertoire. On the *y*-axis is reported the number of times present and, on the *x*-axis, the starting position. As we can see that some triplets are presents in unique positions like: ASS, QYF, TQY, SPQ, DQD and TDQ. Other are more spread and mostly present in the left-hand side of the CDR3.

As we can see, the great majority of all triplets are found principally at the beginning of the central region of the CDR3, in particular around the 5th amino acid position.

11.3 Discussion

Thanks to the use of 1-DBF as features selection method, I was able to boost the success rate. With the use of triplets, the overall classification OSR went from 61% of Table 8.10 to 94% of Table 11.1. This is a very significant result, considering that this type of classification has been one of the hardest. Only with this method I could obtain such good values.

This result demonstrates that specific antigen immunisation can give rise to changes in the TCR repertoires which are consistent and trackable, even when co-

exposed to a mixture of other antigens. Even so, genetically identical mice may contain largely disparate set of repertoires —we see this in our control mice.

This result with low level features of protein sequences like triplets support the idea [16] that triplets are the best elements for the study of CDR3. Triplets seem to be a good midpoint between having a small patch of sequences, but still being informative for computational and biological analysis.

Therefore, these are promising indications for applying this kind of approach to the analysis of clinical samples for the prognosis and diagnosis of patients in the contexts of both infectious and non-infectious (e.g. cancer, autoimmunity, transplantation) disease [17].

The method used here is composed of a double leave-one-out cross-validation system in which I use the 1-DBF to filter and evaluate the features, and then apply the SVM methods to search a subset of those features that produce an optimal result.

In recent years, other studies have been performed in different fields of machine learning with an approach similar to the one adopted here: high-throughput data feature selection followed by a classification method, often SVM [125][126][127][128][129][130][131][132].

All of these methods can be divided into a filter method (which is defined as a criterion to rate and sort the features) and a wrapper/embedded method (in which a classifier is used to search an excellent subset of all features) [133]. However, to avoid any over-fitting issue, the best solution is to apply a double leave-one-out cross-validation; but, at least for my method, it requires a long time and great computational effort. This is probably the most impactful limitation of this approach. In addition, if applied to larger databases, it would probably produce even more accurate results, but at the cost of increased calculation, effort and time.

Using the 1-DBF to find a way to select the most informative p-tuples, I selected 12 triplets that are in positions that are considered of primary importance for the interaction with the peptide. They mostly lay around position 5, this area of the CDR3, as see in section 3.8 and in [12] has been considered relevant for peptide recognition by CDR3.

This result suggests that the use of the 1-DBF has not only increased the classification rate of the repertoires, but it has also selected potentially relevant position and triplets for the antigen recognition.This method is not a valid clinical diagnosis tool [134], but at least a promising classifier of CDR3 sequences in accordance to their specificities [135][136].

The Hidden Markov Model as a way to identify CDR3 groups of sequences and classify them

Chapter 12

Introduction on Markov Chains Models

After my work on the SVM and the Bayes' theorem, reported in the previous sections, in the last year of my doctorate, I moved my attention to another important machine learning and data mining technique: the Markov models.

The Markov Chains (MC) [137][138] and the Hidden Markov Model (HMM) [139] are powerful statistical models that can be applied in a variety of different fields, such as: protein homologies detection [140]; speech recognition [141]; language processing [142]; telecommunications [143]; and tracking animal behaviour [144][145].

HMM has been widely used in bioinformatics since its inception. It is most commonly applied to the analysis of sequences, specifically to DNA sequences [146], for their classification, [147] or the detection of specific regions of the sequence, most notably the work made on CpG islands [148].

In the following chapters, I illustrate my findings in the application of a specific program, Hammock [149], a hidden Markov model-based clustering algorithm. This program can find conserved motifs in short amino acid sequences and its results can reveal interesting characteristics of the CDR3 sequences and motifs of the repertoires.

Furthermore, I create a SVM classification system in which the features of the SVM are not the sub-string of the CDR3 sequences, but the results of Hammock

turned into Profile HMMs.

In the following chapters, I will present an introduction to the Markov chain and HMM, and their application with protein sequences. This will be followed by an explanation of the programs used, and finally by the analytical experiments I performed.

Chapter 13

Hidden Markov Model Theory

In the first part of this chapter, I will describe the classical types of Markov Chain and HMM that are usually studied. In the second part, I will focus on a special case of HMM applied to protein sequence analysis, named Profile HMM [150][151][152].

Profile HMM represents a subgroup of HMM called "Left-to-right HMM" [153]. They may appear disorienting to the reader familiar with classical HMM. Nevertheless, Profile HMMs are the state of the art for sequence analysis, and I hope to provide a clear explanation.

13.1 Overview

The Markov Chain models can be applied to all situations in which the history of a previous event is known, whether directly observable or not (hidden). In this way, the probability of transition from one event to another can be measured, and the probability of future events computed.

The Markov Chain models are discrete dynamical systems of finite states in which transitions from one state to another are based on a probabilistic model, rather than a deterministic one. It follows that the information for a generic state X of a chain at the time t is expressed by the probabilities of transition from the time: $t-1$.

In HMM, the previous rules are observed, but the observer can only see the output of the function associated with each state —not the states directly. In other words, the observer can see the event produced at the time t but cannot observe the

state that has produced the output.

13.2 General Theory

13.2.1 Markov Chain

A Markov Chain provides a theoretical model to describe the behaviour of a discrete-time system.

Definition: For a finite state space $S = \{s_1, s_2, \ ... \ , s_N\}$ with a sequence of random variables $X_1, X_2, \ ... X_n$, assuming values in S for which the transitional probability P, of the state s_j at the time t, is given by the transitional from the state s_i and the time $t-1$, with probability p_{ij}, thus the probability of transition from state s_i to sj (Markov assumption).

In other words, the probability of a state S, at time, is given only by the immediately preceding state; therefore, all events before $t-1$ can be ignored. For this property, the Markov Chain is called memoryless.

The Markov assumption is also known as the Markov Chain of first-order. A Markov Chain of second-order would have the observation at time t depending on time $t-1$ and time $t-2$; however, Markov Chains of orders higher than 1 are rarely used.

As a consequence of the Markov assumption, the entire behaviour of a Markov Chain can be described using a transition matrix. The dimension of a transition matrix is the number of states of the Markov Chain, and all the elements describe the probability of moving from one state to another (or back to the same state):

$$q_{ij} = P\left(q_t = S_j | q_{t-1} = S_i\right) \ 1 \leq i, \ j \leq N$$
$$A = (a_{ij})_{i=1,j=1}^{n,n} = \begin{pmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ a_{n,1} & a_{n,2} & \cdots & a_{n,n} \end{pmatrix} \tag{13.1}$$

Where the following property is true:

$$a_{ij} \geq 0, \sum_{j=1}^{j} a_{ij} = 1 \tag{13.2}$$

The Markov Chain can be described as a triple (S,X,P), a set of states S, with X random variables and a transition probability matrix P.

13.2.2 Example

Let us imagine two events A and B with the following transitional matrix:

$$\begin{array}{rccc} & to & A & B \\ p = & A & 0.9 & 0.2 \\ & B & 0.1 & 0.8 \end{array} \tag{13.3}$$

Every time the system is in the state A, there 90% probability that system will remain in state A for the next time. Every time the system moves from state A to B, it will stay in B for 80% of the times.

We can compute the probability of having the succession of events ABBAB, given a starting A, to be:

$$p = \begin{array}{c} P = P(AA)P(AB)P(BB)P(BA)P(AB) \\ 0.9 \cdot 0.2 \cdot 0.8 \cdot 0.1 \cdot 0.2 = 0.00288 = 0.288 \end{array} \tag{13.4}$$

13.2.3 Hidden Markov Models

As seen so far, for the Markov Chain, each state corresponds to an observable event. However, this type of model appears to be too restrictive, and it cannot be applied to many situations of interest.

The situation might occur in which a stochastic process producing a set of outputs has an underlying process that is not observable, thus the 'hidden' label.

The Hidden Markov Models can be considered as a quintuple N,M,A,B,μ with the following elements:

1. N, hidden states.

2. The observable symbol per state M. What type of output can come out from

each state.

3. The transition probability matrix $A = a_{ij}$.

4. The probability distribution of the observable output.

5. Probability for the initial state, μ.

Transition matrices and emission probability are usually calculated with a pseudo-count [154][155]. This consists of adding a small quantity, typically 1, to all the states in all positions, before calculating the probabilities, to avoid zero-values that could compromise future results.

13.3 Application on Biological sequences

As seen thus far, MC and HMM are powerful methods that can be used for a large variety of purposes. However, we use a special case of HMM named Profile HMM for the study of biological sequences. In the following section, my description of this system should explain the reasoning behind the use of Profile HMM.

13.3.1 Analysis of a MSA

Let us consider a set of functionally related DNA sequences. Our objective is to characterise them as a 'family', and consequently identify other sequences that might belong to the same family [154].

We start by creating a multiple sequence alignment to highlight conserved positions:

$$\begin{array}{cccccccccc} A & C & A & - & - & - & A & T & G \\ T & C & A & A & C & T & A & T & C \\ A & C & A & C & - & - & A & G & C \\ A & G & A & - & - & - & A & T & C \\ A & C & C & G & - & - & A & T & C \end{array}$$

It is possible to express this set of sequences as a regular expression. The family pattern for this set of sequences is:

$$[AT][CG][AC][ACGT]^{*}A[TG][GC]$$

Each position in the regular expression represents the nucleotides in the chain. Multiple options for each position are gathered in a bracket: thus, the first element could equally be an A or a T, the second one a C or G, and so on. The element indicated with a * represents a gap area: only the A is not bracketed, because it is the only possible option of that position.

The regular expression is useful because it allows us to spot the pattern of this family of sequences in a visual and simple compact view. However, the regular expression is not an adequate method when establishing whether other sequences are part of this family.

As an example, let us consider two new sequences 1 and 2:

$$\begin{array}{lcccccccc} 1: & T & G & C & A & - & - & A & G & G \\ 2: & A & C & A & C & - & - & A & T & C \end{array}$$

Both sequences fit the regular expression given above and, based on that alone, they could be considered part of the family. However, we can see that the first sequence is formed by the nucleotides occurring the fewest times in the multiple sequence alignment, while the second is formed by those most common. Indeed, in the first position, the T is present only once in the multiple sequence alignment, while A in all other sequence, similarly for the in second position, the G only once and C, for all remaining sequences.

We need a way to measure the "distance" between a new sequence and the original set of family sequences. To solve this problem, we can use MC and HMM.

13.3.2 Markov Chain

The nature of the state is arbitrary: in this case, we can choose to take four elements present in the sequences, thus: A, T, C, G, and create the following Markov Chain (see Figure 13.1).

From the model in Figure 13.1 we can start computing the probability of tran-

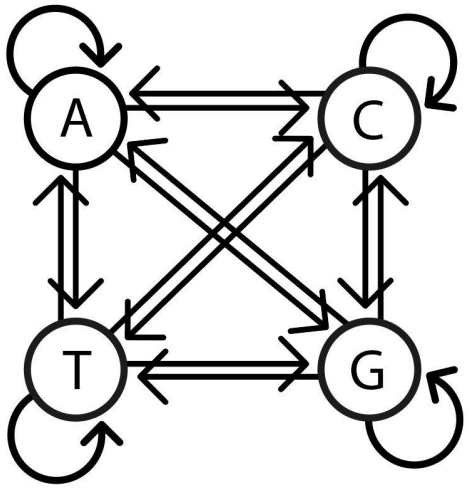


Figure 13.1: Graphical representation of a Markov chain model for DNA: Each circle represents a state of the chain: four states connected with the DNA nucleotides (A, C, G, T). Each edge is the probability of transition from one state to another or to itself.

sition in accordance with the alignment seen previously. As such, we can begin calculating the probability of passing from the nucleotide A to G or from G to T, for example, and forming a transition matrix.

Using the transition matrix, it is possible to obtain the probability of a new sequence through the following relation:

$$P(seq_{test} \mid M_T) = \prod_{i,j=1}^{I,J} P_{ij} \tag{13.5}$$

Where the probability of a sequence (with respect to a transition matrix (M_t)) is given by the product of the transition probability from one state to another of the sequence.

The result obtained through this equation is an indication of how much a sequence can be considered part of a family: the higher the value obtained, the greater the extent to which a sequence can be considered part of the family.

However, the transition matrix is being obtained regardless of the positions of the states within the sequences. This aspect is not negligible, particularly regarding the study of DNA and protein sequences, where undeniably the position of a nucleotide is more important than its numeric presence.

In order to adapt the Markov Model to the study of sequences, the concept of Profile HMM has been introduced.

13.3.3 Profile HMM

The Profile HMM is a variation of the Markov Chain in which the position of a multiple sequence alignment become the states of the model; the transition matrix is the probability to pass from one state/position to the next [154]. In this way, the probability of emission for each state is introduced, and thus the probability to have a particular nucleotide on that state of the Markov Chain.

$$[AT][CG][AC][ACGT]^{*}A[TG][GC]$$

We can rewrite the regular expression presented in the previous section (and above), delivering the assumption of the previous paragraph. Figure 13.2 clarifies this concept:

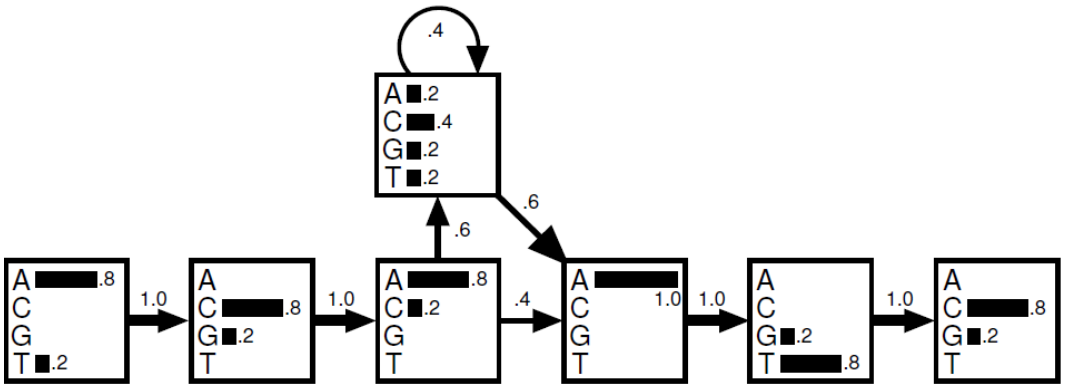


Figure 13.2: Scheme of a Profile HMM: In this plot the multiple sequence alignment taken as example above, has been represented as a Profile HMM, each position of the multiple sequence alignment is a state, for each amino acid are reported the probability of emission of a nucleotide in that state and, the arrows are the transition probability between the state. Source [154].

Figure 13.2 is a visual representation of the application of the HMM method to the case presented here. Each of the boxes is a state of the HMM, corresponding to the position of the multiple sequence alignment or to each group of the regular expression. Inside the boxes are the possible emissions of the state and the probability of each one, while the external arrows exemplify the probability of transition

from one state to the other.

The state on the top of Figure 13.2 is the only one with the property to stay still, rather than to move to a different state. This property can be used to model an insertion state.

To generalise the scheme even further, more states can be added with different properties until a Profile HMM is formed. The general standard architecture of a Profile HMM is presented in Figure 13.3:

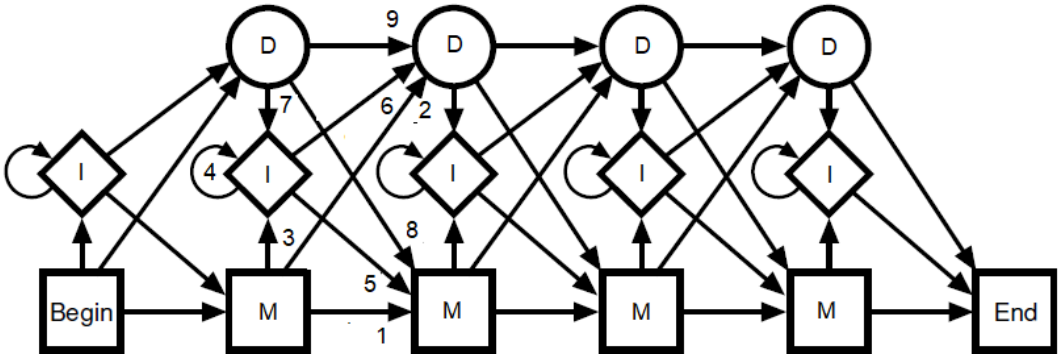


Figure 13.3: General representation of the structure of a Profile HMM: This is a general representation of the structure of a Profile HMM. There is both a 'begin' state and an 'end' state. The square represents the matches' states (M), and the circles and diamonds represent deletion (D) and insertion states (I). The addition of insertion and deletion states make it possible to train and test sequences of different length. The arrows are the transition probability and because for each position, except for the Begin and End state. Because there are nine arrows, this type of Profile HMM is called Plan 9. Original picture from [154].

