Identification of Antigen-Specific Patterns from High-Dimensional Sequencing Data

Yuxin Sun

A dissertation submitted in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

of

University College London.

Department of Computer Science

University College London

June 18, 2020

I, Yuxin Sun, 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 work.

Abstract

T cells recognize antigens using a diverse set of antigen-specific T-cell receptors (TCRs) on the surface. This poses two challenges for studying TCRs that respond to a given antigen. First, the enormous diversity of the TCR repertoire creates an ultra-high dimensional feature space; second, TCRs that respond to an antigen are often correlated. This thesis aims to develop efficient machine learning algorithms concerning both problems for feature selection from high-dimensional feature spaces.

Our research concerns two subproblems: identification of antigen-enriched sequence motifs within the CDR3 region of TCRs and antigen-enriched entire TCR sequences. We apply a string kernel and a Fisher kernel to represent subsequences and develop fast algorithms to learn antigen-specific subsequences from graph-represented features. Both fixed-length and varying-length subsequences from mouse samples are selected with high efficiency and accuracy. Our results also suggest that short subsequences are found at specific positions, which may correspond to the actual interacting regions between TCR and MHC-peptide complex.

We further develop fast algorithms to solve exclusive group Lasso and provide a novel methodology to select entire TCR sequences that are relevant to specific antigens. Our solution concerns a notoriously difficult problem in feature selection to select highly correlated features. Experiments on synthetic data show good performance under various correlation settings. The proposed algorithms are also validated on real-world data to select a sparse set of entire TCRs with high accuracy.

Impact Statement

In this thesis, we focus on uncorrelated and correlated feature selection from high-dimensional feature space with an emphasis on identifying biological patterns to model changes in the T-cell receptor (TCR) repertoire. We develop fast algorithms to convert biological sequences to numerical feature space with string kernel and Fisher kernel and select subsequences from a graph-represented feature space using linear programming boosting. We also develop efficient algorithms to tackle the challenging problem to select correlated features using exclusive group Lasso. We address the unresolved question on group allocation in exclusive group Lasso for stable and consistent feature selection results.

The thesis aims to contribute to both immunology and machine learning. In terms of immunology, the thesis presents new methods to classify the TCR repertoires, as well as identifying antigen-specific sequence features of TCRs. The proposed algorithms can be generalized to recognize antigen-specific patterns in response to many diseases. Our methods can be further extended beyond the TCR repertoires, such as identifying gene expression that correlates with disease status. Potential applications include designing more efficient and effective diagnostic and therapeutic strategies in clinical studies.

In terms of machine learning, the presented algorithms provide novel tools for feature selection from high-dimensional datasets that consist of either uncorrelated or correlated features. We also present fast solutions to the proposed algorithms. The machine learning strategies we have developed may be further generalized to the broad field of feature selection and can be adapted to select relevant features and eliminate irrelevant features in various high-dimensional datasets.

Acknowledgements

First of all, I would like to express my sincere gratitude to both my supervisors, Prof. John Shawe-Taylor and Prof. Benny Chain, for their immeasurable support, their patience, and comprehensive knowledge to both my Ph.D studies and thesis writing. I have been extremely lucky to have them as my supervisors. I simply could not imagine having better supervisors for my Ph.D studies.

Besides my supervisors, I would like to thank all current and past colleagues in our lab, in particular Dr Annemarie Woolston and Dr Mazlina Ismail, who have offered invaluable contributions and help throughout my studies.

My sincere thanks also go to Dong Jin and Xuejiao Zhang, for offering love and support all the time. Thanks for being the best friends in the world.

I am immensely grateful to my parents for their unconditional support for my studies and unshakeable belief in my abilities. I would not have been able to start my Ph.D studies or conduct research without their help.

Finally, special thanks go to my cat Shinsuke for his companionship and for being an angel. I would not have survived these years without him.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The immune system involves the sophisticated cooperation of many different components from two subsystems: the innate immune system and the adaptive immune system. Up to 750 million years old, the innate immune system provides a fast and broad defense against various foreign invaders such as parasites, fungi, bacteria, and viruses. Unlike the innate immune system, the adaptive system adapts to recognize and memorize specific antigens with the help of B cells and T cells. Each T cell has multiple identical T-cell receptors (TCRs) expressed on its surface, recognizing antigens with a diverse set of antigen-specific receptors by somatic rearrangement in an imprecise process with random deletions and additions. T cells differentiate into memory and effector cells in response to their cognate antigens. The clonal selection theory suggests that the adaptive immune response is encoded by differentiated antigen-specific T cells to defend against the counter-specific antigens to which the host body has been previously exposed. A prediction of the theory is that the frequency of the "experienced" memory and effector cells is greater than the frequency of the "inexperienced" naive cells. This prediction naturally leads to our research question: identification of antigen-specific TCRs in response to a given antigen. The introduction of high-throughput sequencing allows reads of millions or even billions of sequences at relatively low costs, opening up new approaches to examining the diversity of the T-cell repertoire but also introducing ultra-high dimensional data for analysis. As a result, our research question on feature selection from the T-cell repertoire poses two major challenges: 1). high dimensionality

of the T-cell repertoire; 2). correlations between antigen-specific TCRs because of their functional similarity.

Feature selection from high-dimensional data has been an important field in machine learning and statistics. In the era of big data, our world is awash with data. The number of features measured on a single subject can be huge and often larger than the number of samples. A sparsity assumption therefore allows us to tackle the high-dimensional problems and extract informative and interpretable features from big data.

A simple solution for a sparse model is ℓ_1 norm. The sparsity-inducing ℓ-norm provides sparse solutions which return only a small subset of features with non-zero weights. These approaches, including Lasso and ℓ_1-regularized SVM, often benefit from efficient algorithms to solve the optimization problem and have therefore been used in many high-dimensional feature selection problems. However, despite good performance with uncorrelated features, ℓ_1-regularized methods generally perform erratically with correlated features and fail to include all correlated features. Considering the non-sparse solution from ℓ_2-regularized methods, a number of approaches with a mixture of ℓ_1 and ℓ_2 norms, such as elastic net, group Lasso, sparse group Lasso, and exclusive group Lasso, have been introduced and applied to various problems. Unlike Lasso, which requires stringent conditions to correctly select all relevant features, elastic net enjoys more relaxed conditions in feature selection, particularly when dealing with correlated features. However, an exhaustive parameter tuning process is needed to fit an elastic net model. Group Lasso and sparse group Lasso are widely used to learn patterns that are naturally grouped by biological pathways, but often do not perform well on data without a clear group structure. Although exclusive group Lasso does not require the input features to be naturally grouped, appropriate group allocation is needed to correctly select features, which is not addressed by previous studies. Furthermore, exclusive group Lasso was designed to select uncorrelated features, and much less is known about its performance in correlated feature selection.

In this thesis, we validate both ℓ_1 and $\ell_{1,2}$-regularized methods with an empha-

sis on feature selection. To cope with the high-dimensionality of the TCR repertoire and possible correlations between antigen-specific TCR sequences, we develop fast algorithms to solve linear programming boosting and exclusive group Lasso which select both subsequences of fixed and varying lengths as well as entire sequences. We address the application of exclusive group Lasso to select correlated features. We also fill the gap in exclusive group Lasso for correlated feature selection with simple, yet powerful pipelines for group allocation.

The main contribution of this thesis is two-fold. First, in terms of immunology, we present selected subsets of both CDR3 and TCR sequences which may be used in other studies concerning the same antigens and disease, as well as providing novel machine learning pipelines which can be generalized to design diagnostic strategies and prognostic plans for other diseases in clinical studies. Second, from a machine learning perspective, we examine feature selection methods that work with both uncorrelated and correlated feature spaces under high-dimensional settings with broad applications in many fields.

The thesis is organized as follows. In Chapters 2 and 3, we review relevant background knowledge of immunology and feature selection with Lasso and its generalizations. Chapter 4 presents fast dynamic programming algorithms by solving an ℓ_1-regularized problem to extract fixed-length substrings from sequencing data with string kernel and Fisher kernel features. We further present an extension of the dynamic programming algorithms to select subsequences of varying lengths in Chapter 5. An application to identify short subsequences from mouse CDR3β sequences are presented in Chapters 4 and 5. In Chapter 6, we examine feature selection from correlated data with exclusive group Lasso and present novel efficient optimization algorithms. We also show novel pipelines on group allocation and model selection for stable feature selection. The proposed methods are validated on extensive experiments on synthetic data with various correlation settings, as well as real-world data to extract entire TCRβ sequences from human samples. Chapter 7 concludes our findings and contributions, with possible directions for future research.

Chapter 2

The immune system

The immune system is a sophisticated, complex network involving many different cells and tissues that defend the body against attacks by various external invaders, including bacteria, viruses, fungi, and parasites, as well as internal threats such as tumor cells and cellular debris. In vertebrates, the immune system consists of two major subsystems: the innate immune system and the adaptive immune system. The innate immune system provides immediate, broad defense against pathogens, while the adaptive immune system creates immunological memory and offers antigen-specific immune responses. Despite functional differences, the innate immune system and the adaptive system together provide fast and solid protection against pathogenic infections.

The innate immune system is an ancient defense strategy that is found in both vertebrates and invertebrates, as well as plants [1, 2]. It provides a fast and powerful defense against a variety of harmful foreign particles and cellular debris. It consists of the physical barriers, secretions, and broad immune responses. The innate immune system can detect common pathogens through pattern recognition receptors (PRRs) that are mainly expressed by innate immune cells including dendritic cells and macrophages [3–5]. PRRs can also be found on B cells and T cells of the adaptive immune system, as well as some non-immune cells. Toll-like receptors (TLRs) are a subset of PRRs and can recognize various microbes. The members of the innate immune system to defend against foreign invaders include professional phagocytes, complement proteins, and other components such as natural killer cells.

Professional phagocytes, such as macrophages, neutrophils, and dendritic cells, can ingest pathogens and cellular debris through phagocytosis. Component proteins create holes on the cell membrane of invaders. Natural killer cells are a type of cytotoxic lymphocyte and provide fast immune response to destroy virus-infected cells. The members of the innate immune system also participate in the adaptive immune response. For example, macrophages and dendritic cells present antigen fragments on the cell surface for T cells, while complement proteins are recruited by antibodies to assist humoral immunity.

The adaptive immune system is a highly specialized subsystem that adapts to provide specific immune responses to various invaders. It creates immunological memory after the initial response to an antigen and evokes a faster and enhanced response to subsequent infections by the same antigen. The adaptive immune system consists of humoral immunity that involves molecules in extracellular fluids such as B cells and antibodies, as well as cell-mediated immunity that involves the participation of antigen-specific T cells and antigen-presenting cells.

In this chapter, we mainly focus on cell-mediated immunity, which is most relevant to this thesis. Appendix A provides a comprehensive review of the innate immune system and humoral immunity.

2.1 Cell-mediated immunity

In contrast to humoral immunity, antibodies do not take part in cellular immunity. Cell-mediated immunity involves various cytokines, phagocytes, and antigen-specific T cells. T cells originate from lymphoid progenitor cells in bone marrow, migrate to the thymus and develop into helper T cells (CD4+ cells) and killer T cells (cytotoxic T cells or CD8+ T cells). T cells cannot interact with pathogenic invaders directly but instead, bind to antigens presented by class I and class II MHC molecules on antigen-presenting cells.

2.1.1 Antigen presentation

Class I MHC molecules

On chromosome 6, humans have three genes for class I MHC proteins: HLA-A,

HLA-B, and HLA-C. The complete class I MHC molecule consists of the three HLA proteins, as well as β2-microglobulin. Class I HLA proteins can be encoded by various polymorphic forms of genes in the human population, and this ensures class I MHC molecules can interact with a diverse range of antigens.

Class I MHC molecules focus on endogenous proteins. They present peptides that are degraded from cytosolic proteins and show what is happening inside a cell to killer T cells. These peptides are transported to the endoplasmic reticulum with the help of transporter proteins such as transporter associated with antigen processing 1 and 2 (TAP1 and TAP2). A peptide that binds to a class I MHC molecule must fit its amino acids at the end of the binding groove of the MHC molecule, and this constraint limits the length of peptides to which a class I MHC molecule can bind (approximately 8-10 amino acids) [6]. If a peptide fits into the class I MHC groove, the MHC I-peptide complex moves to the cell surface to be scanned by killer T cells. Peptides that are not compatible with class I MHC molecules are transported back to the cytoplasm and broken down into individual amino acids to make new proteins. Notably, the proteasome can be modified by other proteins to cleave proteins after hydrophobic or basic amino acids in the sequence, while hydrophobic or basic C-termini have higher binding affinity to TAP transporter proteins and class I MHC molecules. In addition, TAP transporter proteins are more likely to transport peptides with lengths of 8 to 16 amino acids [7]. Such properties ensure higher efficiency of antigen presentation by class I MHC molecules.

Class II MHC molecules

Similar to class I MHC molecules, humans have three highly polymorphic HLA genes to encode class II MHC molecules on chromosome 6: HLA-DP, HLA-DQ, and HLA-DR. Unlike class I MHC molecules, which have rather strict constraints regarding binding peptide lengths, class II MHC molecules can bind to peptides that are longer than the binding groove as the peptide can bind to both ends of the groove, and therefore class II MHC molecules can bind to longer peptides (approximately 13 to 25 amino acids) [6].

Before binding to any other peptides, the groove of a class II MHC molecule

is filled by a protein called the invariant chain. The invariant chain prevents class II MHC molecules from binding to other peptides in the endoplasm reticulum, particularly the peptides that potentially bind to class I MHC molecules. In addition, the invariant chain can direct class II MHC molecules through the Golgi stack to endosomes, which then merge with phagosomes that contain pathogenic invaders engulfed by phagocytosis. At the same time, the invariant chain is destroyed by enzymes in endosomes, except for the portion (called CLIP) that actually sits in MHC groove. CLIP is then catalyzed by a protein named HLA-DM to vacate class II MHC molecules so that class II MHC molecules can bind to peptides from pathogens in endosomes. MHC II-peptide complex then travels to the cell surface to be displayed to helper T cells.

The major differences between class I and class II MHC molecules are summarized in Table 2.1

Table 2.1: Differences between class I and class II MHC molecules.

Class I MHC	Class II MHC
Expressed by most cells	Expressed by a limited set of cells
Present to killer T cells	Present to helper T cells
Display endogenous proteins	Display exogenous proteins
Encoded by HLA-A, HLA-B, HLA-C	Encoded by HLA-DP, HLA-DQ, HLA-DR
Bind to shorter antigens	Bind to longer antigens

Antigen-presenting cells

Dendritic cells Dendritic cells are perhaps the most important antigen-presenting cells as they can activate naive T cells and initiate the immune response. Dendritic cells can be activated by various signals such as TNF from macrophages and neutrophils, chemicals from cells that are killed by the immune system, and signals from PRRs. After activation, a dendritic cell remains in tissues for approximately six hours, collecting samples of invading antigens before it circulates to lymph nodes. In this process, a dendritic cell upregulates the expression of class I MHC molecules if infected by invading antigens. Meanwhile, it also releases class II MHC molecules and binds to antigens from the infection site, as well as produc-

ing more B7 co-stimulatory proteins. A dendritic cell can live approximately one week in lymph nodes, and this ensures that antigens presented by the dendritic cell are up-to-date. The more severe the pathogen attack, the more dendritic cells are involved, activating more naive T cells and triggering a more serious response to the invasion. It is notable that dendritic cells cannot kill invaders directly, but rather induce other components in the immune system to eliminate the invading microbe.

Activated macrophages Macrophages can be activated by cytokines like IFN-γ or TLRs (see Section A.1.2.1). Unlike DCs, macrophages do not activate naive T cells, but rather provide continuous co-stimulatory signals for experienced T cells at the infection site to avoid early termination of T-cell responses.