In Figure 13.3, the Profile-HMM has three hidden states [150], plus a "Begin" and "End" state. The squares are the "matches" states, which represent the frequency (emission) of the amino acid or nucleotides.

The diamonds are called the "insert state". They are used to model the gap insertions of the alignment. They are usually used to test sequences longer than the consensus sequences [150].

The circles are the "deletion states". They play the role of the silent or null state. They do not match residues or gaps and are used to jump from one column to another. These are usually used for sequences shorter than the consensus sequences.

"Begin" and "End" states have been introduced to help the transaction at the beginning and the end of the Profile HMM. They do not produce any emissions.

The arrows represent the transition probabilities. In Figure 13.3, nine arrows are presented in a module (M, I, D states). This configuration is named Plan 9.

Profile HMM are widely used in biological sequences analysis. They have been proven to be useful for protein classification, motif detection, multiple sequences alignment [156], and protein secondary structure prediction [157]. In general, all of them can be summarised in terms of their major uses: aligning a sequence to a profile and scoring a sequence against a profile.

To evaluate a sequence against a profile and obtain a score, more efficient dynamic programming methods than the ones presented above are available. The Forward algorithm [158] identifies the likelihood of a sequence; the Viterbi algorithm [159] identifies the most probable "path" to generate a given sequence; and the Forward-Backward (Baum-Welch) algorithm, used also to train the Profile HMM.

Chapter 14

HMM-Based Programs

Below are brief descriptions of the three main programs I used in the work presented here.

14.1 HMMer

HMMer [160][161] is a free program, generally used to identify homologous DNA or protein sequences by comparing Profiles HMM to a single sequence or entire databases.

An alternative programme mentioned in the literature is SAM [162]. SAM is a low-level program with few automatic options, and it has not been updated since its first publication. It has also been referenced in fewer publications and has a smaller online community.

In contrast, a newly upgraded version of HMMer (HMMer3) was released in early 2015 [163]. It has been used for many recent papers. References and explanations of the underlying algorithm can be found on its own website [161], as well as in academic texts [153][164].

HMMer is generally used to identify if a sequence or a group of amino acids/nucleotides sequences can be considered as member of a family present in private or public databases, as in the case for Pfam [147][165].

From a set of aligned sequences, HMMer allows the creation of a Profile HMM and the subsequent comparison of the HMM to a single sequence or an entire database. Sequences that score higher than a chosen threshold are considered

homologous to the sequences forming the profile-HMM.

In HMMer, there is a small variation in the architecture of the Profile HMM. In particular, there is no transition allowed from delete states to insert states, or vice versa.

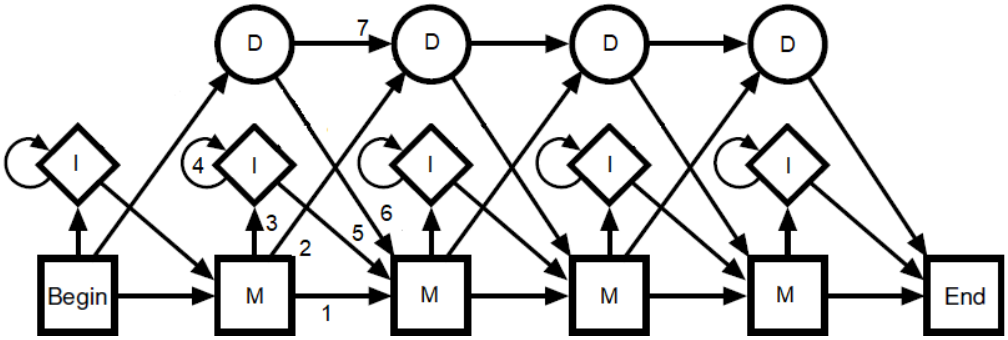


Figure 14.1: Plan 7 in HMMer: Graphical representation of Profile HMM used in Hammock. This kind of Profile HMM architecture has been named Plan 7 because the number of non-zero transition (arrows) in a module (M, I, D, i.e. match, insertion and deletion states, respectively) are reduced to seven. This is in contrast to a normal Profile HMM, which would be Plan 9. This is present in Figure 13.3. [154].

This architecture has been named Plan 7 (see Figure 14.1) because the number of non-zero transition (arrows) in a module (M, I, D states) are reduced to seven [166]: in fact, there are no transition from delete to insert states or conversely from insert to delete [153]. HMMer relays on the Viterbi algorithm for the plan-7 architecture, the algorithm will compute the most probable/maximum-likelihoodre path through the state model [167].

14.2 HH-Suite

HH-suite [168] is an open-source program able to perform pairwise alignment of hidden Markov models. Thanks to this, it is possible to identify if a protein sequence belongs to a database of HMM protein families. This program has been used in a plethora of different bioinformatics tools such as ClustalΩ [169] and Hammock, among others.

In this work, we use principally the functions `hhalign` and `hhmake`. Using the latter, we can turn the resulting multiple sequence alignment of sequences found

by Hammock into a Profile HMM, and with the former we can calculate the pairwise alignments of two different HMMs and a sequence dissimilarity value that can be used as a "distance" between two HMMs. With such a distance, we can create trees and identify groups of similar Hammock results, in order to check the presence of similar clusters among different mice.

14.3 Hammock

The Hammock program [149] can find conserved positions inside large repertoires of short linear sequences. The basic idea of the algorithm is a progressive cluster growth. A few sequences are considered as seeds of new clusters, followed by iterative cycles of two alternating steps: a cluster extension step, where sequences are inserted into the previously formed clusters, and a cluster merging step, where whole clusters are compared and merged.

Two other papers adopting the same progressive clustering growth strategy can be found in the literature: [170] and [171]. Both address the same aims, but [149] could combine the best features and ideas of the previous two with state-of-the-art of bioinformatics tools.

14.3.1 Workflow

As mentioned, the basic idea is a progressive cluster growth. The workflow can be outlined as follows

1. Pre-processing: All duplicate sequences are removed, and the number of duplicates is preserved. The sequences are then sorted by length.

2. Initial greedy clustering: the aim of this step is to identify a small group of very similar sequences using the "Database complexity reduction algorithm" [78], in which:

   (a) First sequence is used as new cluster seed.

   (b) From the second onwards: the sequence is compared, and if its similarity is higher than a pre-defined threshold, it became part of the cluster. Otherwise, it forms a new cluster seed.

3. Cluster selection and alignment: A multiple sequence alignment of all cluster is generated.

4. Cluster extension: Groups are turned into Profile HMMs, which are then used to search for similar sequences. Any sequence with e-value higher than a threshold is added into the appropriate group. HMMer is used for both HMM construction and sequence search (`hmmbuild`, `hmmsearch`). Local alignments are performed using ClustalΩ.

5. Cluster merging: Some clusters may be very similar to each other.

   (a) Similar clusters are identified and merged into larger clusters.

   (b) Local HMM-HMM alignment routine provided by HH-suite [166] is used to measure cluster-cluster similarity.

   (c) The cluster merging step is a bottom-up hierarchical clustering process (the strategy adopted remind the UPGMA algorithm).

   (d) It runs in $O(n^2)$ and it merges only the most similar pair in every step.

6. Iterating the extension and merging steps: Extension and merging steps are repeated (three times by default).

7. Measure of clustering quality using Kullback—Leibler divergence (KLD) [170].

8. Sequences consensus creation (WebLogo [172][173]).

14.3.2 Consideration

The Hammock program algorithm has been specifically designed to cluster short peptide sequences, therefore making it suitable for 16 amino acids average-long CDR3s. It can cluster large amounts of data containing noise, and produce multiple sequence alignments of high-quality clusters.

There are no limits in terms of origin and format that can be used as input to the program —this includes any set of peptide sequences, and even other clusters previously originated by the program itself.

The program does not require any prior data knowledge. It is faster, if compared with existing tools, and very flexible as all parameters can be changed as needed.

Chapter 15

Analyses with Hammock

15.1 Introduction

With the application of Hammock to the CDR3 repertoires, I planned to find the most conserved amino acids and positions, common to all repertoires and to identify emerging sequence motifs that would characterise the different mice groups.

With a special focus on mice repertoires immunised with different antigens, I wanted to highlight the different strategies against the antigens adopted by different mice groups. I hoped to find common, rather than unique, response strategies among mice of the same group.

As mentioned in the introduction, there are many papers in the literature discussing the role of public sequences in the repertoires of different individuals. In my analysis, I found the presence of several shared CDR3 sequences among the repertoires. This value was not particularly high but still relevant, with an average of a tenth of the total sequences shared.

In this chapter, I investigate the presence of public and private sequence within the repertoires. But not looking at the CDR3 sequences but focusing upon the motifs that can emerge from the repertoires.

I start by searching for the presence of motifs within each mouse, and then group the mice immunised with the different antigens (OVA and CFA mice) and analyse the results.

15.2 CDR3 Boundaries

In Figure 15.1, showing a sample of a few motifs obtained by Hammock and plotted using WebLogo [172][173], we can see clearly that the motif C, ... ,FG[X]G is present and repeated in all figures.

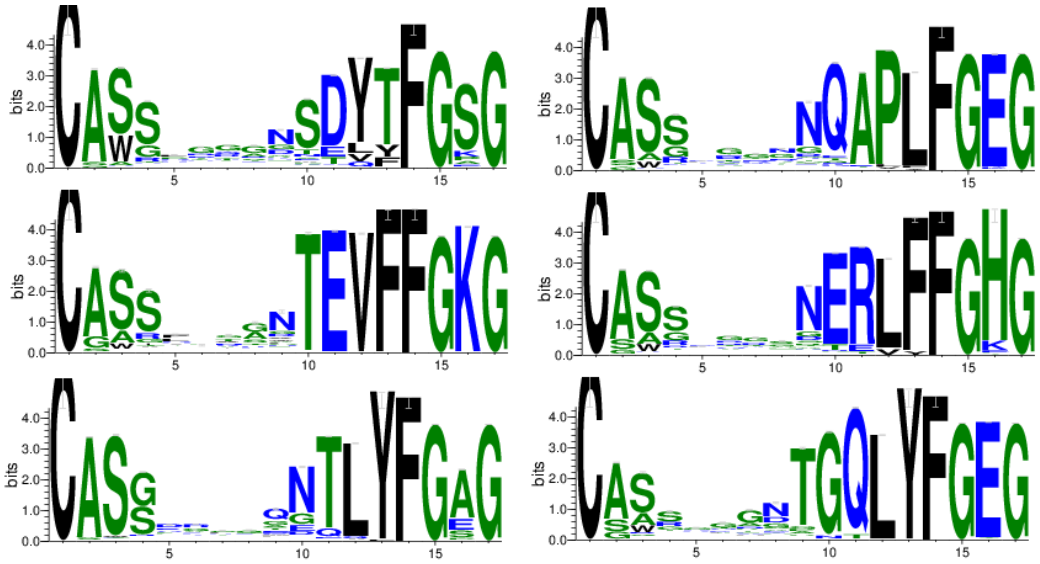


Figure 15.1: Results of Hammock on the entire CDR3 sequence: Here are six randomly chosen examples of the results of using the Hammock on the entire sequence of CDR3. With these plots we can see that the program has found the conservative borders of the CDR3 thus the C and the FG[X]G motif. These were already known, and they are actually the motifs that define the CDR3 itself. Therefore, the program proved itself by founding the conserved position I was expecting to find. For improved clarity, I used only sequences with the length of 17 AA.

Indeed, Hammock looks for the most conserved position among all sequences, using as a reference the most conserved amino acid and positions. But as we saw in the introduction, the CDR3 itself is defined by two border regions: the cysteine on the N-terminal of the sequence and the FG[X]G motif on the C-terminal. Indeed, what the program found are the fixed positions that define the CDR3 itself, rediscovering the boundaries that helped to fix the concept of CDR3 in the first place, and that are used by all programs to determine the CDR3 from the TCR sequence. This is far from a negative result, as it proves the effectiveness of the program and can be considered a valid "positive control" for my investigation.

In the motifs present in Figure 15.1, the N-terminal end (left-hand side) is

present as a cysteine residue, and on the C-terminal end (right-hand side) lies the motif FG[X]G, where X is E, P, K, G, A, S or H.

In Table 15.1, the number of clusters found by Hammock in the mouse repertoires is reported. We see that the number of clusters is relatively low, and often all sequences are reassembled in one cluster. All sequences are gathered together, following the already-known conserved pattern common to all sequences.

Repertoire	**Clusters**	**Repertoire**	**Clusters**
Control_1 1	2	Day_7 5	3
Control_1 2	1	Day_14 1	1
Control_1 3	1	Day_14 2	5
Control_1 4	1	Day_14 3	2
Control_1 5	1	Day_14 4	3
Control_1 6	1	Day_14 5	2
Control_2 1	4	Day_14 6	1
Control_2 2	1	Day_60_1 1	2
Control_2 3	3	Day_60_1 2	2
Day_5 1	2	Day_60_1 3	1
Day_5 2	2	Day_60_1 4	2
Day_5 3	3	Day_60_1 5	3
Day_5 4	2	Day_60_1 6	1
Day_5 5	3	Day_60_2 1	2
Day_5 6	4	Day_60_2 2	4
Day_7 1	5	Day_60_2 3	3
Day_7 2	4	Day_60_2 4	3
Day_7 3	6	Day_60_2 5	5
Day _7 4	4	Total:	95

Table 15.1: CDR3 Boundaries: This table presents the number of clusters found by Hammock per mouse. Because the program tracks only the conserved amino acids and those are always C ... , FG[X]G the number of resulting clusters is very small and, they are all variation of the border. Column description: In the columns Repertoire are listed all the 37 mice repertoire: Control_1: control mice from the first experiment; Control_2: control mice from the second experiment; Day_5: Mice sacrified 5 days post-immunisation; Day_7: Mice sacrified 7 days post-immunisation; Day_14: Mice sacrified 14 days post-immunisation; Day_60_1: Mice sacrified 60 days post-immunisation, from the first experiment; Day_60_2: Mice sacrified 60 days post-immunisation from the second experiment. In Clusters the number of clusters in which the all sequences of each repertoire are gathered. In the last row Total: the sum of all clusters found.

However, if we look at Figure 15.1, we note that other emerging patterns are present at the positions immediately after the first C and before the FG[X]G motif. I

decided to investigate these by physically removing the C and FG[X]G amino acids from all sequences, and then running the program again.

15.3 V and J region tails

When I repeated the program on the CDR3 sequences without the borders motifs, the result was not entirely different from the previous result. A similar number of clusters per mouse is in evidence (see Table 15.2). A random selection of the motifs is reported in Figure 15.2.