Activated B cells Activated B cells can act as antigen presenting cells to display naive and undigested antigens. When B-cell receptors (BCRs) bind to the cognate antigens, the BCR-antigen complex is removed from the B cell surface and transported into B cells, where antigen is processed and loaded onto class II MHC molecules for antigen presentation on the B cell surface. Compared to other antigen-presenting cells, activated B cells provide up to 10000-fold advantage in activating T cells because of the high affinity for antigens [1]. Therefore, even with low levels of antigens circulating, activated B cells can activate T cells rapidly (less than half an hour).

To summarize, dendritic cells initialize T-cell immune response by activating naive T cells, activated macrophages provide continuous stimulation to T cells to avoid early termination of the immune response, and activated B cells can quickly activate helper T cells when the same antigen attacks the host body again.

2.1.2 T cells and T-cell receptors

Subtypes of T-cell receptors

T-cell receptors (TCRs) are molecules on T cell surfaces that actually bind to antigens presented by class I and class II MHC molecules on antigen-presenting cells. Each T-cell receptor is a transmembrane heterodimer with two chains. In humans, on the surface of 95% of all T cells, TCRs are composed of an α chain and a β chain with CD4 or CD8 molecules. $\alpha\beta$ TCRs are very diverse and recognize a huge

variety of antigens. Approximately 5% of human T cells express $\gamma\delta$ chains on their TCRs, and lack CD4 or CD8 molecules. $\gamma\delta$ TCRs are generally less diverse and are most abundant in the mucosa of tongue, intestine, and uterus [8, 9]. Although $\alpha\beta$ TCRs have been widely studied for a long time, much less is known about $\gamma\delta$ TCRs. It is believed that $\gamma\delta$ TCRs mostly focus on unpresented antigens [10]. Table 2.2 shows the differences between $\alpha\beta$ and $\gamma\delta$ TCRs. In this section, we only introduce $\alpha\beta$ TCRs as the majority of our work in this thesis focuses on $\alpha\beta$ TCRs.

Table 2.2: Differences between $\alpha\beta$ and $\gamma\delta$ TCRs.

$\alpha\beta$ TCR	$\gamma\delta$ TCR
Expressed on 95% human T cells	Expressed on 5% human T cells
Express CD4 and CD8	Not express CD4 or CD8
Very diverse	Less diverse
Found in many areas	Mostly found in gut mucosa
Recognize presented antigen	Recognize unpresented antigen
Widely-studied	Still mysterious

V(D)J recombination

Human antibody consists of a heavy chain and a light chain, each with constant and variable domains. Generic loci that encode the variable domains contain multiple copies of three types of gene segments: variable (V) segments, diversity (D) segments (only heavy chains have D regions), and joining (J) segments. Similarly, TCR genes have multiple V and J gene segments on α chains and V, D, and J gene segments on β chains. The diversity of antibodies or BCRs and TCRs come from somatic recombination of DNA within V, D, and J regions, which ultimately results in diverse antigen-binding regions that are capable of recognizing a enormous number of pathogens.

TCR chains undergo the V(D)J rearrangement process in the thymocyte development. Figure 2.1 shows the process of V(D)J recombination of TCR α and β chains. One D gene and one J gene segment from TCR β chain are selected and recombined randomly, followed by the joining of a V gene, and form a complete gene on the variable region. While processing RNA transcript, the constant region with separate exons join the complex of VDJ segments by splicing out the introns.

The rearrangement of the TCR β chain is followed by TCR α chain rearrangement. Since α chains do not have D gene segments, the rearrangement in α chains first occurs between V and J gene segments, followed by the joining of C gene segments. V(D)J rearrangement in B cells is very similar to the process in T cells—the heavy chains are rearranged in a similar way as TCR β chains while the rearrangement of the light chains are similar to TCR α chains.

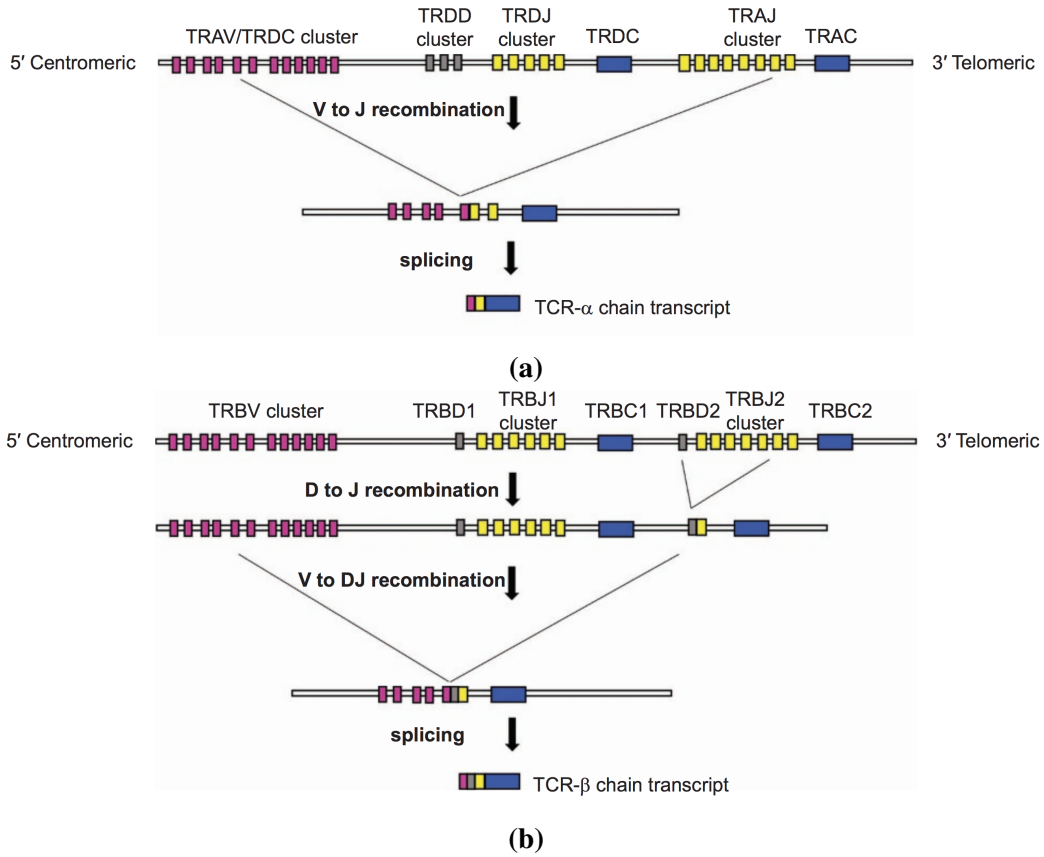


Figure 2.1: Human TCR V(D)J recombination [11]. (a). α chain; (b). β chain.

Complementarity-determining regions

The variable domains in antibodies and TCRs bind to specific antigen epitopes with the complementarity-determining regions (CDRs). There are three types of CDRs on each chain of antibodies and TCRs: CDR1, CDR2, and CDR3. CDR1 and CDR2 are found in the V region within the variable domain, while CDR3 is made up of the D and J region, together with the end of the V region, and any junctional sequences added during recombination between these genes (Figure 2.2a).

Crystallography showed that CDR loops (Figure 2.2b) interact with MHC-peptide complex: CDR1 and CDR2 amino acids are generally responsible for interaction with MHC molecules, providing rigid docking of TCR and MHC-peptide binding; antigenic peptides are mainly recognized by the amino acid residues on CDR3s. Figure 2.3 shows a structural view of TCR binding to class I MHC molecule via its CDR3 region. Consequently, sequence diversity on $\alpha\beta$TCRs is not uniformly distributed along CDRs but is rather skewed towards two CDR3s [12, 13]. This unevenly distributed diversity indicates that CDR3 sequences are the most variable and diverse part on TCRs and determine the high antigen specificity of T-cell responses.

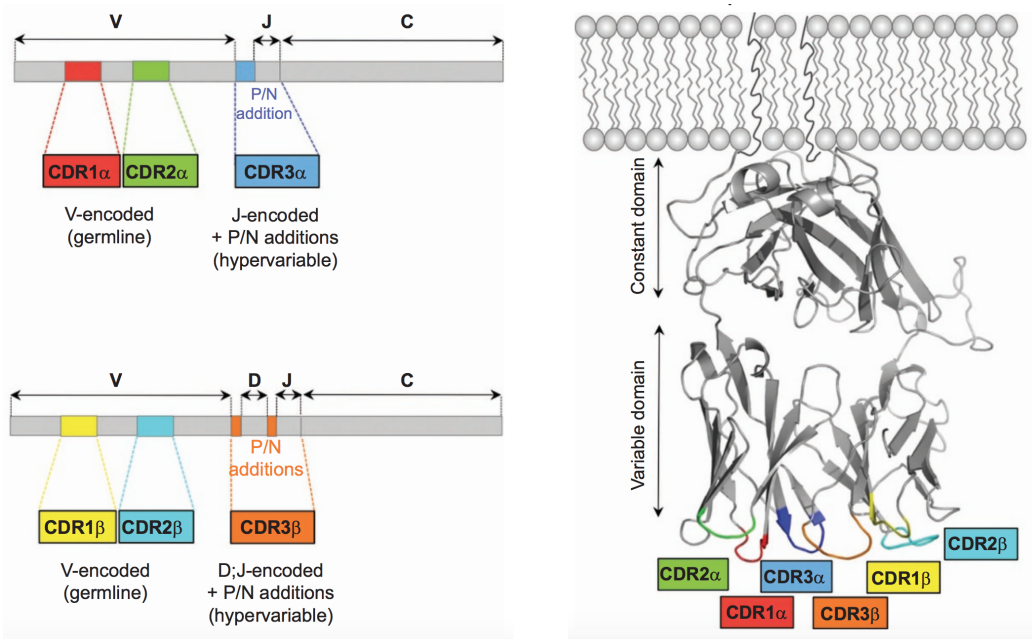
Unlike antibodies and $\gamma\delta$ TCRs which have varying CDR3s lengths, CDR3s on $\alpha\beta$ TCRs have most constrained lengths: lengths of CDR3α and CDR3β sequences have very similar distributions in both humans and mice [14]. Although TCRα sequences do not have D genes, the D gene usage in CDR3β is compensated by more J genes in CDR3α. Despite the constrained lengths, CDR3s on $\alpha\beta$ TCR are highly variable. Previous studies suggested an estimation of 10^6 or even more unique TCRβ CDR3 sequences in an individual [15, 16].

T cell activation

α and β chains of TCR are extracellular proteins anchored on the T cell surface with very short cytoplasmic tails (approximately 3 amino acids). In order to transduce a signal when antigen binds, TCR works closely with a complex of CD3 proteins. Human CD3 proteins consist of four proteins: γ, δ, ϵ and ζ. As opposed to α and β chains, most CD3 proteins are inside a T cell, with a short portion outside the cell. The cytoplasmic tails of CD3 proteins use a complex kinase cascade to deliver the activation signal from TCR α and β chains to the T cell nucleus. Figure 2.4a shows the structure of a TCR complex.

The signals sent by CD3 proteins result in different outcomes: when TCRs recognize self peptides loaded on MHC molecules, TCRs can cause T cell apoptosis to avoid autoimmunity; when TCRs bind to the cognate antigens on MHC molecules, and when TCRs receive co-stimulatory signals, T cell is activated.

In addition to the $\alpha\beta$ chains of TCRs and CD3 proteins, $\alpha\beta$ T cells also ex-



(a) **(b)**

Figure 2.2: Structure of CDRs on TCR [11]. (a). CDR1 and CDR2 are encoded by V genes, while CDR3 consists of V and J gene segments (TCRα) or V, D, and J gene segments (TCRβ); (b). Protein structure of TCR with CDR1, CDR2, and CDR3.

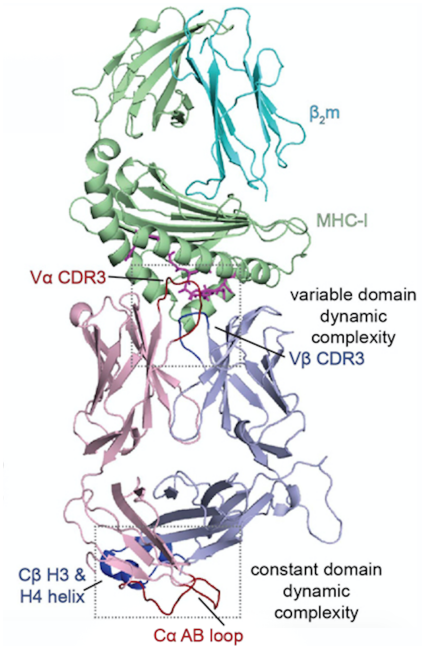
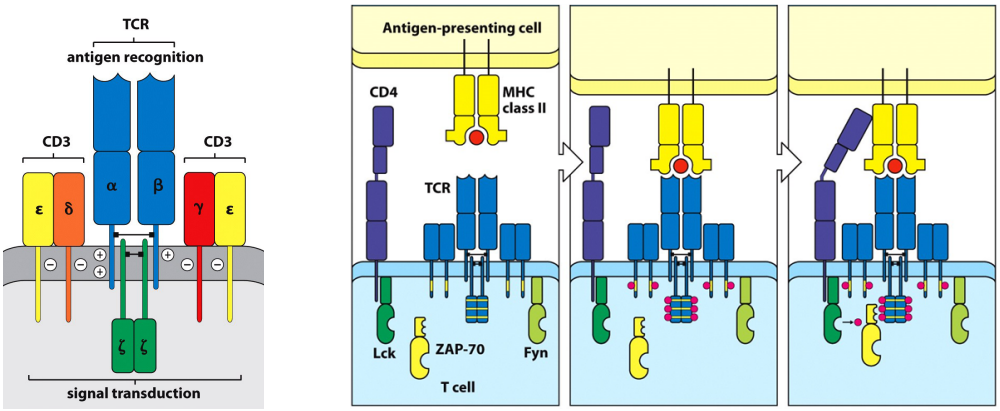


Figure 2.3: Structural view of CDR3 interacting with class I MHC molecule [17].

press CD4 and CD8 co-receptors. Usually, helper T cells express CD4, while killer T cells express CD8. The expression of CD4 and CD8 ensures that killer T cells

bind to class I MHC molecules and helper T cells bind to class II MHC molecules. Furthermore, the binding strength between TCRs and MHC molecules is very weak, while CD4 and CD8 can stabilize such interactions. Figure 2.4b shows how TCR binds to class II MHC molecules as well as how CD4 secures the interaction between TCR and MHC molecule. A similar process is observed for the interaction between TCRs on a killer T cell and class I MHC molecules.



(a) **(b)**

Figure 2.4: TCR complex [18]. (a). Structure of a TCR complex; (b). TCR binds to class II MHC molecule.

T cells require co-stimulatory signals to be activated. The best-studied co-stimulatory protein on antigen-presenting cells is B7, which binds protein CD28 on the surfaces of T cells. Similar to B cells, co-stimulation can amplify the signals from TCRs to the nucleus, so that approximately 100 times fewer TCRs are required to bind to antigens in order to activate T cells. Once naive T cells have been activated, the connection between TCRs and the T cell nucleus becomes strengthened. Therefore, co-stimulation in experienced T cells is not as important as in naive T cells.

Helper T cells and killer T cells activate by slightly different mechanisms. When a naive helper T cell discovers a dendritic cell that displays its cognate antigen, it binds to the class II MHC molecules on this dendritic cell. This binding strength is usually very weak. CD4 molecules then bind to MHC molecules to increase the binding strength. Meanwhile, the T cell upregulates the expression of adhesion molecules on its surface to further increase the binding affinity between T

cell and dendritic cell.

After binding to antigens presented by class I MHC molecules on an activated dendritic cell, CD8 receptors on killer T cells interact with class I MHC molecules to stabilize the binding strength. Analogous to the helper T cell, CD28 molecules on the killer T cell bind to CD80 (B7-1) or CD86 (B7-2) molecules on the antigen-presenting cells to provide co-stimulation signal to finish the activation process. However, the second signal received by the killer T cell can be assisted by cytokines produced by helper T cells. In fact, assistance from helper T cells is necessary to activate long-lived killer T cells, create memory killer T cells, as well as improve the efficiency of the killer T cells.

Functions of helper T cells and killer T cells

As the antigen-presenting cell that initializes T-cell immune response, dendritic cells provide information about the type and location of the infection to naive T cells. The PRRs on dendritic cells recognize various types of pathogens and result in the production of different cytokines, see Section A.1.1 for details. Therefore, cytokine receptors and PRRs on dendritic cells deliver infection type to T cells. Cells from different parts of the body generate characteristic cytokine mixtures, which help to convey location information to T cells. As a result, different mixtures of cytokines induce different T cells into action.

Type 1 helper T cells (Th1 cells) Th1 cytokines are designed to increase cell-mediated immune response to defend against intracellular bacterial or viral attacks. Under such attacks, dendritic cells produce interleukin-12 (IL-12), which stimulates helper T cells to produce TNF, IFN-γ, and IL-2 [19]. TNF helps to activate macrophages and NK cells to kill the invaders. IFN-γ provides continuous activation to macrophages and affects class switching of B cells to produce more IgG3 antibodies, which further promote opsonization of invading viruses and bacteria. IL-2 is a growth factor that increases the proliferation rate of many cells including killer T cells, NK cells, and Th1 cells.

Type 2 helper T cells (Th2 cells) Th2 cells help to increase the antibody-mediated immune response to extracellular parasitic and bacterial infection via the

digestive tract. Triggered by IL-2 and IL-4, Th2 cells increase the production of IL-4, IL-5, and IL-13. IL-4 is a growth factor for B cells and triggers B cells to produce more IgE antibodies during class switching, while IgE is a powerful weapon against parasites. IL-5 triggers B cells to generate IgA antibodies, which focus on destroying bacteria from the digestive tract. IL-5 also activates eosinophils to attack parasites such as helminths. IL-13 is a central regulator in mucus hypersecretion and prompts the production of mucus in the digestive tract [20].

Type 17 helper T cells (Th17 cells) Th17 cells are important in the immune response against fungal or certain extracellular bacterial pathogens. Dendritic cells produce TGFβ and IL-6 to stimulate Th17 cells to produce IL-17 and IL-21, which in turn summon neutrophils and promote uncommitted Th cells into Th17 cells [21]. Both IL-17 and IL-21 influence B cells to produce antibodies that opsonize fungi and bacteria, as well as activate the complement system.

Killer T cells Killer T cells directly interact with the target cells to cause cell apoptosis. Killer T cells produce perforin, the same cytolytic protein found in NK cells and similar to the C9 protein in MAC. Perforin binds to cell membranes, forms pores, and enters the target cell with vesicles made from the target cell membrane. The vesicles contain both perforin and granzyme B. Once inside the target cell, perforin produces holes on the vesicle and releases granzyme B, which further promotes enzymatic chain reactions to trigger cell apoptosis. In addition, Fas ligand, a transmembrane TNF protein on the surface of killer T cells, can bind to Fas on the target cell and induce cell apoptosis.

2.2 Immunological memory

Both the innate immune and adaptive immune systems have immunological memory. The innate immune system has evolved to recognize a wide range of invaders. For example, TLR can bind to LPS which is found on the surfaces of Gram-negative bacteria. Some cells such as NK cells can respond faster in a subsequent attack by a pathogen that has previously attacked the host body [22]. The memory rooted in the innate immune system manifests as broad responses to long-existing common

invaders and does not adapt to the molecular shapes of specific antigens.

The immunological memory of the adaptive immune system, in contrast, is specific and adapts to particular pathogens. Theoretically, the diversity of B cells and T cells ensures that humans can respond to any pathogen, but the number of naive B cells or T cells corresponding to a particular pathogen is too limited to react promptly to pathogenic infections. When naive B cells and T cells are activated to proliferate during the response to an initial attack from an antigen, most of them die by apoptosis after the attack, and some remain as memory cells. Memory cells are generally in greater number and require significantly less MHC-peptide concentration in their activation than naive cells. Co-stimulation is sometimes not necessary to activate memory cells. Therefore, in subsequent immune responses to the same invader, memory cells quickly recognize the cognate antigen and proliferate to provide a faster defense.

B cells and T cells exhibit different behaviors after the initial response. Activated B cells can generate three types of B cells: short-lived plasma cells, long-lived plasma cells, and central memory B cells. Short-lived plasma cells produce a large number of antibodies during the immune response, but have a short life of several days. Long-lived plasma cells produce a modest number of antibodies constantly, providing long-term protection against the cognate antigens. Central memory B cells slowly proliferate and replace long-lived plasma cells, as well as produce more short-lived plasma cells during a pathogenic attack.

Some of the activated T cells become effector T cells and are transported to the infection area during an immune response. Most of the effector T cells die from apoptosis after the invader is eliminated, while a small amount of them become memory effector T cells. Memory effector T cells remain in the infectious area and can respond promptly if exposed to their counter-specific antigens again. Other activated T cells become central memory T cells and mostly remain in lymph nodes. Central memory T cells are activated and mature into effector T cells in subsequent immune responses.

2.3 Immune responses to viruses

Viruses are ultimate parasites with the ability to mutate and evolve. Many viruses are enveloped viruses, which have an additional membrane outside the protein coat enclosing viral DNA or RNA. This membrane structure allows viruses to fuse with the host cell's membrane directly or to be engulfed by endocytosis or similar processes. Once having entered a cell, viruses hijack the infected cells to produce viral protein to replicate virus genetic information and produce identical copies of the invading virus. When a new viron, a single virus particle, exits the host cell, it is wrapped by the host cell's membrane to form a new envelope and can start a new cycle of infection and replication in an uninfected cell with its cellular-membrane envelope.

2.3.1 Innate and B-cell responses

The innate immune system provides early protection against viruses and sends out danger signals to alert the members of the adaptive immune system. PRRs recognize microbes such as viruses and activate different pathways to induce immune responses to defense against viral infections. B-cell response to viral infections also plays a central role in defense against cytopathic viruses via antibodies. Secreted antibodies have the ability to neutralize viruses and provide pathogen-specific long-term antiviral protection.

2.3.2 T-cell responses

Rather than neutralizing intact virus particles, T-cell responses to viruses are restricted to the physical presence of MHC molecules as well as T cells, and only recognize short peptides that are degraded from viruses or virus-infected cells. As introduced in Section 2.1.2, killer T cells and helper T cells are activated by different pathways that involve class I and class II MHC molecules, respectively.

Class I MHC molecules are expressed on almost all types of cells and mainly present endogenous proteins that are synthesized in the intracellular environment. Viruses hijack the infected cells to replicate viral genetic information and proteins, which can be degraded and displayed by class I MHC molecules. In addition,

extracellular viral particles enter antigen-presenting cells via endocytosis and are degraded into short peptide fragments through proteolysis. The peptides are then loaded to class I MHC molecules and transported to the cell surface for presentation to killer T cells. The mechanism of presenting extracellular viral proteins is cross-presentation, which activates killer T cells by antigen-presenting cells that are uninfected by viruses.

Unlike class I MHC molecules, class II MHC molecules only express on professional antigen-presenting cells. The class II MHC pathway mainly presents exogenous viral peptides or remnants from cells infected by viruses. The exogenous antigens are engulfed via phagocytosis and are loaded on class II MHC molecules to traffic to the cell surface for presentation to helper T cells.

During viral infections, one of the most pronounced antigen-presenting cells is the dendritic cell. Dendritic cells are critical for priming naive T cells. Inactivated dendritic cells reside at initial sites of viral infection and, once activated, migrate to nearby lymph nodes, where they encounter and activate naive T cells. Activated T cells rapidly proliferate and differentiate to release inflammatory mediators such as interferons and trigger other antiviral responses. Dendritic cells are very prone to virus infection. The infected dendritic cells can activate killer T cells via class I MHC molecules, while the uninfected dendritic cells are capable of activating both killer T cells and helper T cells, through cross-presentation and the classical class II MHC pathway, respectively.

The immune response of killer T cells to viruses include the synthesis of inflammatory mediators and cytotoxic effector molecules. During the initial phase of viral infections, killer T cells often proliferate significantly. For example, experimental results show that during the first week upon the infection of lymphocytic choriomeningitis viruses (LCMV), LCMV-specific killer T cells increase by more than 10,000-fold in B cell-deficient mice [23]. During the expansion phase, killer T cells increase the production of both inflammatory cytokines and cytotoxic effector molecules, as well as surface proteins such as cytokine receptors and adhesion molecules. Activated killer T cells release granzymes and perforin to kill the virus-

infected cells before the release of progeny viruses. They also generate cytokines and chemokines such as IFN-γ and TNF to mediate the immune responses of other components in the immune system.

Helper T cells assist other members in the immune system, including activating and engaging in cells from the innate immune system such as macrophages and inducing antibody production and class switching of B cells, as well as helping killer T cells. Although the main function of helper T cells is to provide help and to involve in the immune responses from other cells, T cells also offer direct antiviral responses, mostly through the production of IFN-γ and the induction of lysis of the infected cells. The most crucial type of helper T cells is Th1 cells, which produce IFN-γ against many viral and bacterial infections. Other types of helper T cells, such as Th2 and Th17 cells, are not significantly involved in antiviral responses, as introduced in Section 2.1.2.

2.4 Motivation and related work

The recent development of next-generation sequencing (NGS) techniques has opened up robust approaches for deep sequencing the TCR repertoire, allowing massive parallel sequencing of the entire genome and generating ultra-high dimensional sequencing data. This provides opportunities to conduct studies for the immune repertoire at relatively low costs.

The TCR repertoire, or TCR profile, is defined by all unique TCRs of the cell-mediated adaptive immune system of an individual. Clonal selection theory [24] suggests that the adaptive immune response is encoded by the increased frequency of antigen-specific TCRs on T-cell surfaces. Analyzing the TCR repertoire is a crucial test of the clonal selection theory. The size of the TCR repertoire is also an important factor in cancer treatment in which immune cells such as white blood cells decrease as a side effect of chemotherapy. The TCR repertoire is updated with the progress of diseases and can change greatly in response to different infectious antigens. Identification of the counter-specific antigens results from the highly diverse and divergent TCR repertoire, which is generated by the somatic V(D)J re-

combination.

The process of V(D)J recombination ensures the diversity of the TCR repertoire in response to various infections, while the enormous diversity poses great challenges in analyzing the TCR repertoire. In healthy human adults, there are approximately 4×10^{11} T cells [25, 26]. The TRA gene encoding TCR α chain contains 47 V genes, 57 J genes, and 1 C gene. The TRB gene encoding TCR β chain contains 54 V genes, 2 D genes, 13 J genes, and 2 C genes. This results in unique gene combinations of 2,679 α chains, and 2,808 β chains, which leads to 7.5 million $\alpha\beta$ pairs [27]. Similarly, V(D)J recombination produces diverse combinations of TCRγ and δ chains. Nucleotide insertions and deletions at junctions, such as regions between V-J, V-D, or D-J, further result in 10^{15}–10^{20} unique TCRs in theory [28–31]. The actual clonotypes presented in humans beings are at least 100 times less than the number of T cells, with an estimated number of $\alpha\beta$ TCRs ranging from 2.5×10^7 to a potential upper bound of 10^8–10^{11} [32, 33]. Therefore, there is a low likelihood of observing common TCRs that occur in many individuals in the population if all TCRs are generated at an equal probability. Indeed, the majority of TCR repertoires from different individuals are not overlapped and most TCRs are private. Private TCRs are the TCRs that are only found in one individual [34, 35], while public TCRs are those that are found in multiple individuals. Despite the large proportion of private TCRs, a substantial degree of public TCRs have been observed in many types of immune response in various species [36–40]. [41] suggested that 18%–27% CDR3s were shared between any pair of mice. [42] observed 30% public CDR3s in mice, with an average of approximately 10% CDR3s being shared between any two mice. A relatively high proportion of public TCRs have also been observed in humans. The probability of observing the same TCR in two individuals was estimated to be 1,000-fold higher than if all V(D)J recombinations occurred at an equal probability [43]. [44] found an average of approximately 16% and 38% overlapping between any two individuals in their naive and memory T cells, respectively.

Public TCRs indicate that TCRs are not generated with equal probability un-

der V(D)J recombination. The possible mechanism of generating public TCRs includes bias towards usage and combination of V, D, and J genes [45, 46], as well as convergence, where multiple recombinations lead to the same nucleotide sequence and multiple nucleotide sequences lead to the same amino acid sequences for TCRs [47, 48], in the process of V(D)J recombination.

Public TCRs have been shown to play an important role in public T-cell responses to many viruses, such as influenza virus [49, 50], cytomegalovirus (CMV) [51–53], Epstein-Barr virus (EBV) [54–57], human immunodeficiency viruses (HIV) [58, 59], and simian immunodeficiency virus (SIV) [60, 61]. Public TCRs were also discovered in autoimmune diseases [62, 63] and alloreactive disease [64–66]. Therefore, understanding public TCRs is crucial for designing immunotherapy drugs and developing vaccines.

Despite the high diversity of the TCR repertoire, only a small number of TCRs respond to a particular antigen and are therefore relevant to this antigen. Given a particular antigen, its antigen-specific TCRs bind to the same pathogenic peptides and often act together in the adaptive immune response. Therefore, a reasonable assumption is that these TCRs may share a degree of similarity or correlation. This has been observed in several studies. [39] discovered correlation and co-occurrence between TCRs collected from CMV patients. The similarity between CDR3s, the most diverse region in TCRs, was also shown across different mice [42]. However, just as multiple TCRs can recognize the same antigen, the same TCR can recognize multiple antigens, which perhaps leads to unclear structures of correlated TCRs. For example, a TCR recognizing one antigen may be able to identify a different antigen, while this antigen may not be recognized by another TCR responding to the former antigen. This poses the main challenge in identifying a small set of antigen-specific TCRs from high-dimensional spaces with correlation.

One possible approach to analyze the antigen-recognition of TCRs is to break entire sequences into short motifs. Amino acid sequences of proteins have been used to predict the 3D structure of folded proteins and protein-protein interactions [67], as well as the binding affinity between antigen and antibodies [68]. Short

sequencing reads were used to estimate the sharing of TCR β sequences [46]. As CDR3s interact with MHC molecules and bind with pathogenic peptides, many studies have investigated the role and function of CDR3 in the adaptive immune response. CDR3 sequences alone were used to learn TCR-antigen interaction as well as in predicting immunization [69, 70]. Furthermore, previous studies showed that instead of entire CDR3 sequences, short amino acid subsequences along the CDR3 region are responsible for antigen binding and can be used to predict the immunization of the T-cell repertoire [71, 72]. From a machine learning perspective, the use of such short subsequences usually leads to a much smaller feature space, as well as lower noise raised from irrelevant entire TCRs. This motivated the examination of the global changes of the T-cell repertoire in response to specific antigens with short amino acid motifs on the highly diverse CDR3 in Chapters 4 and 5. However, analyzing and identifying public TCRs require efficient algorithms capable of working with entire TCR sequences, which creates ultra-high dimensional spaces with correlated features. Therefore, to discover important public TCRs from high-dimensional feature spaces, we further explored group-based approaches to cope with correlated features with unknown and unclear group structure in Chapter 6.