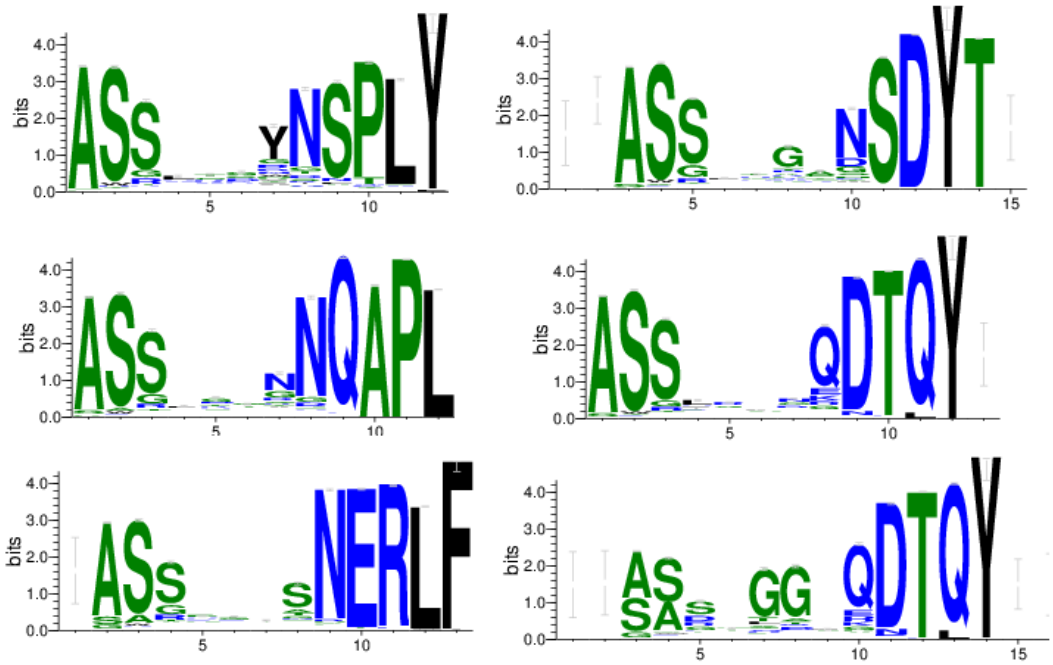


Figure 15.2: Results from Hammock without the CDR3 boundaries: These six randomly chosen examples represent a sample of the 88 clusters found with Hammock. We can note two emerging positions: a single, very conserved triplet on the left-hand side of the pictures; and to the right, a group of four-five amino acids. These two regions are the remaining "tails" of the V and J regions, involved in the creation of the CDR3 during the V(D)J recombination. Furthermore, if we look closer to the bottom left, we can see the presence of a third smaller motif in the centre of the CDR3. It was this that led me to run the Hammock program once again, with sequences stripped of borders and V/J tails.

These results show the emerging motifs of the last part (or "tails") of the V and J region. The CDR3 boundaries correspond to the definition we gave to the CDR3, and these two tails are what I consider to be the result of the V(D)J recombination

processes explained in the introduction.

Repertoire	Clusters	Repertoire	Clusters
Control_1_1	2	Day_7_5	5
Control_1_2	2	Day_14_1	1
Control_1_3	2	Day_14_2	3
Control_1_4	1	Day_14_3	1
Control_1_5	1	Day_14_4	3
Control_1_6	1	Day_14_5	1
Control_2_1	2	Day_14_6	1
Control_2_2	2	Day_60_1 1	1
Control_2_3	1	Day 60_1 2	2
Day_5_1	4	Day_60_1 3	2
Day_5_2	2	Day_60_1 4	2
Day_5_3	2	Day_60_1 5	2
Day_5_4	2	Day_60_1 6	3
Day_5_5	1	Day_60_2 1	5
Day_5_6	1	Day_60_2 2	2
Day_7_1	3	Day_60_2 3	4
Day_7_2	5	Day_60_2 4	4
Day_7_3	4	Day_60_2 5	2
Day_7_4	6	Total:	88

Table 15.2: V and J region tails motifs: Number of V and J region tails motifs present in each mouse using Hammock. For this step the number of motifs is even smaller of the one in Table 15.1. With several mouse sequences gathered in only one cluster. Columns description as Table 15.1.

Looking at Figure 15.2, we can see that the V region (on the left-hand side of the picture) seems to be stable, with the common pattern “ASS”. Meanwhile, the J region (right-hand side), has a different type of structure with different types of combinations.

I proceeded to split all sequences in the two datasets, one with the first three amino acids and one with the last four positions of the CDR3. More simply —one for the V region tail, one for the J region tail.

The results of the analysis of the V region show a small variation in terms of triplet variability in the V region. See Figure 15.3.

I ran Hammock again using this time the J region only. This resulted in twelve different clusters, but without some clear motifs. This small patch of the CDR3 sequence is too small and variable to produce a clear motif.

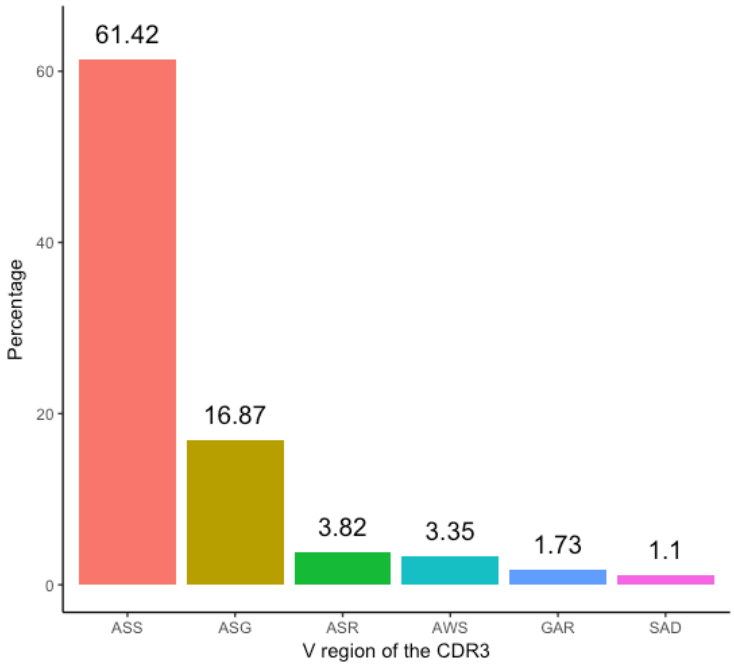


Figure 15.3: V region of the CDR3 in the repertoires: Here, I plotted the most common motifs emerging from my CDR3 formed by the first triplet after the first Cysteine. In order, they are: ASS (61.42%), ASG (16.87%), ASR (3.82%), AWS (3.35%), GAR (1.73%), SAD (1.1%), all other triplets are lower than 1%. The dominant V region triplet is ASS: the motif that we have already encountered as one of the selected triplets with the 1-DBF.

Furthermore, if we look more closely at the picture in the bottom right of Figure 15.2, we can see the presence of a third smaller motif, in the centre of the CDR3. This was the hint that led me to consider the option of running the Hammock program, once again on all sequences stripped of borders and V/J tails.

15.4 The D region of the CDR3

From all of the original CDR3 sequences, I removed the first cysteine in the N-terminal and the last four amino acids in the C-terminal corresponding to the FG[X]G motif. Subsequently, I removed another three amino acids in N-terminal (that I considered the V region) and another four from the C-terminal corresponding to the J region. This accounted for a total of twelve amino acids removed. As such, all sequences shorter than twelve amino acids are not present in this analysis, and the average length of the sequence left is four to five amino acids. The results are reported in Table 15.3.

Repertoire	Clusters	Repertoire	Clusters
Control 1_1	4	Day 7_5	7
Control_1_2	4	Day 14_1	11
Control_1_3	3	Day 14_2	10
Control_1_4	4	Day 14_3	6
Control_1_5	3	Day 14_4	5
Control_1_6	3	Day 14_5	9
Control_2_1	8	Day 14_6	10
Control_2_2	7	Day 60_1 1	7
Control_2_3	6	Day 60_1 2	2
Day_5_1	8	Day 60_1 3	4
Day_5_2	6	Day 60_1 4	5
Day_5_3	8	Day 60_1 5	3
Day_5_4	7	Day_60_1 6	5
Day_5_5	11	Day 60_2 1	10
Day_5_6	8	Day 60_2 2	8
Day_7_1	7	Day 60_2 3	4
Day_7_2	7	Day 60_2 4	6
Day_7_3	4	Day_60_2 5	4
Day_7_4	8	Total:	232

Table 15.3: Number of putative binding site per mice: In this table has been reported the number of central or putative binding site for each mouse-repertoires, as we can see this is the table with higher number of clusters and variation. Columns description as Table 15.1.

We can see that the number of clusters found in this experiment (Table 15.3) is 232, and this value is much higher than in the other experiments: 95 (Table 15.1) and 88 (Table 15.2). This result suggests that this central part of the CDR3 is the bigger source of variability for the CDR3 sequences.

Notably, the number of clusters is slightly proportional to the number of unique sequences of mice, as we can see in Figure 15.4.

Unsurprisingly, a higher variability of the repertoires and the consequence number of resulting clusters correspond to an increasingly higher number of unique sequences.

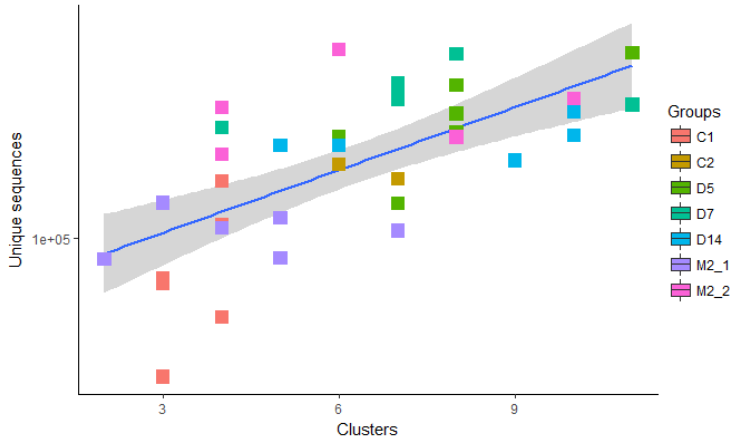


Figure 15.4: Number of clusters vs. Unique sequences This plot demonstrates the relation between the number of clusters found by Hammock in Table 15.3, and the number of unique sequences for that sub-string of CDR3. Unsurprisingly, a higher number of unique CDR3 sequence is correlated to a higher number of clusters found by Hammock. The line in blue is the linear regression of the data. Groups legend: C1 = Sequence from Control mice from the first experiment; C2=Control mice second experiment; D5= Day 5; D7= Day 7; D14= Day 14; M2_1= Two months, first experiment; M2_2= Two months, second experiment

15.5 Cluster class by class

Since I can now identify internal motifs on the CDR3 sequences, I am also able to investigate whether there are unique and distinct motifs in the mice repertoires infected with different antigens, and assess if there are unique and distinct motifs in the mice repertoires infected with different antigens. This is due to the possibility that, if our genetically identical mice are all infected with the same antigens, we might see a similarity in the motifs produced as a reaction to the common stimuli and, possibly, a higher difference between the control motifs vs. the infected mice ones.

The results are reported in Table 15.4:

Repertoires	Clusters
OVA	12
CFA	7
Control	9

Table 15.4: Number of results for all mice: Number of cluster resulting from Hammock for the mice infected by different antigens and the control mice. On the first column is present the list of mice divided per antigen, the second presents the number of clusters.

We can see that the number of cluster motifs is inferior to the 13 motifs present in all mice. The CFA group have a smaller number of motifs —only 7 —while the OVA mice have almost double the number of clusters.

In the following section we report the results separately for each mice group.

15.5.1 All OVAs

Figure 15.5 shows the results from Hammock of all sequences originated by OVA infected mice.

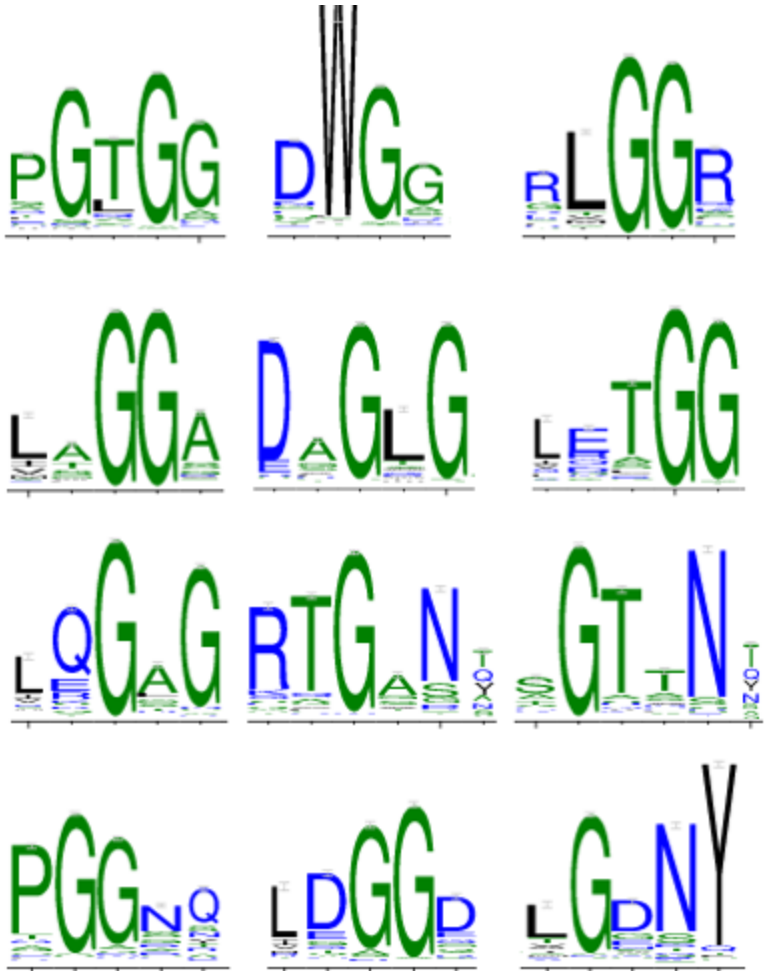


Figure 15.5: Results for Hammock for OVA mice: Results for Hammock for OVA mice. From left to right, top to bottom the picture has been create with the following number of sequences (rows separated by semicolon): 30,114, 50,212, 12,935; 7,398, 9,387, 9,523; 5,712, 5,709, 4,104; 10,711, 4,820, 5,664.

15.5.2 All CFAs

In Figure 15.6, we see the results from Hammock of all sequences originated by CFA-infected mice.

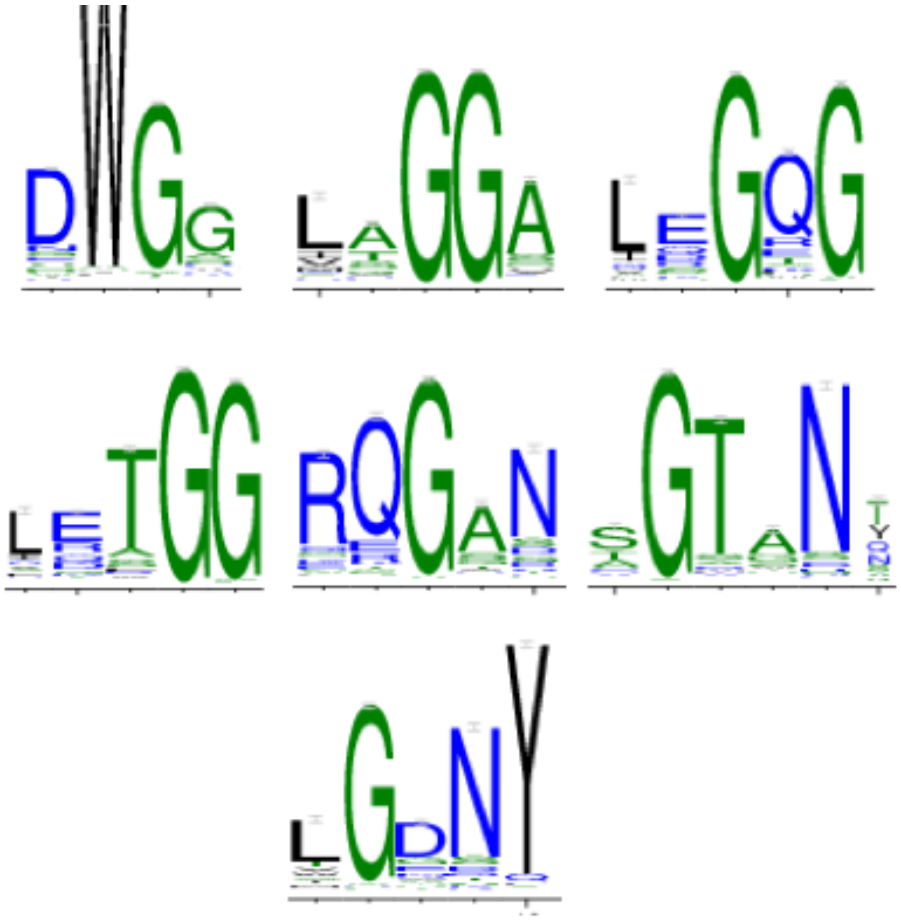


Figure 15.6: Results for Hammock for CFA mice: From left to right, top to bottom the picture has been create with the following number of sequences (rows separated by semicolon): 50,204, 6,522, 8,159; 8,999, 6,183, 4,644; 5,925