Chapter 3

Sparsity learning with Lasso and its generalizations

3.1 Overview

Feature selection is one of the most important tasks in machine learning. In recent years, high-dimensional datasets are increasingly being collected in many fields. Many biological datasets contain tens of thousands of genes or millions of TCRs, while the sample size is usually no more than a few hundred. From a machine learning perspective, when sample size is much smaller than feature size, overfitting is likely to occur and the model becomes unstable; while from a biological viewpoint, it is often of the same importance to identify a subset of predictive features as to achieve high prediction accuracy. In the context of immunology, the cell-mediated immune response is encoded by the increased frequency of antigen-specific T cells upon exposure to the cognate antigens. Therefore, knowledge of a subset of TCRs is beneficial to our understanding of the mechanism of the adaptive immune response and to design more effective strategies for diagnosis, treatment, and prognosis for many diseases.

In this chapter, we introduce sparsity-endorsing algorithms with an emphasis on Lasso and its generalizations. In Section 3.2, we start with non-sparse methods including least squares and ridge regression, followed by Lasso and ℓ_1-regularized SVM. We then focus on generalizations of Lasso which are designed for various

types of problems in Section 3.3. Section 3.4 reviews the sign consistency of Lasso and elastic net. Stability selection is introduced in Section 3.5. A brief survey of related work of Lasso and other methods is given in Section 3.6.

Notation Throughout this thesis, we adopt notations as follows to avoid mathematical clutter. Matrices are denoted by upper case, non-bold letters. Vectors are column vectors by default. All vectors are represented by lower case, bold letters to avoid ambiguity with scalar variables. We denote by $\mathbf{y} \in \mathbb{R}^m$ the input labels and by $\mathbf{w} \in \mathbb{R}^n$ the weight vector. We let $\mathbf{w}^*$ and $\hat{\mathbf{w}}$ denote the true and the estimated weights, respectively. We let X represent an $m \times n$ matrix of input data, X_j denote the j-th feature, and $\mathbf{x}_{(i)}$ denote the i-th sample. The term "relevant features" represents features with nonzero weights such that $\{X_j : \mathbf{w}_j^* \neq 0\}$, and "irrelevant features" represents features with zero weights where $\{X_j : \mathbf{w}_j^* = 0\}$.

3.2 Least squares, ridge regression, Lasso, and support vector machine

3.2.1 Least squares

Ordinary least squares (OLS) is perhaps the most naive model for linear regression problems, where we assume continuous labels $\mathbf{y}$ is computed by the linear combination between input data X and weight vector $\mathbf{w}$, such that

$$\mathbf{y} = X\mathbf{w} + \epsilon \tag{3.1}$$

where $X \in \mathbb{R}^{m \times n}, \mathbf{y}, \epsilon \in \mathbb{R}^m$, and $\mathbf{w} \in \mathbb{R}^n$. ϵ is a random noise vector. Throughout this thesis, we assume ϵ follows a Gaussian distribution with zero mean and some variance such that $\epsilon_i \sim \mathcal{N}(0, \sigma^2), i = 1, 2, \cdots, m$.

The method of least squares was first explained concisely by [73] in 1805. As its name indicates, least squares finds the solution that minimizes the sum of the squared error:

$$\min_{\mathbf{w}} ||\mathbf{y} - X\mathbf{w}||_2^2 \tag{3.2}$$

When the noise ϵ follows a Gaussian distribution, the solution to (3.2) is equivalent to the maximum likelihood estimator, which provides an unbiased solution. OLS has a closed-form solution

$$\hat{\mathbf{w}} = (X^T X)^{-1} X^T \mathbf{y} \tag{3.3}$$

This is a unique solution of (3.2) if $\text{rank}(X) = n$ when the feature columns of X are linearly independent. The in-sample risk of least squares [74] is given by

$$\frac{1}{m}\mathbb{E}||X\hat{\mathbf{w}} - X\mathbf{w}^*||_2^2 = \frac{\sigma^2 n}{m} \tag{3.4}$$

The out-of-sample risk is always larger than the in-sample risk bound. When the distribution of X is fixed, for example, $X \sim \mathcal{N}(0, \Sigma)$, the exact out-of-sample risk for a new data point $\mathbf{x}$ is computed by

$$\mathbb{E}||\mathbf{x}^T\hat{\mathbf{w}} - \mathbf{x}^T\mathbf{w}^*||_2^2 = \frac{\sigma^2 n}{m - n - 1} \tag{3.5}$$

In high-dimensional settings where X is rank deficient, the optimal solution $\mathbf{w}^*$ to problem (3.2) is not unique, although the fitted values $X\mathbf{w}^*$ are always unique. This is to say that there exist infinite solutions $\hat{\mathbf{w}}$ to problem (3.2) that give the same values for $X\mathbf{w}^*$. In this case, it is not possible to make predictions for new data points, as well as interpreting the estimated model. Furthermore, the in-sample risk can be very poor if dimension n exceeds sample size m. Regularization is therefore introduced to solve this problem by reducing the variance at the cost of introducing additional bias.

3.2.2 Ridge regression

One regularization method is ridge regression [75–78], which is defined as

$$\begin{aligned} \min_{\mathbf{w}} \quad & ||\mathbf{y} - X\mathbf{w}||_2^2 \\ \text{subject to} \quad & ||\mathbf{w}||_2^2 \leq t \end{aligned} \tag{3.6}$$

Problem (3.6) is often rewritten in its Lagrangian form

$$\min_{\mathbf{w}} ||\mathbf{y} - X\mathbf{w}||_2^2 + \lambda ||\mathbf{w}||_2^2 \tag{3.7}$$

where $\lambda > 0$ is a regularization parameter that controls the trade-off between the square loss and the regularization term $||\mathbf{w}||_2^2$. Ridge regression is strictly convex and has a closed-form solution

$$\hat{\mathbf{w}} = \left(X^T X + \lambda I\right)^{-1} X^T \mathbf{y} \tag{3.8}$$

where I denotes the identity matrix. $X^T X + \lambda I$ is always full rank, because of the additional term λI. Therefore, (3.8) is always unique even when $m < n$.

Ridge regression uses an ℓ_2-norm regularizer, which does not give sparse solutions. The optimal $\hat{\mathbf{w}}$ is typically nonzero for all entries. For feature selection, one may wish to use Lasso, which makes predictions and also returns sparse estimations of weights.

3.2.3 Lasso

There are several reasons to consider Lasso instead of least squares, ridge regression, or other ℓ_q-norm regularizations. First, a least squares solution usually has high bias and low variance. The out-of-sample risk, or test error, can be high, particularly under high-dimensional settings. Second, the solution consists only of a few nonzero entries and thus selects a subset of features. This improves the interpretability of the model for some real-world problems since we need prediction accuracy along with a subset of features that exhibit the strongest prediction effects. Another advantage comes naturally from a small number of features, where computational cost is generally lower. Finally, in ℓ_q-norm regularized problems, when $q < 1$, the problem is not convex and optimization is very challenging even for modest-sized problems. However, when $q > 1$, the problem is convex but the solution is not sparse. An example of the former is the best subset selection [79, 80], and ridge regression is an example of the latter.

3.2.3.1 Definition

Lasso [81] aims to solve the optimization problem

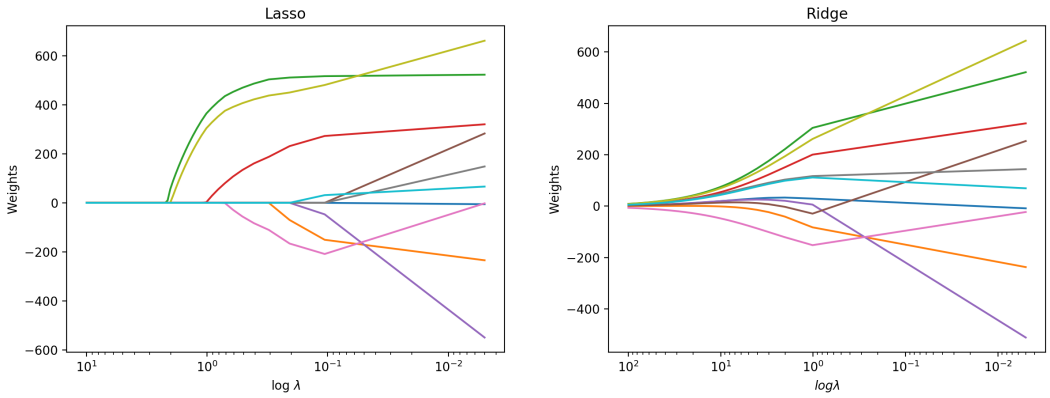
$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}||\mathbf{y} - X\mathbf{w}||_2^2 \\ \text{subject to} \quad & ||\mathbf{w}||_1 \leq t \end{aligned} \tag{3.9}$$

The Lagrangian form of (3.9) is given by

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{y} - X\mathbf{w}||_2^2 + \lambda||\mathbf{w}||_1 \tag{3.10}$$

for some $\lambda \geq 0$. An additional $\frac{1}{2}$ is added for computational convenience. There is a one-to-one correspondence between t in (3.9) and λ in (3.10).

Let us apply ridge regression and Lasso to a regression problem to predict the disease progression of diabetes one year after baseline [82]. Features include age, sex, body mass index, average blood pressure, and six blood serum measurements. Figure 3.1 contrasts the regularization paths using Lasso and ridge regression.



(a) **(b)**

Figure 3.1: Regularization paths for Lasso and ridge regression on the diabetes dataset. ℓ_1-norm in Lasso gives sparse solutions, where $\mathbf{w}$ tends to be sparse when λ is large. ℓ_2-norm in ridge regression ensures all entries in $\mathbf{w}$ are nonzero even when λ is large.

Figure 3.2 compares Lasso and ridge regression in a two-dimensional feature space. The optimal solutions of both methods correspond to the first point when the elliptical contours reach the constraint areas. We see that the sharp corners

and straight edges make it more likely to include zeros on both axes in Lasso. As dimensionality increases, the diamond constraint region becomes a rhomboid with many flat edges and corners for sparse solutions.

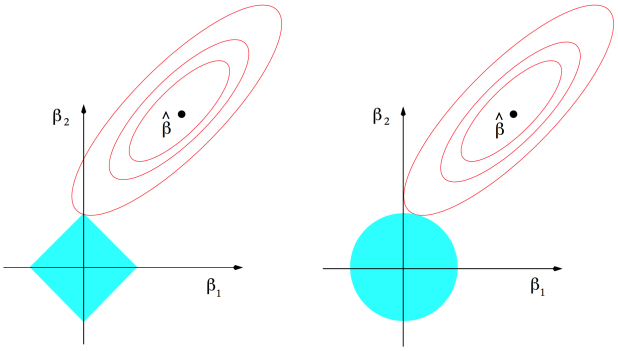


Figure 3.2: A comparison between ℓ_1 (left) and ℓ_2 (right) regularization constraints in Lasso and ridge regression [83]. β_1, β_2: weights.

3.2.3.2 Solving Lasso

Taking the derivative of (3.10) and setting the derivative to 0, we reach the Karush-Kuhn-Tucker (KKT) condition

$$X^T X\mathbf{w} - X^T\mathbf{y} + \lambda\mathbf{s} = 0 \tag{3.11}$$

where each $\mathbf{s}_j \in \mathbf{s}$ is the subgradient of $|\mathbf{w}_j|$ such that

$$\mathbf{s}_j = \begin{cases} \text{sign}(\mathbf{w}_j) & \mathbf{w}_j \neq 0 \\ [-1, +1] & \mathbf{w}_j = 0 \end{cases}$$

Therefore, there is no closed-form solution of Lasso by solving (3.11).

Many techniques from convex optimization theory have been developed to solve Lasso. These include coordinate descent [84–87], subgradient descent, least-angle regression (LARS) [82], and proximal gradient methods [88, 89]. The choice of optimization method depends on specific data and problem settings. See [90] for a concise review of the algorithms used to solve the Lasso problem. In this chapter, we focus on coordinate descent, as it converges fast (faster than LARS if combined

with active set [87]), and it can be extended to many general settings.

Coordinate descent [91, 92] is an optimization algorithm that minimizes along coordinates to look for the global minimum of an optimization problem. During each iteration, coordinate descent minimizes over one coordinate (or a block of coordinates) with all other coordinates (or blocks of coordinates) fixed. [92, 93] proved that for the following type of problem

$$\min_{\mathbf{w}} g(\mathbf{w}) + \sum_j h_j(\mathbf{w}_j) \tag{3.12}$$

coordinate descent is guaranteed to converge if

1. $g(\mathbf{w})$ is convex and differentiable;

2. each $h_j(\mathbf{w}_j)$ is convex;

3. $\mathbf{w}_j$ can be either a scalar or a vector, where a vector corresponds to grouped variables. When $\mathbf{w}_j$ is a vector, each group cannot overlap with other groups.

The structure of problem (3.12) and the mild conditions make coordinate descent particularly suitable for Lasso-like regularization problems and other problems such as ℓ_1-regularized logistic regression [94–96] and SVM [97, 98]. In fact, many generalizations of Lasso can be solved using coordinate descent except for fused Lasso [99]. The idea of applying coordinate descent on Lasso was introduced as the shooting algorithm in [84], but the technique was not fully appreciated, and only 10 years later did it start to attract attention for solving Lasso with high efficiency [87].

Recall the subgradient of Lasso in (3.11), if we take the derivative w.r.t. a nonzero coordinate $\mathbf{w}_j$ and set the derivative to 0, we have

$$X_j^T X_j \mathbf{w}_j + X_j^T (X_{-j} \mathbf{w}_{-j} - \mathbf{y}) + \lambda \mathbf{s}_j = 0 \tag{3.13}$$

where $-j$ denotes the set of indices $\{1, \cdots, j-1, j+1, \cdots, n\}$. The coordinate-wise

update is given by

$$\mathbf{w}_j = \mathcal{S}\left(\frac{X_j^T\left(\mathbf{y} - X_{-j}\mathbf{w}_{-j}\right)}{X_j^T X_j}, \frac{\lambda}{X_j^T X_j}\right) \tag{3.14}$$

where $\mathcal{S}(z, \lambda)$ is the soft-thresholding such that

$$\mathcal{S}(z, \lambda) = \begin{cases} z - \lambda, & z > 0, \lambda < |z| \\ z + \lambda, & z < 0, \lambda < |z| \\ 0, & \lambda \geq |z| \end{cases} \tag{3.15}$$

Coordinate descent works by iteratively solving $\mathbf{w}_j$ for all coordinates $j = 1, 2, \cdots, n, 1, 2, \cdots, n, \cdots$ until convergence. In Lasso, X is often assumed to be standardized where each column is centered with unit variance so that $\frac{1}{m}\sum_i X_{ij} = 0, \frac{1}{m}\sum_i X_{ij}^2 = 1$ for $j = 1, \cdots, n$. Therefore, (3.14) can be simplified as

$$\mathbf{w}_j = \mathcal{S}\left(X_j^T\left(\mathbf{y} - X_{-j}\mathbf{w}_{-j}\right), \lambda\right) \tag{3.16}$$

$X_j^T\left(\mathbf{y} - X_{-j}\mathbf{w}_{-j}\right)$ is the least square estimation when fitting the model to X_j. In the rest of this thesis, we assume that X is standardized by default. Several strategies [87] can be exploited to improve the efficiency of solving Lasso with coordinate descent, including:

1. Active set: after a cycle through n features, the algorithm runs only with the active set (features with nonzero weights) until convergence. Another cycle or KKT condition check is required to ensure the active set has not changed. This is very useful particularly in high-dimensional settings.