15.5.3 All Controls

In Figure 15.7, we see the results from Hammock of all sequences originated by control mice:

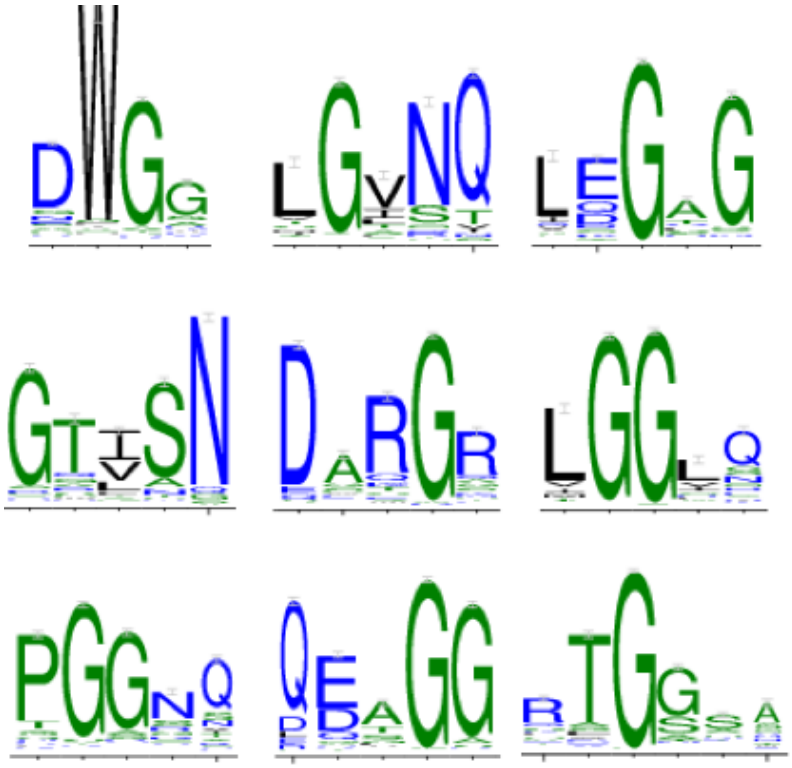


Figure 15.7: Results of Hammock for control mice: From left to right, top to bottom the picture has been create with the following number of sequences (rows separated by semicolon): 17,479, 2,197, 2,811; 2,119, 2,572, 5,041; 2,296, 3,609, 5,086

15.5.4 All sequences combined

In this run, the length of all CDR3 is much shorter than usual; as such, there is a higher number of duplicated sequences, and the unique sequences are therefore much lower than in all other runs. This meant I could use all the sequences in a single Hammock run, obtaining a single result for the entire data set.

The results are the following thirteen different clusters, presented in Figure 15.8.

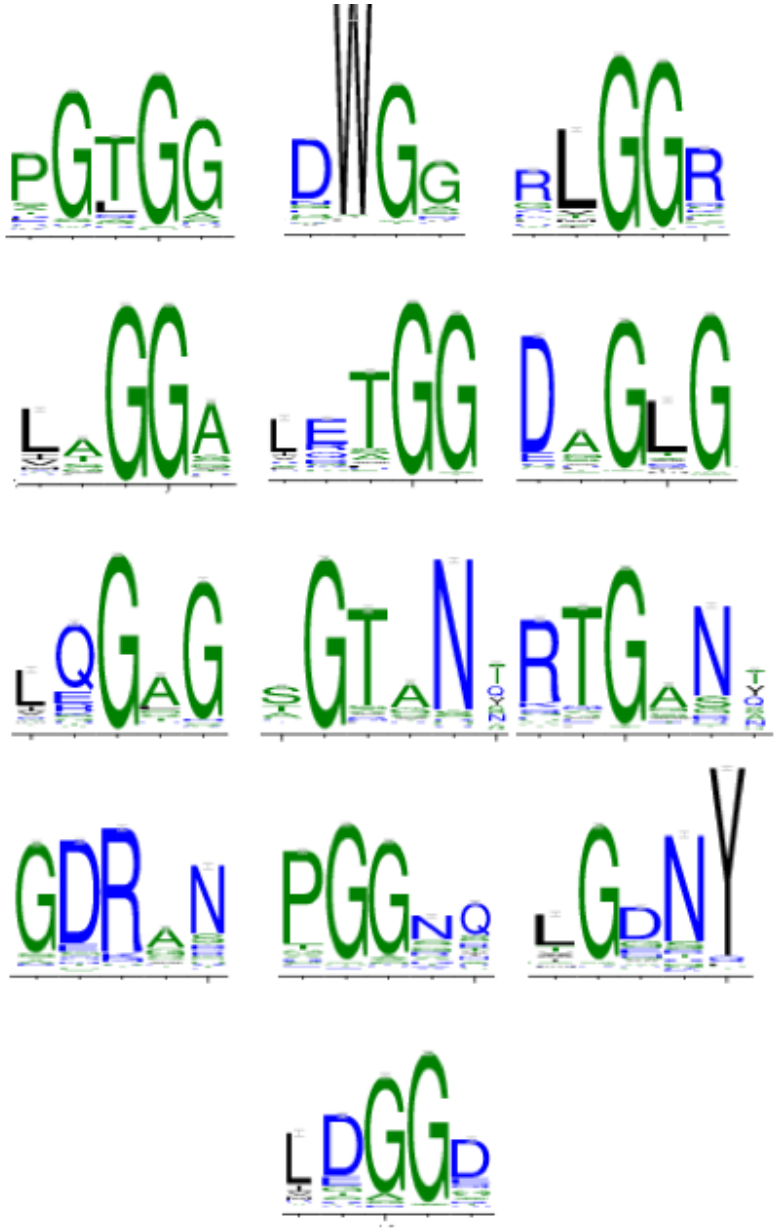


Figure 15.8: The putative binding site of the CDR3: Reported here are the thirteen different results of Hammock applied on all sequences of the dataset of repertoires. The motifs are in the centre of the interaction between the TCR and the peptide + MHC, and twelve of them have the common presence of one or more G in various combinations. A presence of numerous glycine is already being found in the literature [12][50], and to have found it again here could hint to a relevance of this kind of amino acids for the CDR3 and peptide interaction. Legend: From left to right, from top to bottom: the picture has been created with the following number of sequences (rows separated by semicolon): 43,376, 75,258, 19,015; 10,730, 13,823, 14,064; 8,282, 6,262, 8,406; 6,400, 15,433, 8,356; 6,946.

These thirteen motifs have some rather interesting characteristics: they are in the centre of the interaction between the TCR and the pMHC and, because we have already eliminated the V and J region tails, what remains is the D region of the CDR3. Considering what we saw in the introduction (sections 3.8 and 3.9), we can probably consider this section of the CDR3 as the primary centre of interaction with the peptide, and of primary importance to this research.

Since all mice binding site sequences are clustered into thirteen different groups, a few motifs are in evidence, with a limited variability.

Indeed, all thirteen motifs are a combination of the following eleven amino acids: A, D, G, L, N, P, R, T, W, Y, with an over-representation of G.

15.6 Discussion

15.6.1 The putative Binding Site

With the use of Hammock, I divided the repertoires into a small number of high-similarity-sequence clusters. Looking at the motifs resulting from the clusters, we can spot the presence of three different levels of emerging conserved-areas in the CDR3 sequences: the C, ... , FG[X]G border motif, the V and J tails, and the core of the sequence, the D region. These three appear to be a clear division of the CDR3 β chains, and any future research should consider this division and analyse them separately. Indeed, an important piece of future research would be to extend this analysis to the CDR3 α chain and observe the emergent patterns.

Furthermore, the D region is to be considered one of the most important regions of the CDR3. There are indications that this area is of primary importance for the interaction with the peptide carried by the MHC, and probably to be considered the binding site of the peptide. As we have seen in the introduction section 3.8 and in Figure 3.5, this is the area conserved according to the experiment reported in [47], and which other publications like [57] are pointing out as relevant.

With the use of Hammock, we have seen the presence of thirteen different motifs in Figure 15.5: these 13 motifs are made of a combination of eleven different amino acids rotating around the presence of one or more glycine, as also similarly

observed in [174].

In the following Table 15.5, I have reported these eleven amino acids, listing their names, side chain polarity, and charge.

Amino acid	Hydrophobicity	Side chain polarity	Side chain charge
A	1.8	Non-Polar	Neutral
D	-3.5	Acid Polar	Negative
G	-0.4	Non-Polar	Neutral
L	3.8	Non-Polar	Neutral
N	-3.5	Polar	Neutral
P	-1.6	Non-Polar	Neutral
Q	-3.5	Polar	Positive
R	-4.5	Basic Polar	Neutral
T	-0.7	Polar	Neutral
W	-0.9	Polar	Neutral
Y	-1.3	Polar	Neutral

Table 15.5: Summary table of conserved amino acids: In this table is present a summary information on the eleven most conserved amino acids present in the thirteen motifs (Figure 15.8). I obtained this list by looking at the multiple sequences alignment files of the thirteen motifs, I have computed the frequency of each amino acids per position and filtered out the amino acids with frequency lower than 50%. The result is the following eleven amino acids: G, present in thirteen motifs, A in 3, D in 5, L in 4), N in 4, P in 2, Q in 1, R in 3, T in 4, W in 1, Y 1. The table also reports their amino acids' hydrophobicity scale (source [175]) side chain polarity,and side chain charge (source [176]). The lower is the value in hydrophobicity scale the more the amino acids is hydrophilic.

As we can see, the side chain polarities of these eleven amino acids are equally distributed as either polar or non-polar. With the exception of A and L, they all have a hydrophobic side chain, and nine out of eleven have a neutral side chain charge. This implies that all thirteen motifs generated by these amino acids have a largely neutral (therefore not charged) environment, suitable for protein-protein interaction.

These properties are very important, not only during the formation of the protein structure, but also for the interaction between different proteins. For example, amino acids with charged side-chain can form ionic bonds with other charged molecules [177]. In this case, the only charged amino acids are D (aspartic acid) charged negatively and R (Arginine) charged positively. As we can see in Figure 15.8, one of the two is present in six different profiles, giving the neutral charged

profile either a more positive or more negative charge. Only in one profile (Figure 15.8 bottom-left) they are together, and in one they are next to each other.

Looking at the side chain polarity of these amino acids, seven are polar and four non-polar. Polar amino acids have the tendency to be on the outside of a protein, where they can form hydrogen bonds with water and other polar molecules [178]. In addition, almost all amino acids have a hydrophilic behaviour common to surface amino acids. These data suggest that an interaction between CDR3 and the peptide is driven primarily by hydrogen bonds formed by the polarity of the side chain. It suggests, too, that this occurs in a mostly neutral environment with a minor component of positive or negative charged amino acids.

Among all thirteen profiles, the most frequently present amino acid is G (glycine) as it is also often present around binding sites [179]. As we know from the literature [50][174], this is the smallest amino acid with only one hydrogen as a side chain: it can rotate easily, and it is used to add flexibility to the protein chain. However, such flexibility is usually not desirable as a structural component for molecular conformation. Indeed, G and P (proline) are referred to as "helix breakers" because they break the regularity of the α helical conformation. This would likely have the direct effect of helping to keep the CDR3 loop a structurally disordered region.

We have already seen the presence of a glycine "rich region" of the CDR3 in the Introduction, section 3.8, and Figure 3.6. We observed that this presence can be explained either by recombinational biases of the V(D)J recombination, or by the convergent recombination of the amino acids; in other words, the redundancy of the nucleotide triplets.

These two theories are not mutually exclusive, moreover I would add that this phenomenon's presence can be explained by functional reasons. Indeed, thanks to the glycine its small neutral side chain and the presence of other neutral amino acids, a neutral a water-soluble area is created —suitable for the interaction with the peptide [179].

This result seems to recall my findings with the work on the 1-DBF [17]. There, I selected twelve triplets that have been considered highly informative and

likely relevant to the interaction with the peptide. The major part of those was found around the fifth amino acid, and three of them have at least one G present: GQG, GVG, TGV; very similar to the G_G and GT motifs found by Hammock.

15.6.2 The motifs in Controls, OVAs and CFAs

When I applied the Hammock program to the D region for the repertoires divided by group Control, OVAs and CFAs, I found nine, twelve, and seven clusters respectively. Despite the groups being different in a number of sequence and repertoires, the resulting motifs are not completely different when dealing with groups being immunised with different antigens, or being simply a control. In fact, there are similar motifs across all.

15.6.2.1 Assessing the similarity of the clusters

To have a better idea of the similarity and difference of my motifs, I needed to develop a method that would provide me with an estimation of the 'closeness' of the motifs. I chose the program HH-suite. This program is used to make database searches of a set of sequences by converting the sequences into Profile HMMs and searching the databases of HMMs. It is normally used to search in HMM database like Pfam.

One way the program works is to assign a value of similarity between the query and the different profiles of HMM in the database. Therefore, I converted my motifs into multiple sequences alignment using ClustalΩ. This provided a suitable output format to turn them into Profile HMMs using HH-suite and, measure the distance between one motif/HMM to all the rest.

I applied this method to two different set of motifs: those from Control, OVAs, and CFAs from Figure 15.8, and those from Table 15.3. Using the matrix of distance generated by HH-suite between all motifs, I could create the heat map [180] presented in Figure 15.9 and Figure 15.10.

In Figure 15.9, using the heat map and the trees on the side, I investigated the possibility that the different motifs originated from the different mice groups could have produced groups of motifs that resembled the different origins. Using

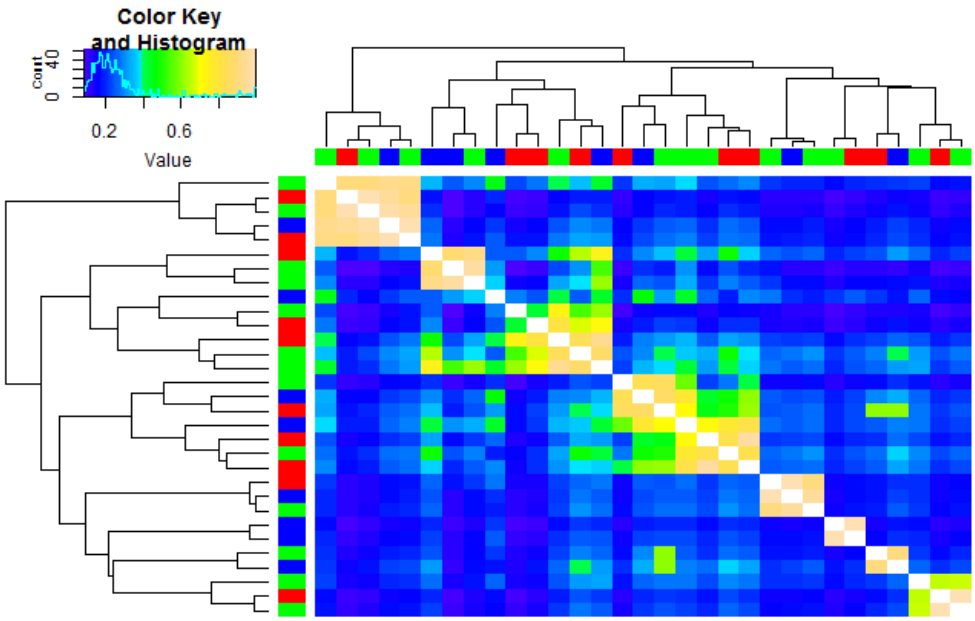


Figure 15.9: Heat map of the motifs from Control: OVA and CFA groups. On the margins, the control motifs are labelled in green, the CFA in blue, and the OVA in red. Using the label colours, we can see that there is not a clear division of motifs, and that those produced by a group are like the others. In other words, there is no clear division between the groups. In the heat-map, the value of similarity given by HH-suite is expressed by a value between 0 (blue) and 1 (white).

the colour label system (green for control motifs, blue for CFA and red for OVA), it is possible to observe that this does not happen and that, in fact, the motifs produced are similar to each other.

A similar result is obtained for the heat map with 232 motifs, Figure 15.10.

Indeed, the results for 232 motifs shows that there is a “gathering” of similar motifs from the mice immunised with the same antigen.