2. Warm start: Lasso can be solved by fitting the model with a sequence of parameters from λ_{max} to λ_{min}. $\lambda_{max} = \max_j |X_j^T\mathbf{y}|$ is the smallest possible λ when all entries of $\mathbf{w}$ receive zero. Since solutions from one λ to the next do not change much, the convergence rate is much faster for the entire sequence of λ.

3.2.4 Support vector machine and linear programming boosting

3.2.4.1 Support vector machine

For binary classification problems, support vector machine (SVM) [100, 101] is commonly used. SVM looks for the maximum-margin solution by solving the following problem.

$$\begin{aligned} \max_{\xi,\mathbf{w}} \quad & M \\ \text{subject to} \quad & \mathbf{y}_i f(\mathbf{x}_{(i)}, \mathbf{w}) \geq M(1-\xi_i) \\ & ||\mathbf{w}||_2 = 1 \\ & \xi_i \geq 0, \sum_i \xi_i \leq C \\ & i = 1, \cdots, m \end{aligned} \tag{3.17}$$

where $f(\mathbf{x}_{(i)}, \mathbf{w}) = \mathbf{x}_{(i)}^T \mathbf{w}$. Problem (3.17) is equivalent to

$$\min_{\mathbf{w}} \sum_i (1 - \mathbf{y}_i f(\mathbf{x}_{(i)}, \mathbf{w}))_+ + \lambda ||\mathbf{w}||_2^2 \tag{3.18}$$

For feature selection, ℓ_1-regularized SVM [102, 103] is often used.

$$\min_{\mathbf{w}} \sum_i (1 - \mathbf{y}_i f(\mathbf{x}_{(i)}, \mathbf{w}))_+ + \lambda ||\mathbf{w}||_1 \tag{3.19}$$

which indeed solves the similar problem as the linear programming boosting algorithm (LPBoost) [104], but uses different optimization techniques from LPBoost. In this thesis, we exploit both LPBoost and ℓ_1-regularized SVM, depending on the specific types of data and problems.

3.2.4.2 Linear programming boosting

LPBoost solves the same problem as ℓ_1-regularized SVM in Section 3.2.4.1,

$$\min_{\mathbf{w}} \sum_i (1 - \mathbf{y}_i f(\mathbf{x}_{(i)}, \mathbf{w}))_+ + \lambda ||\mathbf{w}||_1 \tag{3.20}$$

but uses different optimization algorithms. It makes use of the column generation technique, where the feature matrix does not have to be explicit. We will further

discuss column generation and show how it can be used to solve LPBoost later in this section.

We denote by F an $m \times n$ matrix of the outputs on the training data given by a set of functions $\mathcal{F}$. The primal LP formulation is

$$\begin{aligned} \max_{\xi,\rho} \quad & \rho - D \sum_{i=1}^{m} \xi_i \\ \text{subject to} \quad & \mathbf{y}_i F_i \mathbf{w} + \xi_i \geq \rho \\ & \xi_i \geq 0 \\ & \sum_{j=1}^{n} \mathbf{w}_j = 1, \mathbf{w}_j \geq 0 \\ & i = 1, \ldots, m, j = 1, \ldots, n \end{aligned} \tag{3.21}$$

By introducing Lagrange multipliers, we arrive at the dual form of the LP formulation.

$$\begin{aligned} \min_{\alpha,\beta} \quad & \beta \\ \text{subject to} \quad & \sum_{i=1}^{m} \alpha_i \mathbf{y}_i F_{ij} \leq \beta \\ & \sum_{i=1}^{m} \alpha_i = 1, 0 \leq \alpha_i \leq D \\ & i = 1, \ldots, m, j = 1, \ldots, n \end{aligned} \tag{3.22}$$

Column generation, which is an efficient algorithm for solving linear programs, can be applied to solve LPBoost. It iteratively 1) solves the original problem with a subset of features (the master problem) and 2) looks for a new feature column with a negative reduced cost and adds this column to the master problem (the subproblem). The process is repeated until no new feature columns can be added to the master problem. For each LPBoost iteration, the algorithm optimizes over a subset of features (the master problem), which starts from one weak learner and is dynamically expanded with a base learner function (the subproblem). The base learner f is given by

$$f = \arg\max_{f_j \in \mathcal{F}} \sum_{i=1}^{m} \boldsymbol{\alpha}_i \mathbf{y}_i f_j(\mathbf{x}_{(i)}) \tag{3.23}$$

$\boldsymbol{\alpha}$ is estimated by solving the last iteration of the master problem (3.22). f becomes a new column to be added to the subset of weak learners for the next iteration of the master problem. We assume that the learning functions $\mathcal{F}$ select the best weak learners from hypothesis space F, and F is closed under complementation with the base learner in (3.23).

Therefore, LPBoost can be solved iteratively with the following approach.

1. Master problem: solve for $\boldsymbol{\alpha}, \beta$ from problem (3.22) with current F;

2. Subproblem: look for a new feature column f_j using (3.23) with current $\boldsymbol{\alpha}$, add f_j to F.

LPBoost algorithm is given in Algorithm 3.1. In Chapters 4 and 5, we present efficient dynamic programming algorithms on LPBoost for subsequence detection.

Algorithm 3.1 LPBoost($F, \mathbf{y}, \nu$)

Input: hypothesis space F, labels $\mathbf{y}$, regularization parameter ν
Initialize
$j \leftarrow 0$
$D \leftarrow \frac{1}{m\nu}$
$\beta \leftarrow 0$
$\boldsymbol{\alpha} \leftarrow (\frac{1}{m}, \ldots, \frac{1}{m})$
while $\max_{f_j \in \mathcal{F}} \sum_{i=1}^{m} \boldsymbol{\alpha}_i \mathbf{y}_i f_j(\mathbf{x}_{(i)}) \geq \beta$ **do**
  $j \leftarrow j + 1$
  Find weak learner f_j with (3.23)
  $F_{ij} \leftarrow f_j(\mathbf{x}_{(i)})$
  Solve LP problems with existing feature space

$$(\beta, \boldsymbol{\alpha}) \leftarrow \begin{array}{ll} \text{argmin} & \beta \\ \text{subject to} & \sum_{i=1}^{m} \boldsymbol{\alpha}_i \mathbf{y}_i F_{ij} \leq \beta \\ & \sum_{i=1}^{m} \boldsymbol{\alpha}_i = 1, 0 \leq \boldsymbol{\alpha}_i \leq D \\ & i = 1, \ldots, m, j = 1, \ldots, n \end{array}$$

end while
$\mathbf{w} \leftarrow$ Lagrange multipliers from last LP solution
Return w

3.3 Generalizations of Lasso

In this section, we introduce several generalizations of Lasso to select sparse features from various problems.

3.3.1 Elastic net

Lasso performs rather erratically when features are correlated. It selects one or a few features from each group of highly correlated features [105]. Let us consider a simple but extreme example where the input data contain two identical features: X_i and X_j with identical weights $\mathbf{w}_i^* = \mathbf{w}_j^*$. Under ℓ_1-norm regularization, the corresponding weights are undefined with infinite solutions, as long as $|\mathbf{w}_1| + |\mathbf{w}_2|$ gives the smallest penalty, while under ℓ_2 regularization, the optimal solution is recovered when $\mathbf{w}_i = \mathbf{w}_j$.

Elastic net [105] compromises between ℓ_1 and ℓ_2 regularizations. It minimizes the following problem

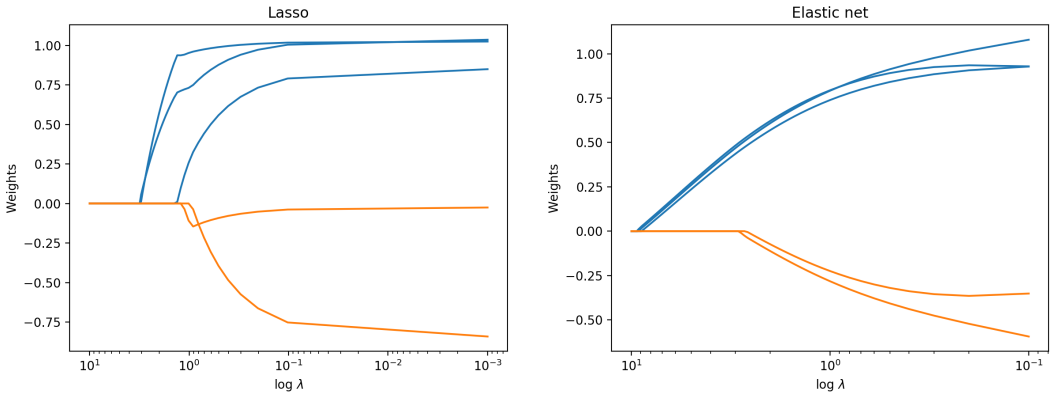
$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{y} - X\mathbf{w}||_2^2 + \lambda_1||\mathbf{w}||_1 + \frac{\lambda_2}{2}||\mathbf{w}||_2^2 \tag{3.24}$$

which is equivalent to

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{y} - X\mathbf{w}||_2^2 + \lambda\left(\alpha||\mathbf{w}||_1 + \frac{1}{2}(1-\alpha)||\mathbf{w}||_2^2\right) \tag{3.25}$$

When $\alpha = 1$, elastic net is equivalent to solving a Lasso problem and when $\alpha = 0$, elastic net solves ridge regression. Unlike Lasso, elastic net has a "grouping effect": it tends to select groups of correlated features altogether [105]. Figure 3.3 shows an example which applies Lasso and elastic net on a small dataset with only 5 features. The dataset of 50 samples is generated as follows:

$$\begin{aligned} &z_1, z_2 \sim \mathcal{N}(0,1), \epsilon_j \sim \mathcal{N}(0,1), j = 0, 1, \cdots, 5 \\ &y = 3z_1 - z_2 + 0.5\epsilon_0 \\ &X_j = z_1 + 0.2\epsilon_j, j = 1, 2, 3 \\ &X_j = z_2 + 0.2\epsilon_j, j = 4, 5 \end{aligned} \tag{3.26}$$



(a) **(b)**

Figure 3.3: Regularization paths for Lasso and elastic net ($\alpha = 0.4$). Blue: $\mathbf{w}_1$, $\mathbf{w}_2$, and $\mathbf{w}_3$; orange: $\mathbf{w}_4$ and $\mathbf{w}_5$

Therefore, X_4 and X_5 are highly correlated and the other three features also share high correlations. We can see from Figure 3.3 that elastic net selects those two groups of correlated features at the same time, with similar estimated weights.

The ℓ_2-norm regularizer in elastic net not only provides reasonable feature selection from correlated feature groups, but it also guarantees strict convexity (when $\lambda_2 > 0$), which gives unique solutions. We shall further discuss model selection consistency of elastic net in Section 3.4.

Elastic net also exploits fast implementation of coordinate descent to solve the optimization problem [85, 106]. The coordinate-wise update of $\mathbf{w}_j$ is given by

$$\mathbf{w}_j = \frac{\mathcal{S}\left(X_j^T\left(\mathbf{y} - X_{-j}\mathbf{w}_{-j}\right), \lambda\alpha\right)}{1 + \lambda(1-\alpha)} \tag{3.27}$$

The solution is equivalent to (3.16) when the ℓ_1-ratio $\alpha = 1$, and can be used to solve ridge regression when $\alpha = 0$.

One obvious disadvantage of elastic net is the extra computational cost from the tuning parameter α. In practice, this parameter sometimes is predefined on subjective grounds, but for more accurate estimation, both α and λ need to be determined on the basis of grid cross-validation at a greater computational expense.

3.3.2 Group Lasso

In some problems, data comes with a natural group structure. For example, sets of dummy variables to encode categorical features can be viewed as grouped features. Group structures also exist in many biological datasets including microarray data or genes which often lie in known biological pathways. It is therefore of interest to determine which biological pathway is most relevant to the learning problem. Consequently, it is desirable to have all features within the same group receive nonzero (or zero) weights at the same time.

The group Lasso [107] is defined as

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{y} - X\mathbf{w}||_2^2 + \lambda \sum_{g \in \mathcal{G}} ||\mathbf{w}_g||_2 \tag{3.28}$$

where $\mathcal{G}$ represents the partitions of predefined grouped features, and $g \in \mathcal{G}$ is the set of indices of a feature group. Clearly, if the group size equals 1 for all groups, the group Lasso is equivalent to solving a standard Lasso optimization. Group Lasso favors larger groups since it equally penalizes all groups. In [107], the authors proposed to penalize group sizes by an additional variable $\sqrt{n_g}$, where n_g is the size of group g. The choice of this penalized term is very subjective, and for simplicity, we only introduce group Lasso without this penalty.

Similar to Lasso and elastic net, group Lasso can be solved by block coordinate descent [85, 107]. When X is orthonormal, there exists a closed-form update for group g.

$$\mathbf{w}_g = \left(||X_g^T \left(\mathbf{y} - X_{-g}\mathbf{w}_{-g}\right)||_2 - \lambda\right)_+ \frac{X_g^T \left(\mathbf{y} - X_{-g}\mathbf{w}_{-g}\right)}{||X_g^T \left(\mathbf{y} - X_{-g}\mathbf{w}_{-g}\right)||_2} \tag{3.29}$$

where $-g$ represents the complement of g in the set of features.

For non-orthonormal cases, the group Lasso can be solved using iterative methods or composite gradient methods [108, 109]. A block coordinate descent approach for non-orthonormal X is derived in [110]

3.3.3 Sparse group Lasso

For some problems, it is desirable to have some intra-group sparsity for the nonzero groups in group Lasso. For example, in gene selection studies, perhaps not all genes in the same biological pathway are active in response to the disease of interest. One of the variations of group Lasso, the sparse group Lasso [111, 112], is designed to work with such problems. The sparse group Lasso solves

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{y} - X\mathbf{w}||_2^2 + \lambda \sum_{g=1}^{\mathcal{G}} \Big(\alpha||\mathbf{w}_g||_1 + (1-\alpha)\,||\mathbf{w}_g||_2\Big) \tag{3.30}$$

The sparse group Lasso can be solved by coordinate descent algorithms [111]. Under orthonormal assumption, the coordinate-wise update takes the form

$$\mathbf{w}_g = \Big(||\mathcal{S}(X_g^T\mathbf{y}, \lambda\alpha)||_2 - \lambda(1-\alpha)\Big)_+ \frac{\mathcal{S}(X_g^T\mathbf{y}, \lambda\alpha)}{||\mathcal{S}(X_g^T\mathbf{y}, \lambda\alpha)||_2} \tag{3.31}$$

3.3.4 Exclusive group Lasso

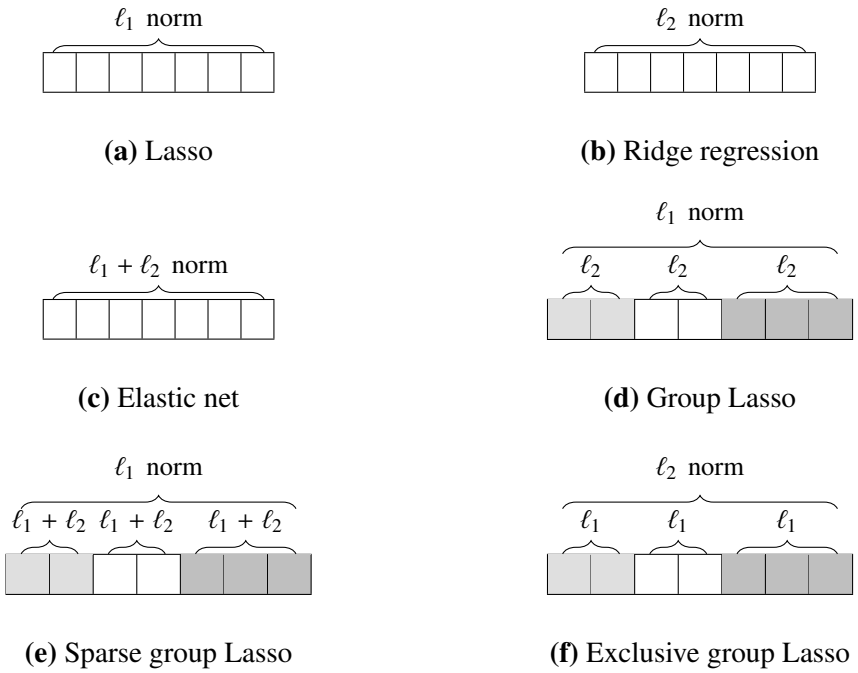
The group Lasso introduces inter-group sparsity. A similar method, the exclusive group Lasso [113], introduces intra-group sparsity by solving the following optimization problem:

$$\min_{\mathbf{w}} ||\mathbf{y} - X\mathbf{w}||_2^2 + \lambda \sum_{g\in\mathcal{G}} ||\mathbf{w}_g||_1^2 \tag{3.32}$$

Compared with group Lasso, which applies ℓ_2-norm regularization on intra-group features and ℓ_1-norm regularization across groups, the exclusive group Lasso applies ℓ_1-norm regularization at intra-group levels and ℓ_2-norm at inter-group levels. As a result, the exclusive group Lasso selects sparse features from individual groups. Figure 3.4 compares the regularization that are applied to the algorithms that have been introduced so far.

The exclusive group Lasso is originally proposed for all types of convex loss functions $f(\mathbf{w})$, and problem (3.32) can be rewritten as

$$\min_{\mathbf{w}} f(\mathbf{w}) + \lambda \sum_{g\in\mathcal{G}} ||\mathbf{w}_g||_1^2 \tag{3.33}$$



(a) Lasso **(b)** Ridge regression

(c) Elastic net **(d)** Group Lasso

(e) Sparse group Lasso **(f)** Exclusive group Lasso

Figure 3.4: A comparison of the regularization terms of Lasso and its generalizations.

The authors proposed an iterative algorithm to solve the exclusive group Lasso with an auxiliary diagonal matrix F.

$$F_{ii} = \frac{1}{|\mathbf{w}_i|} \sum_{g \in \mathcal{G}} I_g^i ||\mathbf{w}_g||_1 \tag{3.34}$$

where I is a one-hot encoding indicator matrix of group allocation. For example, $I_g^i = 1$ if the i-th feature is in group g, otherwise $I_i^g = 0$. Problem (3.33) can be rewritten as

$$\min_{\mathbf{w}} f(\mathbf{w}) + \lambda \mathbf{w}^T F \mathbf{w} \tag{3.35}$$

If we take the derivative of (3.35), we have

$$\frac{\partial f(\mathbf{w})}{\partial \mathbf{w}} + 2\lambda F \mathbf{w} = 0 \tag{3.36}$$

The exclusive group Lasso can be solved using an iterative approach as follows.

1. Update $\mathbf{w}$ using equation (3.36);
2. Update F with $\mathbf{w}$ using equation (3.34).

Exclusive group Lasso was mainly used to eliminate correlated features, removing redundant information by placing each set of correlated features into the same group. In this thesis, we show the flexibility of exclusive group Lasso for correlated feature selection. Section 3.6 discusses the applications of the exclusive group Lasso and other Lasso-type algorithms, as well as the reason why exclusive group Lasso is used to select correlated features, instead of uncorrelated features, in this thesis.

3.3.5 Relaxed Lasso

We have seen that Lasso and its generalizations select features by variable shrinkage. Features that are considered irrelevant to the learning problem are shrunk to zero, contributing nothing to the final linear model. Empirically, this procedure risks over-shrinkage, meaning that the estimated weights are far too small compared to their true values. This is unlikely to be a major problem if we use Lasso or its generalizations only as a feature selection technique, but the prediction accuracy may be rather poor if we want to use the estimations of weights. Indeed, prediction-optimal regularization parameters that minimize the prediction error do not usually lead to consistent feature selection.

One way to solve this problem is to run a two-stage procedure that performs feature selection and evaluates weights with selected features. Such two-step procedures have been adapted to various studies [83, 114, 115]. The methodology in this thesis is also split into two stages: feature selection and weight estimation with selected features. In this section, we take relaxed Lasso as an example to describe the two-stage methods.

The relaxed Lasso [116] fits an ordinary Lasso with regularization parameter λ to the full set of features in the first stage while, in the second stage, it applies Lasso with a scaled regularization parameter $\psi\lambda$ to the features selected by the first stage. Let $\hat{\mathbf{w}}^{\lambda}$ denote the weights estimated by Lasso under parameter λ in the first stage, and $\mathcal{M}^{\lambda} = \{1 \leq j \leq n : \hat{\mathbf{w}}_j^{\lambda} \neq 0\}$ denote the indices of the nonzero features, the

relaxed Lasso solves the following problem in its second stage.

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{y} - X(\mathbf{w} \cdot \mathbf{1}_{\mathcal{M}^\lambda})||_2^2 + \psi\lambda||\mathbf{w}||_1 \tag{3.37}$$

where $\psi \in (0, 1]$. $\mathbf{w} \cdot \mathbf{1}_{\mathcal{M}^\lambda}$ equals 0 for features that are not selected in the first stage. Therefore, only nonzero features from the first stage are used in (3.37). $\psi = 0$ corresponds to OLS. When $\psi = 1$, we see that relaxed Lasso is equivalent to Lasso. Simplified pseudocode of the relaxed Lasso is given by Algorithm 3.2.

Algorithm 3.2 Relaxed Lasso$(X, \mathbf{y}, \Lambda)$

Input: input data X, labels $\mathbf{y}$, regularization parameters $\Lambda = \{\lambda_1, \lambda_2, \cdots\}$, $\Psi = \{\psi_1, \psi_2, \cdots\}, \psi_i \in (0, 1]$
for $\lambda \in \Lambda$ **do**
  $\hat{\mathbf{w}}^\lambda \leftarrow \text{Lasso}(X, \mathbf{y}, \lambda)$
  $\mathcal{M}^\lambda \leftarrow \{1 \le j \le n : \hat{\mathbf{w}}_j^\lambda \neq 0\}$
  for $\psi \in \Psi$ **do**
    $\hat{\mathbf{w}}^{\psi\lambda} \leftarrow \text{Lasso}(X_{\mathcal{M}^\lambda}, \mathbf{y}, \psi\lambda)$
  end for
end for

3.4 Sign consistency

Under linear assumption, the quality of regression models can be evaluated in various ways. In many problems, the predictive performances is measured by the prediction loss, which is equivalent to the mean squared error.

$$\mathcal{L}(\hat{\mathbf{w}}, \mathbf{w}^*) = \frac{1}{m}||X\hat{\mathbf{w}} - X\mathbf{w}^*||_2^2 \tag{3.38}$$

In some applications such as signal processing, the estimated weight $\hat{\mathbf{w}}$ is of more interest, rather than the prediction $X\hat{\mathbf{w}}$. In such cases, we may wish to measure parameter estimation loss, which is defined by

$$\mathcal{L}(\hat{\mathbf{w}}, \mathbf{w}^*) = ||\hat{\mathbf{w}} - \mathbf{w}^*||_2^2 \tag{3.39}$$

Finally, when using Lasso and its generalizations for feature selection, signs of

$\hat{\mathbf{w}}$, or support recovery, are of primary interest.

$$\mathcal{L}(\hat{\mathbf{w}}, \mathbf{w}^*) = \begin{cases} 0, & \text{sign}(\hat{\mathbf{w}}_j) = \text{sign}(\mathbf{w}_j^*), j = 1, \cdots, n \\ 1, & \text{otherwise} \end{cases} \tag{3.40}$$

In this section, we review the support recovery of Lasso and elastic net, as we are more interested in feature selection. Other methods are either not well-defined for support recovery, or they are not the best solution for the research problem in this thesis (see Section 3.6 for a brief discussion of group Lasso and sparse group Lasso).

3.4.1 Lasso

Let us reconsider the support recovery in (3.40) and define a similar notation. As only signs of the estimated weights are of interest, the estimation $\hat{\mathbf{w}}$ is called sign consistent if

$$\begin{aligned} \text{supp}(\hat{\mathbf{w}}) &= \text{supp}(\mathbf{w}^*) \\ \text{sign}(\hat{\mathbf{w}}_S) &= \text{sign}(\mathbf{w}_S^*) \end{aligned} \tag{3.41}$$

where S is the support of $\hat{\mathbf{w}}$. Both equalities hold for Lasso with very high probability under certain conditions, which can be proved using the Primal-Dual Witness (PDW) method [117]. The KKT condition of Lasso is given by

$$X^T(\mathbf{y} - X\hat{\mathbf{w}}) = \lambda \mathbf{s} \tag{3.42}$$

where $\mathbf{s}$ denotes the subgradient of the ℓ_1-norm term such that

$$\mathbf{s}_j = \begin{cases} \text{sign}(\mathbf{w}_j) & \mathbf{w}_j \neq 0 \\ [-1, +1] & \mathbf{w}_j = 0 \end{cases} \tag{3.43}$$

The KKT condition indicates that the optimal subgradient is unique even though $\hat{\mathbf{w}}$ is not unique, and any two Lasso solutions must have the same signs on overlapped active features (features in support S).

Rewriting the KKT condition by separating the relevant and irrelevant features

and weights, we have

$$\begin{aligned} X_S^T(\mathbf{y} - X_S\mathbf{w}_S^*) &= \lambda \mathbf{s}_S \\ X_{-S}^T(\mathbf{y} - X_S\mathbf{w}_S^*) &= \lambda \mathbf{s}_{-S} \end{aligned} \tag{3.44}$$

where $-S$ denotes the complement of S in the set of features. The PDW method is described as follows.

1. Solve for optimal weights $\mathbf{w}_S^*$ with X_S, and set other weights $\mathbf{w}_{-S}^* = 0$;
2. Solve for the subgradient $\mathbf{s}_{-S}$;
3. Check if 1) $||\mathbf{s}_{-S}||_\infty < 1$ and 2) $\text{sign}(\hat{\mathbf{w}}_S) = \text{sign}(\mathbf{w}_S^*)$ hold. We believe that we have reached an sign consistent solution to Lasso if both conditions are satisfied.

Substituting $\mathbf{y}$ with $\mathbf{y} = X\mathbf{w} + \epsilon$ and rewriting the conditions in Step 3, we come to the conditions for Lasso to discover the true features. These conditions are

1. Column normalization

$$\frac{1}{m}||X_i^T X_i||_2^2 \leq 1, \quad i = 1, \cdots, n \tag{3.45}$$

2. Irrepresentable condition or mutual incoherence [118]

$$|X_{-S}^T X_S (X_S^T X_S)^{-1} \text{sign}(\mathbf{w}_S)| \leq 1 - \gamma, \quad \gamma > 0 \tag{3.46}$$

When $\text{sign}(\mathbf{w}_S)$ is unknown, (3.46) needs to hold for all possible $\text{sign}(\mathbf{w}_S)$.

$$|(X_S^T X_S)^{-1} X_S^T X_{-S}| \leq 1 - \gamma \tag{3.47}$$

3. Minimum eigenvalue [117]

$$\Lambda_{\min}(\frac{1}{m} X_s^T X_S) \geq C \tag{3.48}$$

for some $C > 0$, where $\Lambda_{\min}(.)$ is the minimum eigenvalue.

4. Minimum signal [74, 119]

$$\min(\mathbf{w}^*) \geq \lambda||(X_S^T X_S)^{-1}||_\infty + \frac{4\gamma\lambda}{\sqrt{C}} \tag{3.49}$$

for $\lambda \geq (2\sigma/\gamma)\sqrt{2m\log(n)}$.

The irrepresentable condition or mutual incoherence assumes that the relevant features X_S and the irrelevant features X_{-S} are not correlated, while the minimum eigenvalue condition requires that the relevant features cannot be overly correlated with each other. Besides, the minimum signal condition ensures that the weights are large enough to be detected. If any of those conditions fail, then the probability that Lasso discovers true features is below $\frac{1}{2}$ [120].

3.4.2 Elastic net

Like the irrepresentable condition, elastic irrepresentable condition, is derived for elastic net [121]. The condition is given by

$$||X_{-S}^T X_S \left(X_S^T X_S + \frac{\lambda_2}{2m} I\right)^{-1} \left(\text{sign}(\mathbf{w}_S^*) + \frac{\lambda_2}{\lambda_1}\mathbf{w}_S^*\right)||_\infty \leq 1 - \gamma \tag{3.50}$$

Therefore, elastic net enjoys a more relaxed condition than Lasso to achieve sign consistency, because of the term $\frac{\lambda_2}{m} I$ inside the inverse. This may explain the difference in performance on correlated features using Lasso and elastic net. In Chapter 6, we examine the experimental conditions for the exclusive group Lasso to select the correct set of features, and we compare the feature selection performance with both Lasso and elastic net.

3.5 Stability selection

So far, we have reviewed various Lasso-type algorithms with an emphasis on feature selection. Previous studies showed that Lasso and elastic net require stringent conditions to select correct features, while in practice, such conditions are easily to violate. Stability selection [122] uses a combination of feature selection algorithms and subsampling to achieve consistency in feature selection even when the

necessary conditions for consistency of Lasso are violated.

Stability path is similar to the regularization path shown in Section 3.2.3. It describes the probabilities for each feature to be selected over a range of regularization parameters with random subsamples from the dataset. We denote $\hat{S}^{\lambda}(D)$ as the set of selected variables from a set of samples D for a given parameter λ. The selection probability is defined as

$$\Pi_K^{\lambda} = P[K \leq \hat{S}^{\lambda}(D)] \tag{3.51}$$

where D is a random subsample of the data of size $|D| = \frac{m}{2}$, and set $K \subseteq \{1, ..., n\}$. Therefore, Π_K describes the probability of set K being selected in $\hat{S}^{\lambda}(D)$. The size is chosen to be $\frac{m}{2}$ because this resembles the bootstrap [123, 124]. The set of stable features is defined as

$$\hat{S} = \{K : \max_{\lambda} \hat{\Pi}_K^{\lambda} \geq \pi\} \tag{3.52}$$

where π is a pre-defined threshold for selection probabilities. Any variables with selection probability higher than the threshold π are considered to be consistent. It is noticeable that for each variable, only the highest selection probability from a set of regularization parameters is considered. Therefore, determining the correct regularization parameter is less demanding because it avoids the risk incurred from high-dimensional feature space. The stability selection algorithm is given by Algorithm 3.3.

Algorithm 3.3 Stability selection$(X, \mathbf{y}, \Lambda, N)$

Input: input data X, labels $\mathbf{y}$, regularization parameters $\Lambda = \{\lambda_1, \lambda_2, \cdots\}$, number of iterations N
for $\lambda \in \Lambda$ **do**
  for $i = 1 : N$ **do**
    Subsample without replacement to generate subsets $X_{I_i}, \mathbf{y}_{I_i}$, where $I_i \subseteq \{1, \cdots, m\}, |I_i| = \frac{m}{2}$
    Run selection algorithms with X_{I_i} and y_{I_i}
  end for
  $\Pi_K^{\lambda} \leftarrow \frac{1}{N} \sum_i \mathbb{I}\{K \in \hat{S}_i^{\lambda}\}$
end for
$\hat{S}^{stable} \leftarrow \{K : \max_{\lambda} \hat{\Pi}_K^{\lambda} \geq \pi\}$

Both Lasso and randomized Lasso were used as selection algorithms for stability selection [122]. The randomized Lasso solves the problem

$$\min_{\mathbf{w}} \frac{1}{2}||\mathbf{y} - X\mathbf{w}||_2^2 + \lambda \sum_j \frac{\mathbf{w}_j}{\beta_j} \tag{3.53}$$

where β_j is a variable randomly drawn from $\alpha \in (0, 1]$. When $\alpha = 1$, (3.53) is equivalent to the vanilla Lasso. Features with sufficiently large weights can be identified by the stability selection under some conditions over the minimum and maximum eigenvalues. The conditions are indeed more relaxed than the irrepresentable condition or elastic irrepresentable condition. However, constraints on eigenvalues still put restrictions on the correlation between features. As we shall see in Chapter 6, stability selection with the original Lasso still cannot distinguish relevant and irrelevant features under high-correlation settings.