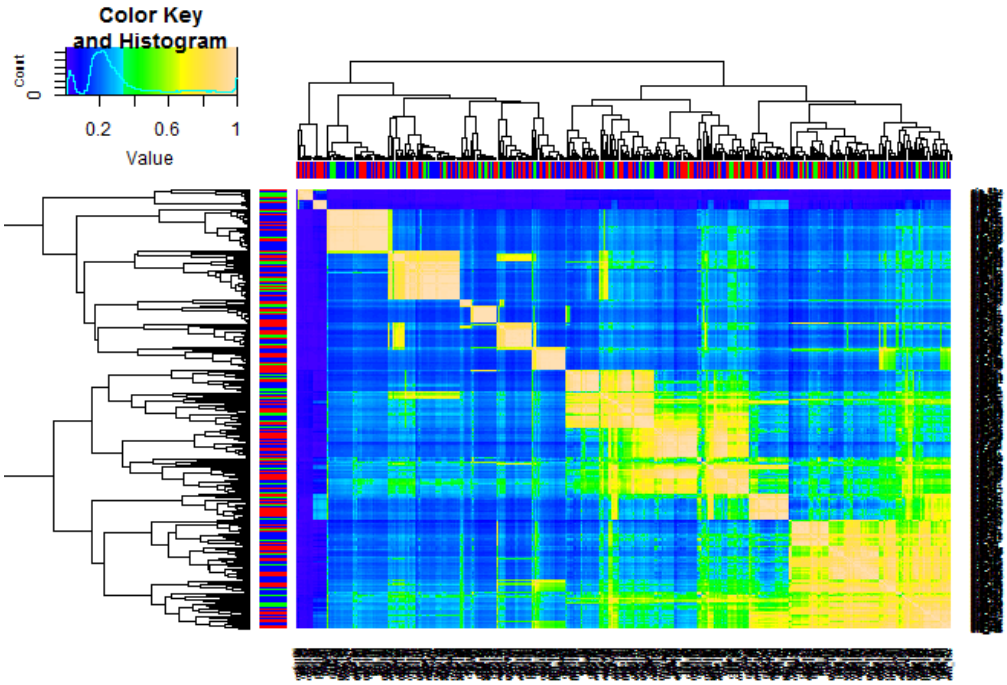


Figure 15.10: Heat map for all motifs: Heat map of the motifs resulting from all mice analysed by Hammock, one by one. On the margins the control motifs are labelled in green, the CFA in blue, and the OVA in red. Additionally, in this heatmap, we can see that there is not a clear division of motifs, and that those produced by one group are like the others. In the heat-map the value of similarity given by HH-suite is expressed by a value between 0 (blue) and 1 (white).

Apparently, the idea to use Profile HMM and the HH-suite program was a logical progression of prior research. Indeed, a different laboratory published a new program named pHMM-Tree [181] that creates trees of Profile HMM. The concept and tools of this experiment were very similar to mine, although they created a wider and more structured system, with different options in input or output. Using my old Profile HMM, I ran pHMM-Tree program and created Figure 15.11.

The trees from Figure 15.10 and Figure 15.11 are different; however, both show the absence of a clear difference between the motifs originated by mice immunised with the same antigen.

With these experiments, I hoped to find the different reactions from the mice with common immunisation, and to find motifs that would be simultaneously different between the mice groups and similar among themselves. In other words, I hoped to see a plot in which OVA and CFA motifs are gathered in two distinct groups. However, this was not the case; OVA and CFA motifs do not appear to be intrinsically different. Although there motifs are present in some mice groups and not in others, this is not sufficient to establish a clear trend. The reason for this is not yet clear, and more research is required.

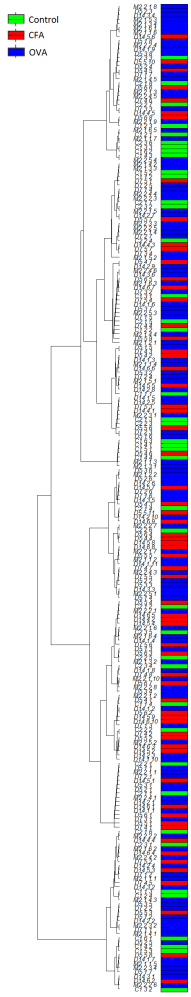


Figure 15.11: pHMM-tree result using the 232 motifs: Dendrogram of the 232 motifs resulting from Hammock in which similarity has been computed using pHMM-Tree program. On the right margins, the control motifs are labelled in green, the CFA in blue, and the OVA in red. With this plot I am testing the hypothesis whether repertoires immunised with same antigens would produce similar results. In the previous two pictures I used a method of my own making using a combination of HH-suite and HMMer (see Figure 15.9) and here I repeated the test using a newly published method pHMM-Tree [181]. Even in this case, we cannot detect large groups of similar motifs coming from similar mice.

Chapter 16

Hammock Results as features of a SVM classification test

16.1 Experiment Description

Here, I describe my work on a new strategy for the classification of CDR3 repertoires, using a combination of HMM and SVM classification methods.

In the experiments with the SVM, I encountered the presence of duplets and triplets in all repertories and used the resulting vectors for the training/test of the SVM. Here, instead, I am using the Profile HMM originated by the Hammock motifs as features of the SVM.

Using a program like HMMer, it is possible to convert any given multiple sequences alignment into a Profile HMM. And with HH-suite, it is possible to evaluate how a new sequence is closely related to a Profile HMM. Each time I test a sequence against a Profile HMM, I obtain a score value; if the score value is higher than a determinate threshold, the new sequences could be considered a member of the Profile HMM. For example, if I have a Profile HMM trained by a set of immunoglobulin sequences, and I want to test if a new unknown sequence is an immunoglobulin or not, I would test it against the Profile HMM. If the resulting score value is high enough, the sequences will be considered part of the "family" sequence.

In the same way, if I have a sequence or set of sequences from an OVA mouse, and I test it against a Profile HMM, and it is considered a member of the HMM, so I

would expect that it originated from other OVA mice. Let us see this in more detail through the workflow of this experiment:

1. I selected five sets of random 50,000 sequences from a mouse repertoire, for example, OVA 1. I use these five sets as my test set.

2. I merged all remining OVA mice in one large group, leaving out, of course, our test mouse. Then I divided the groups into two halves.

   (a) From the first half, I selected five sets of 50,000 random sequences. This defines my training set.

   (b) I run Hammock with the second half, obtaining a group of motifs. These motifs will be turned into multiple sequence alignments using ClustalΩ and into Profile HMMs using HMMer.

3. All remaining CFA mice were merged together and run with Hammock, and its results turned into Profile HMMs.

4. The features of the SVM will now be formed by all profiles originating the CFA mice (point 3) and the second half of OVA sequences (point 2b).

5. The first half of all OVA (point 2a) was then used as training test for all profile-HMM/SVM-feature (point 4).

6. This workflow results into five numerical vectors for the test set and five for the training, and I could at this point perform the SVM.

Every time a new mouse was used as a test set, I needed to run a new Hammock run, resulting in a different number of Profile HMMs. A report of these changes is presented in Table 16.1.

While the other parts of the features formed by the motifs for all OVA and all CFA mice stay the same, I already have the Hammock results from the analysis shown in the previous chapter: twelve motifs for OVA mice, and seven for CFA.

When a CDR3 sequence is tested or trained against the Profile HMM list, I cannot be sure that such sequence is considered belonging to only one Profile HMM;

Test Mouse	All w/o test	Test Mouse	All w/o test
CFA 1	8	OVA 2	11
CFA 2	4	OVA 3	8
CFA 3	5	OVA 4	8
CFA 4	6	OVA 5	10
CFA 5	5	OVA 6	11
CFA 6	9	OVA 7	11
CFA 7	6	OVA 8	11
CFA 8	5	OVA 9	6
CFA 9	9	OVA 10	10
CFA 10	8	OVA 11	4
CFA 11	5	OVA 12	6
CFA 12	5	OVA 13	4
CFA 13	7	OVA 14	10
OVA 1	10	OVA 15	10

Table 16.1: Number of clusters per groups: In this table are reported the number of hammock clusters formed by all mice immunised with the same antigen, less the test mouse. Every time, I test a mouse repertoire for an SVM with profile-HMM as feature, I need to re-run Hammock. As a consequence, the number of Profile HMM for that class of mice, in this case OVA or CFA, changes.

however, my assumption is that a higher number of sequences from one particular test class will be considered part of that trained class. This majority of sequences would drive the SVM to classify the repertoire correctly.

In Table 16.2 the results of the experiment are reported:

Mice	Classified	Mice	Classified
OVA 1	0	OVA 15	1
OVA 2	0	CFA 1	0
OVA 3	1	CFA 2	1
OVA 4	1	CFA 3	0
OVA 5	1	CFA 4	1
OVA 6	1	CFA 5	0
OVA 7	1	CFA 6	0
OVA 8	1	CFA 7	1
OVA 9	0	CFA 8	0
OVA 10	0	CFA 9	1
OVA 11	1	CFA 10	0
OVA 12	1	CFA 11	0
OVA 13	1	CFA 12	1
OVA 14	1	CFA 13	0
			57%

Table 16.2: Results for the SVM+HMM classification experiment: Here are listed all 28 immunised mice, with their classification result (1 if correctly classified, 0 if not classified correctly). The assignment is given by a majority of test vectors assigned (3 out of 5). The result of 57% is not high, and indeed, only slightly more successful than random classification. In actual fact, we could obtain a value of 53.6% by classifying all as OVA.

16.2 Discussion

With this experiment, I aimed to merge two important and well-known machine learning techniques and create another use for the motifs found by Hammock (in addition to creation of the heat-map, section 15.6.2).

Unfortunately, as for the case seen in Figure 15.9, Figure 15.10 and Figure 15.11, the motifs found in the mice repertoires are too similar to each other. As such, this did not allow me to have a clear classification of the test/train set of sequences, nor distinct numerical vectors for the SVM.

This method combines a great number of programs and techniques: Hammock for the motifs; ClustalΩ for the multiple alignments; HMMer for Profile HMM creation; HH-suite for test sequences vs. HMM; and SVM for the classification.

Despite the result of this experiment being low and not promising, I cannot say that this workflow is totally erroneous. It is likely that, with a different set of HMM families formed by longer sequences or better-defined patterns, it would bring more successful results.

Part III

Conclusions

Chapter 17

Conclusions

In this thesis, I have reported my research on the use of machine-learning methods for the classification of the CDR3 sequences, the analysis of repertoires, and the interpretation of the results, given the current literature. This doctorate project falls under the overall aim of Professor Chain's lab of understanding the functioning of T-cell immunity by developing computational methods for the analysis of the TCR repertoire.

As I demonstrate in the Results section of this thesis, the T-cell receptor plays a crucial role in the immune system: comprehending its mechanisms is vital for the progress of research in immunology. There are still many unanswered questions regarding the TCR, especially regarding its interaction with peptides carried by MHC. Although, on one hand, the interaction between peptides and MHC is well known [18][19][20] and there are many computational models with which to predict it [182][183], on the other, there are few papers on TCR and peptides interaction and none at all on the model proposed within this study.

17.1 Current Challenges in the TCR repertoires studies

A number of major challenges remain in interpreting T-cell receptor repertoire data. As we have seen, the protein is composed of two distinct chains (α and β). These are paired in the structure; but most methods for sequencing large numbers of TCRs do not retain the pairing information (pairing problem). Researchers are trying to

avoid this problem through single cell sequencing [184][185].

Furthermore, β chains are historically the main target of studies because of their higher variability (due to the D region) [186][187]. Therefore, the number of α chains in the literature is lower.

Another major issue in the study of TCR is the absence of research about the large sets of crystal structures of the TCR, bound to MHC/ peptide. X-ray crystallography is understandably difficult, and to be able to capture the two molecules (TCR and MHC) and the peptide during the protein interaction results in few ternary (MHC-peptide-TCR) complexes.

All these issues have a bearing on the study of the CDR3, the little patch of TCR which is considered to play the major role in the interaction concerning CDR3 sequences.

Aware of these limitations, I conducted my doctoral research by proceeding from the most common and basic analytical techniques, and gradually increasing the complexity of the methods. In doing so, I began by using the methods common to existing literature —such as statistical and quantitative analysis of the repertoires —and subsequently extended these to more recent, advanced machine learning techniques.

17.2 The Data-Set: CDR3 numbers, sharing and diversity

This thesis is based entirely upon a set of TCR sequences obtained from our collaborators (Friedman Laboratory at the Weizmann Institute). The data were obtained when repertoire sequencing by high-throughput sequencing was in its infancy, and the library preparation and sequence quality have since been improved. Nevertheless, the data set is an interesting starting point for an analysis of TCR repertoire and its relationship to specificity [16][17][69].

In chapter 4 I have reported the statistical and quantitative analysis of the 37 murine β-chain CDR3 repertoires used in the thesis. The collection of TCR β chains from splenic $CD4^+$ T cells resulted in a large difference in the number of

sequences per repertoire, ranging from a few hundreds of thousands in the Control mice to several million in the Day 7 mice, (see section 3.6).

This discrepancy in numbers is reflected to a lesser extent in the number of unique sequences (section 4.3). The standard deviation for the whole database passes from $2.6 \cdot 10^6$ for all sequences to $1.4 \cdot 10^5$ for all unique sequences. Interesting to be noted is that the number of unique sequences seems to reach a plateau when at least a million CDR3 sequences are recorded. I posit that we can consider one million sequences as the minimum number of CDR3 required to have an acceptable representation of the internal variability of the whole CDR3 repertoire within the body.

Another confirmation of previous studies is the average length of the CDR3. If we consider the C and FG[X]G motif present at the beginning and end of the sequence, the total length is 16 amino acids (see Figure 4.7) with a variance of 3.55 amino acids. Although the median of sequences is formed by 17 amino acids, the high number of very short sequences pushes the mean to a lower value.

From the analysis of the repertoire, we have also seen two other important indexes: the Jaccard Index and the Gini coefficient.

With the former, we can see that —despite all the mice being genetically identical —the type of CDR3 produced is substantially different and, even if immunised with the same antigen, the immune responses produced are different. The very few sequences shared confirms that the immunisation event alters the repertoires, but it does not drive repertoire convergence [16]. This result proves that identical stimuli would not necessarily produce similar responses, and that looking at the whole CDR3 sequence to search for the immune response strategy is not the right move. Indeed, the number of shared and private sequences is not dissimilar within a small number of genetically identical mice or a large population of unrelated individuals.

Finally, with the Gini coefficient, we saw that the event of immunisation does not push the repertoire to an over-representation of one or a few sequences, but that immunised repertoires continue to be extremely diverse and contain a high number of different CDR3 sequences [188].

17.3 The significance of short protein motifs in repertoire classification

After the quantitative analysis of the repertoires, I considered the application of different bioinformatics tools and methods. At the beginning, I started using the common tools of sequence analysis (such as using alignment models like ClusterΩ to create a consensus of the CDR3) to plot sequence logos and try to create profiles of sequences in order to highlight families of sequences and produce phylogenetic trees.

However, I realised quickly that these methods were unfeasible for these kinds of protein sequences. The length of sequences is too short, with too much variation: these methods would be unable to produce multiple sequence alignment with any meaning. Even if we were using sequences all of equal length, like all sequences of 16 amino acids, the results would be still unusable for identifying variations in the different kinds of CDR3 repertoires.

Other bioinformatics methods for protein structural analysis, like building by homology, are not enforceable because we do not have the information for the α chain. The solution we adopted —and the main topic of this thesis —is to utilise the methods of classification to explore the property of the repertoires. If we were able to create a valid prediction tool for the classification of the repertoires, we could verify whether the same features used by the tool also have a biological relevance. This concept was previously adopted by the lab and published in [16]. Therefore, I reviewed this initial work and extended the method to a new set of repertoires just sequenced, which I called Group B (see Section 4.2).

17.4 Experiments

17.4.1 Repertoire classification using SVM

My work on [16] is reported in chapters 5, 6, 7 and 8, including an introduction to the methods, my review of the workflow, and the results.

I analysed the work done in [16], using some of the core ideas behind it. In

this study, thanks to the use of numerical factors and k-means algorithm mean it has been possible to decrease the complexity of the data and reduce the very large number of CDR3s in each repertoire to a smaller number of numerical features. It was possible to convert the string of amino acids into actual numerical vectors and, with the Support Vector Machine, to have a high value of success rate.

I explored several of the method's parameters. I compared the results of converting amino acids to numerical vectors, using Atchley factors to capture the physical/chemical properties of the amino acids along with two other related sets of such factors (Sandberg, Kidera). In order to test the importance of this conversion, I also used an arbitrary conversion which yielded a similar set of numerical values, but which did not contain any implicitly biological meaning. Using these different feature sets, I tested the results of the k-means using different numbers of clusters, extended the bag of words using duplets and other p-tuples not reported in this thesis, and tried different SVM kernels.

Unexpectedly, it became apparent that the simple counting of p-tuples in repertoires (for example, the frequency of the 8,000 possible triplets) was sufficient to give good classification performance using the SVM, without any prior feature reduction or clustering steps. Thus, clustering of amino acids according to chemical or physical properties did not seem to be important for repertoire classification.

With the workflow of [16], I carried out the classification of the repertoire following two main aims: the classification of repertoires based on their sacrifice date, and classification based on the antigen with which they were immunised, repeating the experiment for Group A (data from the first experiment) and Group A and B combined (all data). All results are reported in section 8.1.

For the classification based on the sacrifice day, the results are relatively high for duplets and triplets, with values between 70% and 81% (check Table 8.9 and Table 8.10). These values are high but they suffer from a few misclassifications, and more if compared with [16]. This is, due to different subsampling of the data, the use of default in-function SVM parameters used during the process of SVM training. This does not, however, diminish the conclusions that can be drawn. Firstly, that the

immune response is carried by a large numbers of low frequency CDR3 with shared features (p-tuples); and secondly that, in addition, there is not a dominant clonotype and both high and low frequency p-tuples contribute to the immune response.

Events of immunisation can profoundly change the repertoires, such that to distinguish and to classify control repertoires versus immunised mice is the easiest of all classification. More difficult is the classification of immunised mice, and this suggests that each individual generates a seemingly unique response. However, with the progressive advance of the classification method, I believe that we will be able to use these low-level features in approaches for the analysis of clinical samples that will be more and more reliable.

For classification of repertoires based on antigen stimulation, i.e. the ability to distinguish between CFA+OVA and CFA which were not reported in the paper [16], the OSR is not high, with an average success rate of 64%. This is slightly higher than random classification, but not enough to pursue in the application of this method, for the classification of different antigen-immunised repertoires. It is hard to say why one type of classification works better than the other, and this is probably due to various different factors. One is probably the fact that control mice are easily correctly classifiable, while in OVA vs. CFA classifications, Control mice are absent.

Also, especially regarding the classification of OVA vs. CFA, the mice repertoires have been collected in a non-optimal time frame; in C57BL/6 mice, the $CD4^+$ T cells have a peak of response around 9 days, after the two first collection points (day 5 and 7), followed after a first half-life phase after 3 days, therefore before the third time point (day 14) and, a second half-life phase after 35 days, way before the fourth time point (day 60)[189].

17.4.2 1-Dimesion Bayesian Function

At this point of the study, my focus was to find a different method to classify the OVA vs. CFA. At that time, I noticed that if I decreased the number of p-tuples (all duplets, triplets etc.) used for the SVM classification, the OSR could increase dramatically.

Intrigued by this idea, I decided to see if it was possible to select a valid subset of features; that is, a not arbitrary subset of features, that could increase the success rate, and might be later considered as having biological relevance.

I started by selecting a p-tuple at the time and iteratively trying to classify it as belonging to OVAs or CFAs repertoires. Then, sorting them from the highest to lowest classification rate, and using a progressively larger subset of features, until I had the best CDR3 success rate. Indeed, if a single feature can correctly classify itself, perhaps, a small collection of features with the same characteristic would lead to greater OSR in the overall CDR3 classification. After a few attempts, I applied the formula of the 1-Dimensional Bayesian Function (1-DBF). This function can evaluate the likelihood of an element being part of one of two classes, using the mean and standard deviation of the samples. This process worked well, and I could boost the success rate from the OVA vs. CFA, from 61% to 94% using only twelve triplets. This work was later published in [17].

The principal part of this work was to use an initial step to select a limited number of relevant features, and then to use these in a non-linear SVM thus reducing the noise/over-fitting but retaining the flexibility of the non-linear kernels. An alternative approach to achieve this end was explored by another PhD student in our group [69] —this was using linear boosting and 1-norm optimisation for automatic feature selection.

In [17], I combined two different machine learning methods —Bayes and SVM —using the features selected from the former as input into the latter. Therefore, I had to use two different test sets, an inner and an outer test, to avoid any danger that success rate derived from overfitting. Indeed, in my first iteration of this workflow, I did not consider this eventuality, and all my result were 100% accurate. This was clearly a methodological mistake. As previously mentioned, this kind of approach has been repeated in different studies in recent years [125]-[132]. This suggests that this method holds potential, and I would like to extend the research to other datasets.

With this classification system, I isolated 12 triplets that seemed to have a real

biological meaning and relevance in terms of amino acid positions and composition. Interestingly, some of the triplet expressing combinations of glycine will return in my work with Hidden Markov Models.

17.4.3 HMM based analysis and classification

In the last part of my thesis, I focused upon the use of HMM-based programs for the classification and analysis of my repertoires. I sought a way to apply Markov Chain and HMM methods for the majority of my doctorate. I knew that HMMs are one of the most used machine learning techniques for sequence analysis [150], and I tried different approaches to find families of sequences within the database, and try to classify the repertoires, but with scarce results, which have not been reported here.

With the publication of the Hammock program [149] I could identify the three areas of CDR3 (reported in chapter 15), to identify protein motifs for each group of mice, to create phylogenetic trees of the motifs (a similar concept is published in [181]), and I was able to use and combine many programs and techniques available in the literature. Although, with these methods, I could not produce valid classification methods (chapter 16), I think this is the part of my research in which we have the best and cleanest results in terms of visualisation and analysis of the entire CDR3 repertories.

First, we had a clear view of the CDR3 repertoires in a single plot. We could see and identify three different parts of the CDR3: *1)* the borders of the sequences (C, ... , FG[X]G), that confirms what was already reported in the literature; *2)* the presence of a V and J "tails" within the CDR3; *3)* the presence of a numerous combination of glycine forming motifs and patterns that, until this point, had not been considered.

17.5 Future Work

It would be worth repeating the analysis with the 1-DBF with new data and comparing their results. I am curious to see whether the resulting motifs found on the D region of the CDR3 can change, and how much, with respect to different repertoire.

In doing so, I would hope to conclude if the motifs found are unique or shared.

I think that a deeper analysis of the results of Hammock —concerning motif presence and composition —would be important and highly relevant for a future study. I conducted a small research project on this during the last period of this study (section 15.4), but this should be extended.

Furthermore, obtaining repertoires of α and β-chain CDR3s combined would be a fruitful line of enquiry. Indeed, I am convinced that the study of motifs and patterns within the CDR3 repertoires is the path for this field to follow, as other studies are pursuing the same goal [190].

Bibliography

Bibliography

[1] N Hall. Advanced sequencing technologies and their wider impact in microbiology. *Journal of Experimental Biology*, 210(9):1518–1525, 2007.

[2] David Sadava, David M Hillis, H Craig Heller, and May Berenbaum. *Life: the science of biology*, volume 3. Macmillan, 2009.

[3] KA Wetterstrand. DNA Sequencing Costs: Data from the NHGRI Genome Sequencing Program (GSP), Available at `https://www.genome.gov/27541954/dna-sequencing-costs-data/`.

[4] Andreas Lossius, Jorunn N. Johansen, Frode Vartdal, T Holmy, and Trygve Holmøy. High-throughput sequencing of immune repertoires in multiple sclerosis. *Annals of Clinical and Translational Neurology*, 3(4):295–306, Apr 2016.

[5] Jorg J A Calis, Brad R Rosenberg, C Whitehead Presidential, and Fellows Program. Characterizing immune repertoires by high throughput sequencing: strategies and applications. *Trends in Immunology*, 35(12):581–590, Oct 2014.

[6] Harlan S Robins, Paulo V Campregher, Santosh K Srivastava, Abigail Wacher, Cameron J Turtle, Orsalem Kahsai, Stanley R Riddell, Edus H Warren, and Christopher S Carlson. Comprehensive assessment of T-cell receptor β-chain diversity in $\alpha\beta$ T cells. *Blood*, 114(19):4099–4107, Nov 2009.

[7] Paul Klarenbeek, Paul Tak, Marieke Doorenspleet, Barbera van Schaik, Felix Wensveen, L Gottschal, Marja Jakobs, Ingrid Derks, E Eldering, Antoine

Kampen, Frank Baas, and Niek Vries. High throughput sequencing (HTS) provides full repertoire analysis of the B and T cell receptors in humans and mice, both in blood and synovial tissue. *Ann Rheum Dis*, 69(33):33, 2010.

[8] Grzegorz A. Rempala and Michałl Seweryn. Methods for diversity and overlap analysis in T-cell receptor populations. *Journal of mathematical biology*, 67(6-7):1339–1368, Dec 2013.

[9] Shalyn C Clute, Yuri N Naumov, Levi B Watkin, Nuray Aslan, John L Sullivan, Katherine Luzuriaga, Raymond M Welsh, and Roberto Puzone. Broad cross-reactive T cell receptor repertoires recognizing dissimilar Epstein-Barr and influenza A virus epitopes. *J Immunol.*, 185(11):6753–6764, 2010.

[10] E Bridie Day, Carole Guillonneau, Stephanie Gras, Nicole L La Gruta, Dario A A Vignali, Peter C Doherty, Anthony W Purcell, Jamie Rossjohn, and Stephen J Turner. Structural basis for enabling T-cell receptor diversity within biased virus-specific $CD8^+$ T-cell responses. *Proceedings of the National Academy of Sciences of the United States of America*, 108(23):9536–9541, Jun 2011.

[11] Juscilene S. Menezes, Peter van den Elzen, Jordan Thornes, Donald Huffman, Nathalie M. Droin, Emanual Maverakis, and Eli E. Sercarz. A public T cell clonotype within a heterogeneous autoreactive repertoire is dominant in driving EAE. *The Journal of clinical investigation*, 117(8):2176–2185, Aug 2007.

[12] Phillippa Marrack, James P. Scott-Browne, Shaodong Dai, Laurent Gapin and John W. Kappler. Evolutionarily conserved amino acids in TCR V regions and MHC control their interaction. *Annu Rev Immunol.*, pages 171–203, 2008.

[13] Lei Yin, James Scott-Browne, John W. Kappler, Laurent Gapin, and Philippa Marrack. T cells and their eons-old obsession with MHC. *Immunological reviews*, 250(1):49–60, Nov 2012.

[14] EBI. Multiple Sequence Alignment, 2016. Available at `http://www.ebi.ac.uk/Tools/msa/`.

[15] Carsten Kemena and Cedric Notredame. Upcoming challenges for multiple sequence alignment methods in the high-throughput era. *Bioinformatics*, 25(19):2455–2465, 2009.

[16] Niclas Thomas, Katharine Best, Mattia Cinelli, Shlomit Reich-Zeliger, Hilah Gal, Eric Shifrut, Asaf Madi, Nir Friedman, John Shawe-Taylor, and Benny Chain. Tracking global changes induced in the CD4 T-cell receptor repertoire by immunization with a complex antigen using short stretches of CDR3 protein sequence. *Bioinformatics*, 30(22):3181–3188, 2014.

[17] Mattia Cinelli, Yuxin Sun, Katharine Best, James M. Heather, Shlomit Reich-Zeliger, Eric Shifrut, Nir Friedman, John Shawe-Taylor, and Benny Chain. Feature selection using a one dimensional naïve Bayes' classifier increases the accuracy of support vector machine classification of CDR3 repertoires. *Bioinformatics*, 33(7):951–955, 2017.

[18] Kenneth Murphy. *Janeway's immunobiology*. Garland Science, 2011.

[19] Jonathan; Male, David; Brostoff and David B Roth. *Immunology*. Elsevier, 2012.

[20] Dennis R Burton, Seamus J Martin, Peter Delves, and Ivan M Roitt. *Roitt's Essential Immunology*. John Wiley & Sons, Incorporated, 13 edition, 2016.

[21] Etymology of immunity. Available at `https://www.etymonline.com/word/immunity?ref=etymonline_crossreference`.

[22] Alexander Birbrair and Paul S Frenette. Niche heterogeneity in the bone marrow. *Ann N Y Acad Sci.*, 1370(1):82–96, Mar 2016.

[23] Wikipedia contributors. Haematopoiesis, 2018. Available at `https://en.wikipedia.org/wiki/Haematopoiesis`.

[24] Etymology of Dendrite. Available at https://www.etymonline.com/word/dendrite?ref=etymonline_crossreference.

[25] Jana Sarkander, Shintaro Hojyo, and Koji Tokoyoda. Vaccination to gain humoral immune memory. *Clinical & Translational Immunology*, 5(12):e120, Dec 2016.

[26] D. Ribatti, E. Crivellato, and A. Vacca. The contribution of Bruce Glick to the definition of the role played by the bursa of Fabricius in the development of the B cell lineage. *Clinical and Experimental Immunology*, 145(1):1–4, 2006.

[27] Bali Pulendran and Rafi Ahmed. Immunological mechanisms of vaccination. *Nature Immunology*, 12(6):509–517, 2011.

[28] Fenggen Yan, Xiumei Mo, Junfeng Liu, Siqi Ye, Xing Zeng, and Dacan Chen. Thymic function in the regulation of T cells, and molecular mechanisms underlying the modulation of cytokines and stress signaling (Review). *Molecular Medicine Reports*, 16(5):7175–7184, 2017.

[29] David Goodsell. T-Cell Receptor complexed with MHC I and II , 2014. Available at https://commons.wikimedia.org/wiki/File:63-T-CellReceptor-MHC.tif.

[30] Steven G.E. Marsh. Nomenclature for factors of the HLA system, update March 2017. *Human Immunology*, 78(5-6):461–465, 2017.

[31] Maria Ciofani and Juan Carlos Zúñiga-Pflücker. Determining $\gamma\delta$ versus $\alpha\beta$ T cell development. *Nature reviews. Immunology*, 10(9):657–663, Sep 2010.

[32] Heather J. Melichar, Kavitha Narayan, Sandy D Der, Yoshiki Hiraoka, Noemie Gardiol, Gregoire Jeannet, Werner Held, Cynthia A Chambers, and Joonsoo Kang. Regulation of $\gamma\delta$ Versus α T lymphocyte differentiation by the transcription factor SOX13. *Science*, 315(5809):230–233, 2007.

[33] Katherine J L Jackson, Marie J. Kidd, Yan Wang, and Andrew M. Collins. The shape of the lymphocyte receptor repertoire: Lessons from the B cell receptor. *Frontiers in Immunology*, 4(SEP):1–12, 2013.

[34] George Johnson and Tai Te Wu. The Kabat database and a bioinformatics example. *Methods in Molecular Biology; Antibody Engineering: Methods and Protocols*, 248:11–25, 2004.

[35] Michael S. Kuhns, Andrew T. Girvin, Lawrence O. Klein, Rebecca Chen, Kirk D. C. Jensen, Evan W. Newell, Johannes B. Huppa, Björn F. Lillemeier, Morgan Huse, Yueh-hsiu Y.-h. Chien, K. C. Garcia, M. M. Davis, and Others. Evidence for a functional sidedness to the TCR. *Proceedings of the National Academy of Sciences*, 107(11):5094–5099, 2010.

[36] René L Warren, J Douglas Freeman, Thomas Zeng, Gina Choe, Sarah Munro, Richard Moore, John R Webb, and Robert A Holt. Exhaustive T-cell repertoire sequencing of human peripheral blood samples reveals signatures of antigen selection and a directly measured repertoire size of at least 1 million clonotypes. *Genome research*, 21(5):790–797, May 2011.

[37] Daniel J. Laydon, Charles R. M. Bangham, and Becca Asquith. Estimating T-cell repertoire diversity: limitations of classical estimators and a new approach. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 370(1675):20140291, 2015.

[38] Olga V Britanova, Ekaterina V Putintseva, Mikhail Shugay, Ekaterina M Merzlyak, Maria A Turchaninova, Dmitriy B Staroverov, Dmitriy A Bolotin, Sergey Lukyanov, Ekaterina A Bogdanova, Ilgar Z Mamedov, Yuriy B Lebedev, and Dmitriy M Chudakov. Age-related decrease in TCR repertoire diversity measured with deep and normalized sequence profiling. *The Journal of Immunology*, 192(6):2689–2698, Mar 2014.