Another advantage of stability selection is that it is less sensitive to the choice of regularization parameters in some examples [122]. However in practice, stability selection can be sensitive to parameters. To address both problems, in Chapter 6, we present various methods in conjunction with stability selection as well as a novel strategy for model selection.

3.6 Related work

In machine learning, feature selection is very important mainly for two main reasons. First, it increases the prediction accuracy and reduces the computational cost by eliminating irrelevant features. Second, it improves model interpretability by identifying a sparse representation of the data. Feature selection can be grouped into three categories: filter, wrapper, and embedded methods [125]. Filter methods select features regardless of the learning method; they include various metrics such as Pearson correlation, mutual information, and Fisher score. Wrapper methods evaluate useful subsets of features with learning algorithms. Common wrapper methods are forward feature selection, backward feature elimination, and recursive feature elimination. Embedded methods combine feature selection with prediction. Algorithms that have been introduced so far are generally embedded methods, as

feature selection is embedded when fitting the model. In this section, we review the broad application of Lasso and its generalizations for feature selection, with an emphasis on biology, particularly immunology.

Many biological functions and pathways are sophisticated processes with complex combinations of the participants involved. For example, the marginal effects of a single nucleotide polymorphism (SNP) (the effects of a SNP on a disease) may be different from the joint effects when it functions with other SNPs. Such marginal effects may be weaker than the joint effects for some SNPs. Besides, a SNP that is not related to a disease may be correlated to a SNP that is related to the same disease. Early work on feature selection in the context of biology often ranked the features by predetermined criteria such as correlation coefficients [126, 127] and parametric or non-parametric statistical tests [128–130]. However, this ranking approach often made use of univariate statistical tests and overlooked the mutual information from the combination of features.

Lasso [81] naturally selects a subset of features and predicts with the selected features efficiently. Therefore, Lasso and ℓ_1-regularized methods have been extensively applied for feature selection in various biological problems, such as gene selection [131–134] and cell-type-specific markers for sorting immune cells [135]. ℓ_1-regularized methods have also been used in sparse feature selection in studies on T cells and TCRs [136–138]. However, Lasso often performs erratically when features are correlated, which is a general problem in biological research as features represent the biochemical process where the components act together in groups. Many studies that exploited Lasso for feature selection did not consider feature correlation in biology, including many gene selection studies, where genes are known to share a high level of correlation [131–135].

Group Lasso [107] enforces sparsity on groups of correlated features and is widely used in many biological problems. Entire groups are either selected or deselected by group Lasso penalty so that both genes and their functions that are relevant to the learning problem will be selected. Since many genes can function in multiple biological pathways, it is necessary to allow overlaps in group Lasso [139, 140].

A close relative of group Lasso is sparse group Lasso [111, 112], which enforces sparsity both on features from different groups and on features in the same group. Therefore, sparse group Lasso provides more reasonable solutions since features in the same pathway do not have to be activated simultaneously. Although the original sparse group Lasso does not allow overlaps across groups, [141] presents an approach that adapts overlap group Lasso in [140] into the sparse group Lasso framework.

Both group Lasso and sparse group Lasso studies have often used predefined feature groups based on the biological functions of genes [110, 141, 142]. This approach has several disadvantages. First, biological pathways are still incomplete and are not available for all genes. Second, genes will be allocated into the same group regardless of their relationship to a biological function. For example, genes that are closely or loosely related to a function will be allocated into the same group. Third, although many genes have known biological pathways, such prior knowledge is still absent for many other biological features. This limits the application of group Lasso and sparse group Lasso to many other problems.

As regards the limitations of grouping features by their biological functions, many attempts have been made to group features without prior knowledge of their biological functions. For example, [143] adapted the methods in [144] and constructed networks of genes based on their correlation. However, sets of biological features often work in different joint pathways and statistical groups do not usually match functional groups [145, 146]. Another possibility is to group features by clustering algorithms such as k-means [147] and hierarchical clustering [148]. A disadvantage of this approach arises from the uncertainty of the initial states and the results of the clustering algorithms [149]. In addition, clustering algorithms cannot allocate genes to multiple groups and some algorithms, such as hierarchical clustering, are often computationally expensive.

A group-free method is elastic net, which requires more relaxed conditions than Lasso to select relevant features correctly, even when these features are correlated, as discussed in Section 3.4. Therefore, elastic net has also been applied to

various biological problems. [150, 151] applied elastic net for analysis of infections and vaccinations. It also has been used for other immunology problems such as analyzing the impact of age, sex, and CMV status on the human immune system [152]; modeling differential T cells including naive, central memory, and effector memory T cells [153]; and detecting T-cell genes associated with type I diabetes [154] or lung cancer [155]. One limitation of elastic net is the nested cross-validation over a grid of parameters, which increases the computational time. In most studies, the ratio of the ℓ_1 term, α, is to a subjective value [151, 152, 154, 155].

Similar to group Lasso, the exclusive group Lasso also handles group-based feature space. Group Lasso uses an $\ell_{2,1}$-norm regularization to enforce inter-group sparsity where entire groups are either selected or eliminated simultaneously, while exclusive group Lasso uses an $\ell_{1,2}$-norm regularization to enforce intra-group sparsity where all groups are kept with sparse feature selection within each group. The exclusive Lasso penalty was introduced in [156] to remove the positive correlation of weights among classes in multi-class classification problems. It was further improved to work with arbitrary group structures in other problems in [113], but again, this algorithm was used to select uncorrelated features. Indeed, most studies used exclusive group Lasso to select uncorrelated features [157–160]. Like group Lasso, the use of exclusive group Lasso is often restricted to applications with prior knowledge of groups. For example, in multi-class classification problems, weights of the same feature in different classes can be considered as a group with some level of intra-group sparsity [160, 161]. In biological problems such as Alzheimer's disease, features can be grouped by the modalities in multimodal diagnosis of the disease [162]. Another application in biology is personalized drug design, where groups naturally come from samples taking various drug treatments [163]. This has largely limited the application of exclusive group Lasso in many studies since the prior knowledge of groups is not always available [164, 165]. A possible solution is to group features based on their correlation [113]. However, as discussed with group Lasso and in Section 2.4, the correlations between biological features may not reveal a stable group structure relevant to the learning problem. In addition,

computing the correlation under high-dimensional settings can be very expensive in both computational time and memory space. Random groups were used in [159] but the experiments were based on a single random partition of uncorrelated features, which is very likely to give unstable results on correlated features. A recent study adapted ℓ_p-norm regularization ($0 < p < 1$) in independently interpretable Lasso [166], which handles feature correlation and selects uncorrelated features, without grouping the features [164]. However, the ℓ_p-norm regularized algorithm did not perform well when data has high coherence. [162] presented an approach using weighted $\ell_{1,p}$-norm regularization ($p > 1$), but applied this approach to data with known grouping structure.

Stability selection [122] provided a flexible framework that combines subsampling with high-dimensional selection algorithms. [122] showed that stability selection in conjunction with Lasso captured relevant features correctly even when Lasso alone failed to select the correct set of features. Various selection algorithms or criteria have been used in the stability selection framework in different biological problems. A combination of ℓ_1-regularized SVM and bootstrap, which is very similar to stability selection, was proposed to identify biomarkers from gene expression data [167]. ℓ_1-regularized algorithms such as Lasso were also applied in the framework of stability selection to select features from genes and neuroimages [168–170]. Other selection criteria, such as iterative sure independence screening [171] and minimum redundancy maximum relevance [172], were also used with stability selection [173, 174].

Although stability selection with high-dimensional selection algorithms provides a flexible and powerful tool to correctly select relevant features, multiple parameters, such as the threshold for selection probabilities, are required for accurate selection results. Some studies exploiting stability selection used a fixed threshold [173] or simply selected the features that were chosen in all iterations [167] or in majority subjects [174]. Although [122, 173] argued that the threshold for choosing features does not have a high impact on the results, some studies showed different threshold values still lead to different results and applied an exhaustive

nested cross-validation procedure to tune the subsampling size, the number of iterations, the threshold of selection probabilities, as well as regularization parameters in the selection algorithm and the final classifier [169, 170]. Furthermore, [169, 170] provided a niche approach to control the false discovery rate and evaluate the robustness to variations of the input data, by randomly permuting a small number of features. In this thesis, we adapt a close idea to ease the exhaustive cross-validation procedure of stability selection, as will be discussed in Chapter 6.

Although most studies used exclusive group Lasso to select uncorrelated features, the flexibility of exclusive group Lasso penalty inspired us to select correlated features. This can be achieved by allocating relevant but correlated features into different groups to avoid competition between these features. Since the intrinsic group structure is unknown in many applications, the exclusive group Lasso can be impractical to tackle real-world problems. To our knowledge, no study has given practical instructions on how to partition feature groups for consistent feature selection in exclusive group Lasso when the underlying structure of features is unknown. In this thesis, we present a random grouping allocation strategy in conjunction with a novel approach of artificial features as well as stability selection. We explore the application of exclusive group Lasso for correlated feature selection in the context of biology, and we provide efficient algorithms to solve exclusive group Lasso and practical methods for group allocation in Chapter 6. Meanwhile, we explore and develop fast algorithms of the widely used ℓ_1-regularized methods to work with kernel features constructed from biological sequences in Chapters 4 and 5.

Chapter 4

ℓ_1-regularization in identifying subsequences

In this chapter, we propose a machine learning pipeline for feature selection from biological sequences. Following a brief review of string kernel and Fisher kernel in Section 4.1, we show how to construct features with string kernel and Fisher kernel features from sequencing data in Section 4.2. Section 4.3 presents efficient dynamic programming algorithms to solve LPBoost with features indexed by substrings with matrices and graphs. Experimental results to extract antigen-specific mouse CDR3β subsequences are shown in Section 4.4. We conclude our work of this chapter in Section 4.5.

4.1 String kernel and Fisher kernel

4.1.1 String kernel

Most machine learning algorithms are designed to work on vector inputs, in which each entry of a data sample is represented by a vector. This prevents the algorithms to work directly with sequencing data such as text documents and biological sequences. The string kernel [175–177] is a kernel function that operates on strings and measures the similarity between strings. Such kernels allow kernelized learning algorithms to work with strings of varying lengths. Therefore, it is widely used in the areas of text classification and bioinformatics in which sequences are to be processed. In this section, we focus on the p-spectrum string kernel and we introduce

how to apply string kernel features to construct features in Section 4.2.1.

Notation We denote the alphabet by Σ, with the number of symbols in Σ being $|\Sigma|$. We let Σ^k denote the set of strings that are indexed by k symbols from Σ. We let s_k denote the k-th symbol in a string s. We use the notation $s[i : j]$ to denote a *substring* of s, such that $s[i : j] = s_i s_{i+1} \cdots s_j$. u is defined as a substring of s, if $s = v_1 u v_2$ and $v_1, v_2 \in \bigcup_{i=0}^{\infty} \Sigma^i$.

The p-spectrum string kernel Perhaps the most natural way to measure the similarity between two strings is to count the number of common substrings that appear in both strings. This gives the p-spectrum kernel between strings s and t over all p-mers [176]:

$$K(s,t) = \sum_{u \in \Sigma^p} \phi_p^u(s) \phi_p^u(t) \tag{4.1}$$

where the embedding is defined as

$$\left(\phi_p^u(s)\right)_{u \in \Sigma^p} = |\{(v_1, v_2) : s = v_1 u v_2\}| \tag{4.2}$$

It is notable that in this thesis, either string kernel or Fisher kernel introduced in the next section is not directly used in the presented algorithms. Instead, we make use of the features on which these kernels are generated as shown in Section 4.2. We refer to the corresponding features as "string kernel features" and "Fisher kernel features" to distinguish from string and Fisher kernels.

4.1.2 Fisher kernel

The Fisher kernel [178] is computed from the gradient of modeling a finite set of smooth parameters Θ. Such a model consists of probability distributions on some input space X and can be written as

$$M = \mathcal{P}_\theta(x) : \theta \in \Theta$$

where $x \in X$. The likelihood of a data point x in model $m(\theta)$ given parameters θ is defined as

$$L_\theta(x) = \mathcal{P}(x|\theta)$$

We then use the gradient of the log likelihood to compute the Fisher score and the Fisher information matrix.

Definition 4.1. [Fisher score] The Fisher score of a data point x for a given set of parameters θ is defined as the gradient of the log likelihood of x w.r.t the parameters in model $m(\theta)$.

$$g(\theta, x) = \left(\frac{\partial \log L_\theta(x)}{\partial \theta_i} \right)_{i=1}^{N} \tag{4.3}$$

Definition 4.2. [Fisher information] The Fisher information of model $m(\theta)$ given parameters θ^0 is

$$I = \mathbb{E}\left[g(\theta^0, x) g(\theta^0, x)^T \right] \tag{4.4}$$

where $\mathbb{E}[.]$ is the expectation. We further derive the Fisher kernel from the Fisher information matrix.

Definition 4.3. [Fisher kernel] The Fisher kernel of model $m(\theta^0)$ given parameter set θ^0 is defined as

$$K(x, z) = g(\theta^0, x)^T I^{-1} g(\theta^0, z) \tag{4.5}$$

In practice we set I to be the identity matrix and Fisher kernel is given by

$$K(x, z) = g(\theta^0, x)^T g(\theta^0, z) \tag{4.6}$$

4.1.3 Fisher kernel on Markov strings

A text document can be viewed as being generated by a k-stage Markov process. If we denote the probability that a symbol x in the document follows a length-k string u as $\mathcal{P}_{u \to x}$, the probability of the document can be computed as

$$\prod_{i=1}^{|d|-k} \mathcal{P}_{d[i:i+k-1] \to d[i+k]} \tag{4.7}$$

The first k symbols are fixed or an additional k special symbols can be added to the beginning of the document [177]. If we parameterize $\theta = (c_{u \to x})_{u \in \Sigma^k, x \in \Sigma}$ such that

$$\mathcal{P}_{u \to x} = \frac{c_{u \to x}}{\sum_{x \in \Sigma} c_{u \to x}} \tag{4.8}$$

and we take the derivative of (4.7), the Fisher score becomes

$$\begin{aligned}
g(\theta^0, d)_{u\to x} &= \frac{\partial \log L_\theta(d)}{\partial c_{u\to x}} \\
&= \sum_{i=1}^{|d|-k} \frac{\partial \log \mathcal{P}_{d[i:i+k-1]\to d[i+k]}}{\partial c_{u\to x}} \\
&= \sum_{i=1}^{|d|-k} \frac{\partial \log \frac{c_{d[i:i+k-1]\to d[i+k]}}{\sum_{d[i+k]\in\Sigma} c_{d[i:i+k-1]\to d[i+k]}}}{\partial c_{u\to x}} \\
&= \sum_{i=1}^{|d|-k} [d[i:i+k-1]=u] \left([d[i+k]=x] \frac{1}{c_{d[i:i+k-1]\to d[i+k]}} \right. \\
&\quad \left. - \frac{1}{\sum_{x\in\Sigma} c_{d[i:i+k-1]\to x}} \right) \\
&= \sum_{i=1}^{|d|-k} [d[i:i+k-1]=u] \left([d[i+k]=x] \frac{1}{c_{d[i:i+k-1]\to d[i+k]}} - 1 \right)
\end{aligned} \tag{4.9}$$

If all $c_{u\to x} = \mathcal{P}_{u\to x}$ is set to be $|\Sigma|^{-1}$ for θ^0 and ignore any constant terms, the above equation then equals to

$$g(\theta^0, d)_{u\to x} = |\Sigma| \operatorname{tf}(ux, d) \tag{4.10}$$

where $\operatorname{tf}(ux, d)$ is the term frequency of string ux in document d. Omitting the second term in (4.10), the result becomes a p-spectrum string kernel scaled by $|\Sigma|$. Similarly, the result of (4.9) can be simplified to

$$g(\theta^0, d)_{u\to x} = \frac{\operatorname{tf}(ux, d)}{\mathcal{P}_{u\to x}} \tag{4.11}$$

Using the identify matrix as Fisher information, the kernel can be written as

$$K(d, d') = U_d^T U_{d'} \tag{4.12}$$

where U_d is a vector with its components being $g(\theta^0, d)_{u\to x}$ for all $u \in \Sigma^k$. In this way, the string kernel can be viewed as a Fisher kernel derived from a Markov model of the documents with uniform transition probabilities [179].