[39] Baojun Zhang, Qingzhu Jia, Cheryl Bock, Gang Chen, Haili Yu, Qingshan Ni, Ying Wan, Qijing Li, and Yuan Zhuang. Glimpse of natural selection of

long-lived T-cell clones in healthy life. *Proceedings of the National Academy of Sciences*, 113(35):9858–9863, 2016.

[40] R.M. Ritzel, J Crapser, AR. Patel, R. Verma, J.M. Grenier, A. Chauhan, E.R. Jellison, and L.D.. McCullough. Age-Associated Resident Memory CD8 T Cells in the Central Nervous System Are Primed To Potentiate Inflammation after Ischemic Brain Injury. 5(7):3318–3330, 2016.

[41] Meriem Attaf, Eric Huseby, and Andrew K. Sewell. $\alpha\beta$ t cell recteptors as predictors of health and disease. *Cell. Mol. Immunol.*, 12(4):391–399, Jul 2015.

[42] X. L. Hou, L. Wang, Y. L. Ding, Q. Xie, and H. Y. Diao. Current status and recent advances of next generation sequencing techniques in immunological repertoire. *Genes and immunity*, 17(3):153–164, 2016.

[43] Grant D Lythe, Robin E. Callard, Rollo L. Hoare, and Carmen Molina-Paris. How many TCR clonotypes does a body maintain? *Journal of Theoretical Biology*, 389:214–224, Jan 2015.

[44] C Chothia, D Ross Boswell, and A M Lesk. The outline structure of the T-cell $\alpha\beta$ receptor. *The EMBO journal*, 7(12):3745, 1988.

[45] Elvin A Kabat. *Sequences of immunoglobulin chains: Tabulation and analysis of amino acid sequences of precursors, V-regions, C-regions, J-chain and gbs2-microglobulins (NIH publication)*. National Institutes of Health, 1979.

[46] http://opig.stats.ox.ac.uk/webapps/stcrdab/. 4MNQ: CDR loop sequence and structure. Available at `http://opig.stats.ox.ac.uk/webapps/stcrdab/CDR?pdb=4MNQ`. 2017.

[47] Michael E Birnbaum, Juan L Mendoza, Dhruv K Sethi, Shen Dong, Jacob Glanville, Jessica Dobbins, Engin Özkan, Mark M Davis, Kai W Wucherpfennig, and K Christopher Garcia. Deconstructing the peptide-MHC specificity of T cell recognition. *Cell*, 157(5):1073–1087, 2014.

[48] Lawren C Wu, Delphine S Tuot, Daniel S Lyons, K Christopher Garcia, and Mark M Davis. Two-step binding mechanism for T-cell receptor recognition of peptide–MHC. *Nature*, 418(6897):552–556, 2002.

[49] Jens Hennecke and Don C Wiley. Structure of a complex of the human α/β T cell receptor (TCR) HA1.7, influenza hemagglutinin peptide, and major histocompatibility complex class II molecule, HLA-DR4 (DRA*0101 and DRB1*0401): insight into TCR cross-restriction and alloreactivity. *The Journal of experimental medicine*, 195(5):571–581, 2002.

[50] Benjamin North, Andreas Lehmann, and Roland L Dunbrack. A new clustering of antibody CDR loop conformations. *Journal of molecular biology*, 406(2):228–256, Feb 2011.

[51] Tania Cukalac, Wan-Ting Kan, Pradyot Dash, Jing Guan, Kylie M Quinn, Stephanie Gras, Paul G Thomas, and Nicole L La Gruta. Paired TCR $\alpha\beta$ a analysis of virus-specific $CD8^+$ T cells exposes diversity in a previously defined 'narrow' repertoire. *Immunology and cell biology*, 93(9):804–814, 2015.

[52] Ellis L Reinherz, Kemin Tan, Lei Tang, Petra Kern, Jin-huan Liu, Yi Xiong, Rebecca E Hussey, Alex Smolyar, Brian Hare, Rongguang Zhang, and Others. The crystal structure of a T cell receptor in complex with peptide and MHC class II. *Science*, 286(5446):1913–1921, 1999.

[53] Jens Hennecke, Andrea Carfi, and Don C Wiley. Structure of a covalently stabilized complex of a human alphabeta T-cell receptor, influenza HA peptide and MHC class II molecule, HLA-DR1. *The EMBO journal*, 19(21):5611–5624, 2000.

[54] Yuan-hua Ding, Kathrine J Smith, David N Garboczi, Ursula Utz, William E Biddison, and Don C Wiley. Two human T cell receptors bind in a similar diagonal mode to the HLA-A2/Tax peptide complex using different TCR amino acids. *Immunity*, 8(4):403–411, 1998.

[55] Jean-Baptiste Reiser, Claudine Darnault, Annick Guimezanes, Claude Grégoire, Thomas Mosser, Anne-Marie Schmitt-Verhulst, Juan Carlos Fontecilla-Camps, Bernard Malissen, Dominique Housset, and Gilbert Mazza. Crystal structure of a T cell receptor bound to an allogeneic MHC molecule. *Nature immunology*, 1(4):291–297, 2000.

[56] Evan W Newell, Lauren K Ely, Andrew C Kruse, Philip A Reay, Stephanie N Rodriguez, Aaron E Lin, Michael S Kuhns, K Christopher Garcia, and Mark M Davis. Structural basis of specificity and cross-reactivity in T cell receptors specific for cytochrome c–I-Ek. *The Journal of Immunology*, 186(10):5823–5832, 2011.

[57] David K Cole, Anna M Bulek, Garry Dolton, Andrea J Schauenberg, Barbara Szomolay, William Rittase, Andrew Trimby, Prithiviraj Jothikumar, Anna Fuller, Ania Skowera, and Others. Hotspot autoimmune T cell receptor binding underlies pathogen and insulin peptide cross-reactivity. *The Journal of clinical investigation*, 126(6):2191, 2016.

[58] M Regner. Cross-reactivity in T-cell antigen recognition. *Immunology and cell biology*, 79(2):91–100, Apr 2001.

[59] Catherine Mazza, Nathalie Auphan-Anezin, Claude Gregoire, Annick Guimezanes, Christine Kellenberger, Alain Roussel, Alice Kearney, P Anton Van Der Merwe, Anne-Marie Schmitt-Verhulst, and Bernard Malissen. How much can a T-cell antigen receptor adapt to structurally distinct antigenic peptides? *The EMBO journal*, 26(7):1972–1983, 2007.

[60] Leremy A Colf, Alexander J Bankovich, Nicole A Hanick, Natalie A Bowerman, Lindsay L Jones, David M Kranz, and K Christopher Garcia. How a single T cell receptor recognizes both self and foreign MHC. *Cell*, 129(1):135–146, 2007.

[61] Ryan W. Nelson, Daniel Beisang, Noah J. Tubo, Thamotharampillai Dileepan, Darin L. Wiesner, Kirsten Nielsen, Marcel Wüthrich, Bruce S.

Klein, Dmitri I. Kotov, Justin A. Spanier, Brian T. Fife, James J. Moon, Marc K. Jenkins, and Others. T cell receptor cross-reactivity between similar foreign and self peptides influences naive cell population size and autoimmunity. *Immunity*, 42(1):95–107, 2015.

[62] Yiyuan Yin and Roy A Mariuzza. The multiple mechanisms of T cell receptor cross-reactivity. *Immunity*, 31(6):849–851, Dec 2009.

[63] Vanessa Venturi, David A Price, Daniel C Douek, and Miles P Davenport. The molecular basis for public T-cell responses? *Nature reviews. Immunology*, 8(3):231–238, Mar 2008.

[64] Paul G Thomas, Andreas Handel, Peter C Doherty, and Nicole L La Gruta. Ecological analysis of antigen-specific CTL repertoires defines the relationship between naive and immune T-cell populations. *Proceedings of the National Academy of Sciences*, 110(5):1839–1844, 2012.

[65] Hanjie Li, Ye Congting, Guoli Ji, Jiahuai Han, C Ye, Guoli Ji, Jiahuai Han, Ye Congting, Guoli Ji, Jiahuai Han, C Ye, Guoli Ji, and Jiahuai Han. Determinants of public T cell responses. *Cell Research*, 22(1):33–42, 2012.

[66] Janko Nikolich-Žugich, Mark K Slifka, and Ilhem Messaoudi. The many important facets of T-cell repertoire diversity. *Nature Reviews Immunology*, 4(2):123–132, Feb 2004.

[67] V. Venturi, K. Kedzierska, D. A. Price, P. C. Doherty, D. C. Douek, S. J. Turner, and M. P. Davenport. Sharing of T cell receptors in antigen-specific responses is driven by convergent recombination. *Proceedings of the National Academy of Sciences*, 103(49):18691–18696, 2006.

[68] A Madi, E Shifrut, S Reich-Zeliger, H Gal, K Best, W Ndifon, B Chain, I R Cohen, and N Friedman. T-cell receptor repertoires share a restricted set of public and abundant CDR3 sequences that are associated with self-related immunity. *Genome Res.*, 24(10):1603–1612, Oct 2014.

[69] Yuxin Sun, Katharine Best, Mattia Cinelli, James M. Heather, Shlomit Reich-Zeliger, Eric Shifrut, Nir Friedman, John Shawe-Taylor, and Benny Chain. Specificity, privacy, and degeneracy in the CD4 T cell receptor repertoire following immunization. *Frontiers in Immunology*, 8(APR), 2017.

[70] Wikipedia. Black 6 mice, 2011. Available at `http://www.slate.com/articles/health_and_science/the_mouse_trap/2011/11/black_6_lab_mice_and_the_history_of_biomedical_research.html`.

[71] https://www.jax.org/. C57BL/6J. Available at `https://www.jax.org/strain/000664`.

[72] Wikipedia. Ovalbumin, 2016. Available at `https://en.wikipedia.org/wiki/Ovalbumin`.

[73] Santa Cruz. Available at `https://www.scbt.com/scbt/product/freunds-complete-adjuvant` Technology. Technology, Santa Cruz.

[74] Worthington Biochemical Company. Worthington. Available at `http://www.worthington-biochem.com/oa/default.html`.

[75] Andrew D Nisbet, Richard H Sandry, Arthur J G Moir, Linda A Fothergill, and John E Fothergill. The Complete Amino-Acid Sequence of Hen Ovalbumin. *European Journal of Biochemistry*, 115(2):335–345, 1981.

[76] Niclas Thomas, James Heather, Wilfred Ndifon, John Shawe-Taylor, and Benjamin Chain. Decombinator: a tool for fast, efficient gene assignment in T-cell receptor sequences using a finite state machine. *Bioinformatics (Oxford, England)*, 29(5):542–550, Mar 2013.

[77] Katharine Best. *Computational approaches to the analysis of the T cell receptor repertoire*. Phd thesis, 2016, Available at `http://discovery.ucl.ac.uk/1485661/`.

[78] Zhou Li, Ma Long, Liu ChunMei, Shi Bin, Yu Jiang, Ma Rui, Ma Qingqing, and Yao XinSheng. Composition and variation analysis of TCR beta-chain CDR3 repertoire in the thymus and spleen of MRL/lpr mouse at different ages. *Immunogenetics*, 67(1):25–37, 2015.

[79] C Pannetier, M Cochet, S Darche, A Casrouge, M Zoller, and P Kourilsky. The sizes of the CDR3 hypervariable regions of the murine T-cell receptor beta chains vary as a function of the recombined germ-line segments. *Proc Natl Acad Sci U S A*, 90(9):4319–23., 1993.

[80] Z C Kou, J S Puhr, M Rojas, W T McCormack, M M Goodenow, and J W Sleasman. T-Cell receptor V-beta repertoire CDR3 length diversity differs within CD45RA and CD45RO T-cell subsets in healthy and human immunodeficiency virus-infected children. *Clin Diagn Lab Immunol*, 7(6):953–959, 2000.

[81] E P Rock, P R Sibbald, M M Davis, and Y H Chien. CDR3 length in antigen-specific immune receptors. *The Journal of experimental medicine*, 179(1):323–328, Jan 1994.

[82] Patrick Miqueu, Marina Guillet, Nicolas Degauque, Jean-Christophe Doré, Jean-Paul Soulillou, and Sophie Brouard. Statistical analysis of CDR3 length distributions for the assessment of T and B cell repertoire biases. *Molecular immunology*, 44(6):1057–1064, 2006.

[83] A Casrouge, E Beaudoing, S Dalle, C Pannetier, J Kanellopoulos, and P Kourilsky. Size estimate of the $\alpha\beta$ TCR repertoire of naive mouse splenocytes. *J. Immunol.*, 164(11):5782–5787, Jun 2000.

[84] Paul Jaccard. The distribution of the flora in the alpine zone. *New phytologist*, 11(2):37–50, 1912.

[85] Corrado Gini. Variability and Mutability. *Cuppini, Bologna*, 1912.

[86] Rachael J M Bashford-Rogers, Anne L Palser, Brian J Huntly, Richard Rance, George S Vassiliou, George A Follows, and Paul Kellam. Network properties derived from deep sequencing of human B-cell receptor repertoires delineate B-cell populations. *Genome Res.*, pages 1874–1884, 2013.

[87] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of machine learning*. MIT press, 2012.

[88] Niclas Thomas. *Computational Approaches to the Study of T Cell Migration and the T Cell Receptor Repertoire*. PhD thesis, London, Uk, 2013.

[89] V. Vapnik and A. Lerner. Pattern recognition using generalized. *Automation and Remote Control*, 24:774–780, 1963.

[90] B E Boser, I M Guyon, and V N Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152, 1992.

[91] C Cortes and V Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[92] Peter L Bartlett. The Sample Complexity of Pattern Classification with Neural Networks: The Size of the Weights is More Important than the Size of the Network. *IEEE Trans. Information Theory*, 44(2):525–536, 1998.

[93] John Shawe-Taylor, Peter L Bartlett, Robert C Williamson, and Martin Anthony. Structural risk minimization over data-dependent hierarchies. *Information Theory, IEEE Transactions on*, 44(5):1926–1940, 1998.

[94] John Shawe-taylor and Nello Cristianini. Margin Distribution and Soft Margin, 1999.

[95] N Cristianini and J Shawe-Taylor. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press, 2000.

[96] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. MIT press, 2002.

[97] S. Abe. *Support vector machines for pattern classification* . Springer, 2010.

[98] Leslie G Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

[99] M A Aizerman, E A Braverman, and L Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. In *Automation and Remote Control,*, number 25 in Automation and Remote Control,, pages 821–837, 1964.

[100] F. Leisch K. Hornik, A. Weingessel and M. D. M. David. *Package "e1071"*. meyerr-projectorg, 2018. Available at `https://cran.r-project.org/web/packages/e1071/e1071.pdf`.

[101] Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[102] Josef Sivic and Andrew Zisserman. Efficient visual search of videos cast as text retrieval. *PAMI*, 31(4):591–606, Apr 2009.

[103] G Csurka, C Bray, C Dance, and L Fan. Visual categorization with bags of keypoints. *Workshop on Statistical Learning in Computer Vision, ECCV*, pages 1–22, 2004.

[104] Thorsten Joachims. *Text categorization with support vector machines: Learning with many relevant features*. Springer, 1998.

[105] David G Lowe. Object Recognition from Local Scale-Invariant Features. In *Proceedings of the International Conference on Computer Vision-Volume 2*, ICCV '99, page 1150, Washington, DC, USA, 1999. IEEE Computer Society.

[106] Wikipedia. Euclidean distance, 2016. Available at `https://en.wikipedia.org/wiki/Euclidean_distance`.

[107] N G De Bruijn. A combinatorial problem. *Proceedings of the Section of Sciences of the Koninklijke Nederlandse Akademie van Wetenschappen*, 49(7):758–764, 1946.

[108] Phillip E C Compeau, Pavel A. Pevzner, and Glenn Tesler. How to apply de Bruijn graphs to genome assembly. *Nature Biotechnology*, 29(11):987–991, 2011.

[109] Aaron Sievers, Katharina Bosiek, Marc Bisch, Chris Dreessen, Jascha Riedel, Patrick Froß, Michael Hausmann, and Georg Hildenbrand. K-mer content, correlation, and position analysis of genome dna sequences for the identification of function and evolutionary features. *Genes*, 8(4):1–18, 2017.

[110] Ezzeddin Kamil Mohamed Hashim and Rosni Abdullah. Rare k-mer DNA: Identification of sequence motifs and prediction of CpG island and promoter. *Journal of Theoretical Biology*, 387:88–100, 2015.

[111] Daniel R. Zerbino and Ewan Birney. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, 18(5):821–829, 2008.

[112] Manpreet Kaur and Usvir Kaur. Comparison Between K-Mean and Hierarchical Algorithm Using Query Redirection. *International Journal of Advanced Research in Computer Science and Software Engineering*, 3(7):1454–1459, 2013.

[113] Malay K. Pakhira. A linear time-complexity *k*-means algorithm using cluster shifting. *Proceedings - 2014, 6^{th} International Conference on Computational Intelligence and Communication Networks, CICN 2014*, pages 1047–1051, 2014.

[114] William H E Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of Classification*, 1(1):7–24, 1984.

[115] Wikipedia. k-means clustering, 2016. Available at `https://en.wikipedia.org/wiki/K-means_clustering`.

[116] http://www.genome.jp. Amino acid indices, substitution matrices and pair-wise contact potentials. Available at `http://www.genome.jp/aaindex/`.

[117] William R. Atchley, Jieping Zhao, Andrew D. Fernandes, and Tanja Druke. Solving the protein sequence metric problem. *Proceedings of the National Academy of Sciences of the United States of America*, 102(18):6395–6400, May 2005.

[118] Maria Sandberg, Lennart Eriksson, Jörgen Jonsson, Michael Sjöström, and Svante Wold. New chemical descriptors relevant for the design of biologically active peptides. A multivariate characterization of 87 amino acids. *Journal of medicinal chemistry*, 41(14):2481–2491, Jul 1998.

[119] Akinori Kidera, Yasuo Konishi, Oka Masahito, Ooi Tatsuo, and Harold A. Scheraga. Statistical Analysis of the Physical Properties of the 20 Naturally Occurring Amino Acids. *Journal of Protein Chemistry*, 4(23):23–55, Jan 1985.

[120] Vincent Labatut and Hocine Cherifi. Accuracy measures for the comparison of classifiers. *The 5^{th} International Conference on Information Technology*, (May):11, 2011.

[121] Darren J. Wilkinson. Bayesian methods in bioinformatics and computational systems biology. *Briefings in bioinformatics*, 8(2):109–116, Mar 2007.

[122] L Cosmides and J Tooby. Are humans good intuitive statisticians after all? Rethinking some conclusions from the literature on judgement under uncertainty. *Cognition*, 58:197–213, 1995.

[123] Mr Bayes and Mr Price. An Essay towards solving a Problem in the Doctrine of Chances. By the late Rev. Mr. Bayes, FRS communicated by Mr. Price, in

a letter to John Canton, AMFRS. *Philosophical Transactions*, 53:370–418, 1763.

[124] Pierre-Simon Laplace. *Essai philosophique sur les probabilités*. Paris Bachelier, 1814.

[125] Tim O.F. Conrad, Martin Genzel, Nada Cvetkovic, Niklas Wulkow, Alexander Leichtle, Jan Vybiral, Gitta Kutyniok, and Christof Schütte. Sparse Proteomics Analysis - a compressed sensing-based approach for feature selection and classification of high-dimensional proteomics mass spectrometry data. *BMC Bioinformatics*, 18(1):1–20, 2017.

[126] Wei Du, Zhongbo Cao, Tianci Song, Ying Li, and Yanchun Liang. A feature selection method based on multiple kernel learning with expression profiles of different types. *BioData Mining*, 10(1):1–16, 2017.

[127] Yang Yang, Ning Huang, Luning Hao, and Wei Kong. A clustering-based approach for efficient identification of microRNA combinatorial biomarkers. *BMC Genomics*, 18(Suppl 2):1–14, 2017.

[128] Qi Zhang, Yang Xiao, Jingfeng Suo, Jun Shi, Jinhua Yu, Yi Guo, Yuanyuan Wang, and Hairong Zheng. Sonoelastomics for breast tumor classification: A radiomics approach with clustering-based feature selection on sonoelastography. *Ultrasound in Medicine & Biology*, 43(5):1058–1069, 2017.

[129] Janez Brank, Marko Grobelnik, and Natasa Milic-frayling. *Feature selection using support vector machines.* Number January. 2002.

[130] Myungjin Moon and Kenta Nakai. Stable feature selection based on the ensemble L 1 -norm support vector machine for biomarker discovery. *BMC Genomics*, 17(S13):1026, 2016.

[131] E.E. Bron, M. Smits, W.J. Niessen, and S. Klein. Feature selection based on the SVM weight vector for classification of dementia. *IEEE Journal of Biomedical and Health Informatics*, 19(5):1617–1626, 2015.

[132] Enkelejda Miho, Alexander Yermanos, Cédric R. Weber, Christoph T. Berger, Sai T. Reddy, and Victor Greiff. Computational strategies for dissecting the high-dimensional complexity of adaptive immune repertoires. *Frontiers in Immunology*, 9(FEB):1–27, 2018.

[133] Qiong Liu, Qiong Gu, and Zhao Wu. Feature selection method based on support vector machine and shape analysis for high-throughput medical data. *Computers in Biology and Medicine*, 91(October):103–111, 2017.

[134] Huidong Li, Pei Zhang, Shuaifang Yuan, Huiyuan Tian, Dandan Tian, and Min Liu. Modeling analysis of the relationship between atherosclerosis and related inflammatory factors. *Saudi Journal of Biological Sciences*, 24(8):1803–1809, 2017.

[135] Mark M. Davis, Cristina M. Tato, and David Furman. Systems immunology: Just getting started. *Nature Immunology*, 18(7):725–732, 2017.

[136] Quentin Marcou. Probabilistic approaches to the adaptive immune repertoire: a data-driven approach. 2018.

[137] James R Norris. *Markov chains*. Cambridge University Press, 1998.

[138] Britannica. Markov process, 2016. Available at `https://www.britannica.com/topic/Markov-process`.

[139] Leonard E Baum and Ted Petrie. Statistical inference for probabilistic functions of finite state Markov chains. *The annals of mathematical statistics*, pages 1554–1563, 1966.

[140] Kevin Karplus, Christian Barrett, and Richard Hughey. Hidden Markov Models for Detecting Remote Protein Homologies Santa Cruz , CA 95064 Abstract 1 Introduction 2 Test Sets. 1999.

[141] Lawrence R Rabiner. Readings in Speech Recognition. pages 267–296, 1990.

[142] Jayaweera and Dias. Hidden Markov Model Based Part of Speech Tagger for Sinhala Language. *CoRR*, abs/1407.2, 2014.

[143] Hans-Gunter Hirsch. HMM adaptation for applications in telecommunication. *Speech Communication*, 34(1-2):127–139, 2001.

[144] Selcuk Sandikci, Pinar Duygulu, and Bulent Ozgule. HMM Based Behavior Recognition of Laboratory Animals. 2012.

[145] I D Jonsen, M Basson, S Bestley, M V Bravington, T A Patterson, M W Pedersen, R Thomson, U H Thygesen, and S J Wotherspoon. State-space models for bio-loggers: A methodological road map. *Deep Sea Research Part II: Topical Studies in Oceanography*, 88–89:34–46, 2013.

[146] M J Bishop and E A Thompson. Maximum likelihood alignment of DNA sequences. *Journal of molecular biology*, 190(2):159–165, Jul 1986.

[147] Robert D Finn, Penelope Coggill, Ruth Y Eberhardt, Sean R Eddy, Jaina Mistry, Alex L Mitchell, Simon C Potter, Marco Punta, Matloob Qureshi, Amaia Sangrador-Vegas, Gustavo A Salazar, John G Tate, and Alex Bateman. The Pfam protein families database: towards a more sustainable future. *Nucleic Acids Research*, 44(Database-Issue):279–285, 2016.

[148] Hao Wu, Brian Caffo, Harris A Jaffee, Rafael A Irizarry, and Andrew P Feinberg. Redefining CpG islands using hidden Markov models. *Biostatistics*, page kxq005, 2010.

[149] Adam Krejci, Ted R Hupp, Matej Lexa, Borivoj Vojtesek, and Petr Muller. Hammock: a hidden Markov model-based peptide clustering algorithm to identify protein-interaction consensus motifs in large datasets. *Bioinformatics*, 32(1):9–16, 2016.

[150] Byung-Jun Yoon. Hidden Markov Models and their Applications in Biological Sequence Analysis. *Current genomics*, 10(6):402–415, Sep 2009.

[151] Naila Mimouni, Gerton Lunter, Charlotte Deane, Lunter Gerton, and Deane Charlotte. Hidden Markov Models for Protein Sequence Alignment. *Oxford: University of Oxford*, pages 1–26, 2004.

[152] A Krogh, M Brown, I S Mian, K Sjölander, and D Haussler. Hidden Markov models in computational biology. Applications to protein modeling. *Journal of molecular biology*, 235(5):1501–1531, Feb 1994.

[153] T Koski. *Hidden Markov Models for Bioinformatics*. Computational Biology. Springer Netherlands, 2001.

[154] Anders Krogh. An Introduction to Hidden Markov Models for Biological Sequences. *Computational Methods in Molecular Biology*, 32:45–63, 1998.

[155] Wikipedia. Pseudocount, 2016. Available at `https://en.wikipedia.org/wiki/Pseudocount`.

[156] P Baldi, Y Chauvin, T Hunkapiller, and M A McClure. Hidden Markov models of biological primary sequence information. *Proceedings of the National Academy of Sciences of the United States of America*, 91(3):1059–1063, Feb 1994.

[157] V Di Francesco, J Garnier, and P J Munson. Protein topology recognition from secondary structure sequences: application of the hidden Markov models to the alpha class proteins. *Journal of molecular biology*, 267(2):446–463, Mar 1997.

[158] Frederick Jelinek. *Statistical Methods for Speech Recognition*. MIT Press, Cambridge, MA, USA, 1997.

[159] Andrew J. Viterbi. Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithm. pages 260–269, 1967.

[160] Robert D Finn, Jody Clements, and Sean R Eddy. HMMER web server: interactive sequence similarity searching. *Nucleic acids research*, 39(Web Server issue):W29—-37, Jul 2011.

[161] HMMer. HMMer, 2016. Available at `http://hmmer.org/`.

[162] Compbio. Sequence Alignment and Modeling System, Available at `https://compbio.soe.ucsc.edu/sam.html`.

[163] S R Eddy. HMMER: Profile hidden Markov models for biological sequence analysis. 2001.

[164] S A Krawetz and D D Womble. *Introduction to Bioinformatics: A Theoretical And Practical Approach*. Humana Press, 2003.

[165] Pfam. Pfam, 2016. Available at `http://pfam.xfam.org/`.

[166] Yale. HMMER User Guide, 2016. Available at `http://www.csb.yale.edu/userguides/seq/hmmer/docs/node11.html`.

[167] Timothy F. Oliver, Leow Yuan Yeow, and Bertil Schmidt. High performance database searching with hmmer on fpgas. In *21th International Parallel and Distributed Processing Symposium (IPDPS 2007), Proceedings, 26-30 March 2007, Long Beach, California, USA*, pages 1–7, 2007.

[168] Johannes Soding, Johannes Söding, and Johannes Soding. Protein homology detection by HMM-HMM comparison. *Bioinformatics (Oxford, England)*, 21(7):951–960, Apr 2005.

[169] Clustal Omega. Clustal Omega, 2016. Available at `http://www.ebi.ac.uk/Tools/msa/clustalo/`.

[170] Massimo Andreatta, Ole Lund, and Morten Nielsen. Simultaneous alignment and clustering of peptide data using a Gibbs sampling approach. *Bioinformatics (Oxford, England)*, 29(1):8–14, Jan 2013.

[171] Taehyung Kim, Marc S. Tyndel, Haiming Huang, Sachdev S. Sidhu, Gary D. Bader, David Gfeller, Philip M. Kim, Kim Taehyung, Marc S. Tyndel, Haiming Huang, Sachdev S. Sidhu, Gary D. Bader, David Gfeller, and

Philip M. Kim. MUSI: an integrated system for identifying multiple specificity from very large peptide or nucleic acid data sets. *Nucleic acids research*, 40(6):e47, Mar 2012.

[172] T D Schneider and R M Stephens. Sequence logos: a new way to display consensus sequences. *Nucleic acids research*, 18(20):6097–6100, Oct 1990.

[173] Gavin E Crooks, Gary Hon, John-Marc Chandonia, and Steven E Brenner. WebLogo: a sequence logo generator. *Genome research*, 14(6):1188–1190, 2004.

[174] Philippa Marrack, James P Scott-Browne, Shaodong Dai, Laurent Gapin, and John W Kappler. Evolutionarily conserved amino acids that control TCR-MHC interaction. *Annu. Rev. Immunol.*, 26:171–203, 2008.

[175] Jack Kyte and Russell F. Doolittle. A simple method for displaying the hydropathic character of a protein. *Journal of Molecular Biology*, 157(1):105–132, 1982.

[176] IARC. Amino Acids Properties, Available at `http://p53.iarc.fr/AAproperties.aspx`.

[177] PK. Gupta. *Molecular Biology and Genetic Engineering*. Rastogi Publications, New Delhi, , 2nd edit edition, 2009.

[178] Roberts K Alberts B, Johnson A, Lewis J, Raff M and Walter P. Chapter 3. Proteins pag:125-129. In *Molecular Biology of the Cell*, chapter Polar amin. Garland Science, New York, 5th editio edition, 2008.

[179] H O Villar and L M Kauvar. Amino acid preferences at protein binding sites. *FEBS Lett*, 349(1):125–130, 1994.

[180] Leland Wilkinson and Michael Friendly. The History of the Cluster Heat Map. *The American Statistician*, 63(2):179–184, 2009.

[181] Luyang Huo, Han Zhang, Xueting Huo, Yasong Yang, Xueqiong Li, and Yanbin Yin. pHMM-tree: phylogeny of profile hidden Markov models. *Bioinformatics*, 33(7):1093–1095, 2017.

[182] Shanfeng Zhu, Keiko Udaka, John Sidney, Alessandro Sette, Kiyoko F Aoki-Kinoshita, and Hiroshi Mamitsuka. Improving MHC binding peptide prediction by incorporating binding data of auxiliary MHC molecules. *Bioinformatics*, 22(13):1648–1655, 2006.

[183] Laurent Jacob and Jean-Philippe Philippe Vert. Efficient peptide MHC-I binding prediction for alleles with few known binders. *Bioinformatics*, 24(3):358–366, 2008.

[184] K "Held, E Beltran, M Moser, R Hohlfeld, and K " Dornmair. T-cell receptor repertoire of human peripheral CD161hiTRAV1-2+MAIT cells revealed by next generation sequencing and single cell analysis. *Hum. Immunol.*, 76(9):607–614, Sep 2015.

[185] David Redmond, Asaf Poran, and Olivier Elemento. Single-cell TCRseq: Paired recovery of entire T-cell α and β chain transcripts in T-cell receptors from single-cell RNAseq. *Genome Medicine*, 8(1):1, 2016.

[186] E Rosati, C M Dowds, E Liaskou, E K K Henriksen, T H Karlsen, and A Franke. Overview of methodologies for T-cell receptor repertoire analysis. *BMC Biotechnol.*, 17(1):61, Jul 2017.

[187] Daniel J. Woodsworth, Mauro Castellarin, and Robert A. Holt. Sequence analysis of T-cell repertoires in health and disease. *Genome Medicine*, 5(10):1, 2013.

[188] Qian Qi, Mary M Cavanagh, Sabine Le Saux, Hong Namkoong, Chulwoo Kim, Emerson Turgano, Yi Liu, Chen Wang, Sally Mackey, Gary E Swan, L Cornelia, Richard A Olshen, Scott D Boyd, Cornelia M Weyand, Lu Tian, and J Jörg. Diversification of the antigen-specific T cell receptor repertoire after varicella zoster vaccination. *HHS Public Access*, 8(332):20, 2016.

[189] R. J. De Boer, D. Homann, and A. S. Perelson. Different Dynamics of $CD4^+$ and $CD8^+$ T Cell Responses During and After Acute Lymphocytic Choriomeningitis Virus Infection. *The Journal of Immunology*, 171(8):3928–3935, 2003.

[190] Pradyot Dash, Andrew J. Fiore-Gartland, Tomer Hertz, George C. Wang, Shalini Sharma, Aisha Souquette, Jeremy Chase Crawford, E. Bridie Clemens, Thi H.O. Nguyen, Katherine Kedzierska, Nicole L. La Gruta, Philip Bradley, and Paul G. Thomas. Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature*, 547(7661):89–93, 2017.