4.1.4 Related work

Both string and Fisher kernel have been widely used in biological studies. The string kernel is a kernel function that works on sequencing data by measuring the similarity or dissimilarity between two sequences or documents. It allows kernel-based algorithms such as SVM and kernel PCA to directly work with strings. The string kernel was applied to measuring document similarity for text classification problems [175]. The spectrum string kernel was introduced in [176] in conjunction with SVM for protein classification and homology detection. SVM with mismatch string kernel, which measures the dissimilarity between biological sequences, was shown to perform as well as the Fisher kernel in protein homology detection [180]. String kernels were modified to look for hidden global alignment properties from DNA or RNA sequences [181]. Domain adaption at transfer string kernel was developed to predict DNA-protein binding sites [182]. Links between the string kernel and recurrent neural network (RNN) have been studied [183] and string-kernel-derived RNN was used to model gaps in biological sequences for protein classification. Deep graph kernels were also applied to string kernels to generate deep string kernels [184]. A very brief review of string kernels was given in [185]. Various string kernels with fast computational algorithms are amply discussed in [177]. See [186] for an extensive review of early work using kernel methods in computational biology. Fisher kernel is derived from generative probabilistic models such as hidden Markov and links generative models to discriminative models such as SVM. Early applications include protein homology detection [187, 188]. Fisher kernel in conjunction with SVM was used for classification with biological sequences [189]. It was widely applied to image classification to construct Fisher vectors [190] to generate more compact representation of images than bag-of-visual words. Therefore, Fisher kernel was used for diagnosis and analysis from medical images [191–193]. Indeed, Fisher kernel is a special case of marginalised kernels, which are particularly useful for biological sequences [194].

4.2 Constructing features from sequencing data

4.2.1 String and Fisher kernel features

Either string kernel or Fisher kernel has been used to construct features from biological sequences. Instead of contiguous p-mers that occur in a sequencing document, we consider longer, non-contiguous subsequences of fixed length l. In this thesis, we specifically refer to *subsequence* as non-contiguous arrays of characters, in order to distinguish with contiguous substrings. Given a subsequence $u \in \Sigma^l$, the corresponding feature of a document d is defined as

$$\phi^u_{(p,l)}(d) = \sum_{u' \in I_u} \phi^{u'}_p(d) \tag{4.13}$$

where $\phi^{u'}_p(d)$ is a p-spectrum string kernel or Fisher kernel feature of a p-mer u' in document d, u' being a substring of u such that

$$I_u = \{u' \in \Sigma^p \mid \exists v_1, v_2, u = v_1 u' v_2\} \tag{4.14}$$

The full feature space of document d is defined over all possible subsequences of length l.

$$\phi^u_{(p,l)}(d), \forall u \in \Sigma^l \tag{4.15}$$

We define the p-mer features $\phi^{u'}_p(d)$ as *base features* to distinguish from $\phi^u_{(p,l)}(d)$, the features that correspond to subsequences of length l.

Consider a document containing only one DNA sequence

GACACAGACAC

Note that this toy example shows a simplified case, while in our experiments, a sequencing document contains multiple biological sequences. If we set $p = 3$, we see that there exist five triplets in this sequence: GAC, ACA, CAC, CAG, AGA. To compute the Fisher kernel features, we need the transition probabilities which are computed by the probabilities that a duplet transfers to a particular singlet or base (A, T, C, or G in the context of DNA) among all other nitrogenous bases. The transition probabilities for the triplets that appear in the sample sequences computed

by (4.8) are shown in Table 4.1. The string and Fisher kernel features are given in Table 4.2.

Table 4.1: Transition probabilities ($p = 3$)

GAC	ACA	CAC	CAG	AGA
1	1	0.66	0.33	1

Table 4.2: String and Fisher kernel base features of DNA sequence GACACAGACAC ($p = 3$).

ϕ	GAC	ACA	CAC	CAG	AGA
String features	2	3	2	1	1
Fisher features	2	3	3	3	1

Table 4.3 shows the string and Fisher kernel features for four sample quintuplets ($p = 3, l = 5$). GACAC and ACACA contain triplets that occur in the sequence and the corresponding features are computed by the sum of the base features; ACAGT contains two triplets in the sequence, with corresponding features computed by $f(ACA) + f(CAG)$; ACGTA contains no triplets from the sequences and thus receives 0 for both string and Fisher features. We can compute the features of all possible $4^5 = 1024$ quintuplets in the same way.

Table 4.3: Example string and Fisher kernel features of DNA sequence GACACAGACAC ($p = 3, l = 5$).

ϕ	GACAC	ACACA	ACAGT	ACGTA
String features	7	8	4	0
Fisher features	8	9	6	0

4.2.2 Weighting features

The Fisher features are equivalent to the sum of term frequencies weighted by the reciprocal of the transition probabilities. Practically, some transitions are very frequent whilst some transitions occur rarely. Rare transitions receive very high weightings which result in large features. An IDF-type logarithm weighting scheme

was used to control this effect in the semantic bag-of-words kernel [177, 179].

$$\log \frac{D}{\mathrm{df}(s)} \tag{4.16}$$

where D and $\mathrm{df}(s)$ represent the number of documents and the number of documents containing term s, respectively.

In order to obtain a similar effect to the IDF-type weighting in Fisher kernel on Markov strings, we can reparameterize the transition probability to make use of the the negative logarithm to control the "enlarged" effect caused by very small transition probabilities from the rare transitions. The transition probability can be reparameterized as

$$\mathcal{P}_{u\to x} = \exp(-\exp(-t_{u\to x})) \tag{4.17}$$

where $t_{u\to x}$ is a new auxiliary parameter so that the derivative of the log probability is still a logarithm to simulate the IDF-type weighting. The derivative of the log probability is given by

$$\frac{\partial \log \mathcal{P}_{u\to x}}{\partial t_{u\to x}} = \exp(-t_{u\to x}) = -\log \mathcal{P}_{u\to x} \tag{4.18}$$

In this way, $-\log \mathcal{P}_{u\to x}$, instead of $\frac{1}{\mathcal{P}_{u\to x}}$, is used to weight the term frequencies in Fisher kernel features.

We applied the same logarithm weighting in our experiments. The feature construction step results in a $|\Sigma|^l$-dimensional feature space, which grows exponentially at the rate of $|\Sigma|$ as we increase l, the length of the target subsequence. Obviously, this causes problems in both storage space and computational time. The dimensionality further increases if we relax the constraint of length to explore varying-length substrings. For example, including all subsequences from length l_1 to l_l results in a feature space of dimension $|\Sigma|^{l_1} + \cdots + |\Sigma|^{l_l}$. In this chapter, we only focus on fixed-length subsequences where l is predefined, and we will discuss how to efficiently cope with substrings of varying lengths in Chapter 5.

As discussed in Chapter 3, ℓ_1-regularized methods provide sparse solutions for both feature selection. In this chapter, we apply LPBoost, which uses ℓ_1 regular-

ization to select subsequences. In addition, the column generation technique used in LPBoost offers further computational convenience in solving the optimization problem using string and Fisher kernel features.

4.3 Solving LPBoost

4.3.1 Matrix-represented data

Recall the dual form of the LPBoost in Section 3.2.4.2

$$\begin{aligned} \min_{\alpha,\beta} \quad & \beta \\ \text{subject to} \quad & \sum_{i=1}^{m} \alpha_i \mathbf{y}_i F_{ij} \leq \beta \\ & \sum_{i=1}^{m} \alpha_i = 1, 0 \leq \alpha_i \leq D \\ & i = 1, \ldots, m, j = 1, \ldots, n \end{aligned} \tag{4.19}$$

In each iteration, the algorithm introduces a new column $\phi^u_{(p,l)}(d_i)$ of subsequence u to F which maximizes

$$\sum_{i=1}^{m} \hat{\alpha}_i \mathbf{y}_i \phi^u_{(p,l)}(d_i) \tag{4.20}$$

where d_i denotes the i-th sequencing document.

We notice that the string kernel or the Fisher kernel features are indeed the sum of the base features of p-mer substrings as in equation (4.13), we can therefore only save the base features and look for length-l subsequences which sum up the corresponding p-mer base features and maximize (4.20). The p-mer substring is then added to the feature columns for the next iteration of LPBoost optimization. In this way, we effectively reduce the dimensionality of the feature space from $|\Sigma|^l$ to $|\Sigma|^p$. The recursive process to look for the optimal subsequence of current iteration

is given by (4.21).

$$\begin{aligned}\max_u \sum_i \boldsymbol{\alpha}_i \mathbf{y}_i \phi^u_{(p,l)}(d_i) &= \max \sum_i \sum_{u':u=v'_1u'v'_2} \boldsymbol{\alpha}_i \mathbf{y}_i \phi^{u'}_p(d_i) \\ &= \max \sum_i \boldsymbol{\alpha}_i \mathbf{y}_i \left(\sum_{u':u[1:l-1]=v'_1u'v'_2} \phi^{u'}_p(d_i) + \phi^{u[l-p+1:l]}_p(d_i) \right) \\ &= \max \left(\sum_i \boldsymbol{\alpha}_i \mathbf{y}_i \phi^{u[1:l-1]}_{(p,l-1)}(d_i) + \sum_i \boldsymbol{\alpha}_i \mathbf{y}_i \phi^{u[l-p+1:l]}_p(d_i) \right)\end{aligned} \tag{4.21}$$

It is important to locate the substring that maximizes (4.20) during each iteration and this needs to be done efficiently. We propose a fast dynamic programming approach to compute $\phi^u_{(p,l)}$ in terms of $\phi^{u[1:l-1]}_{(p,l-1)}$ and $\phi^{u[l-p+1:l]}_p$. If we start with $v = u[l-p+1:l]$, we can compute $\phi^u_{(p,l)}$ via DP which iteratively extends the length by looking for the optimal symbol before v. At each iteration, we create a matrix DP to store $\max \sum \boldsymbol{\alpha}_i \mathbf{y}_i \phi^v_{(p,l)}(d_i)$, where DP is a $(p-1)$-dimensional array such that each dimension or sheet is indexed by the symbols in the alphabet Σ. For example, when $p = 3$, DP for DNA sequences is a 4×4 matrix over 4 nitrogenous bases of DNA, and DP for amino acid sequences is a 20×20 matrix over 20 amino acids (see Appendix B), respectively. When $p = 4$, DP becomes $4 \times 4 \times 4$ and $20 \times 20 \times 20$ matrices for DNA and amino acid sequences, respectively.

If we denote by i_x the index of symbol x in a matrix, DP created at j-th iteration is computed by

$$DP^j(i_{v_2}, \cdots, i_{v_p}) = \max_{v_1} \left(DP^{j-1}(i_{v_1}, \cdots, i_{v_{p-1}}) + \sum_i \boldsymbol{\alpha}_i \mathbf{y}_i \phi^v_p(d_i) \right) \tag{4.22}$$

Once we have obtained the optimal features after the algorithm converges, we can track backwards to obtain the substrings corresponding to the optimal features which maximize (4.20) and are added to LP columns during the optimization. To track the optimal substrings, we create a matrix ID with the same dimensionality as DP at each iteration to store the symbol v_1 that maximizes equation (4.22).

Therefore the entry of ID at j-th iteration is

$$ID^j(i_{v_2}, \cdots, i_{v_p}) = \arg\max_{v_1} \left(DP^{j-1}(i_{v_1}, \cdots, i_{v_{p-1}}) + \sum_i \boldsymbol{\alpha}_i \mathbf{y}_i \phi_p^v(d_i) \right) \tag{4.23}$$

Example 4.4 shows the structure of DP and ID and the iterative procedure to look for an optimal length-5 string that ends with AC when $p = 3$.

Example 4.4. When $p = 3$, both DP and ID for amino acid sequences are 20×20 matrices that are indexed by Σ, which can be simplified as[1]

$$\begin{array}{c} \\ A \\ C \\ \vdots \\ Y \end{array} \begin{array}{c} \begin{array}{cccc} A & C & \cdots & Y \end{array} \\ \begin{pmatrix} DP(1,1) & DP(1,2) & \cdots & DP(1,20) \\ DP(2,1) & DP(2,2) & \cdots & DP(2,20) \\ \vdots & \vdots & \ddots & \vdots \\ DP(20,1) & DP(20,2) & \cdots & DP(20,20) \end{pmatrix} \end{array}$$

Let us take an entry $(i_A, i_C) = (1, 2)$ for example. The dynamic programming process involves three iterations as follows:

$$\begin{aligned} \text{Iteration 1: } & DP^1(1,2) = \max_x \sum_i \boldsymbol{\alpha}_i \mathbf{y}_i \phi_p^v(d_i) \\ & ID^1(1,2) = x^* \\ \text{Iteration 2: } & DP^2(1,2) = \max_x \sum_i \boldsymbol{\alpha}_i \mathbf{y}_i \phi_p^v(d_i) + DP^1(i_x, 1) \\ & ID^2(1,2) = x^* \\ \text{Iteration 3: } & DP^3(1,2) = \max_x \sum_i \boldsymbol{\alpha}_i \mathbf{y}_i \phi_p^v(d_i) + DP^2(i_x, 1) \\ & ID^3(1,2) = x^* \end{aligned}$$

where $v = xAC$ and x^* denotes the optimal symbol that gives the maximum.

The pseudocode of the dynamic programming approach is given in Algorithm 4.1. Algorithm 4.2 gives the pseudocode to track backwards to recover the

[1] Matrix ID has exactly the same structure as DP. Σ refers to the amino acid alphabet (Table B.1 in Appendix B).

optimal substrings. The complete algorithm to solve the LPBoost from string and Fisher kernel features is given in Algorithm 4.3.

Algorithm 4.1 matrix_dp($\phi_p, \mathbf{y}, \hat{\boldsymbol{\alpha}}, l, A$): dynamic programming for optimal base learners from matrix-represented features

Input: base feature space of p-spectrum kernels ϕ_p, labels $\mathbf{y}$, $\hat{\boldsymbol{\alpha}}$ solved by last LP optimization, length l, set of substrings $A = \bigcup \Sigma^p$
Initialize
$DP^0 \leftarrow \mathbf{0}$
Dynamic programming
for $k = 1$ **to** $l - p + 1$ **do**
  for $v \in A$ **do**
    $key = DP^{k-1}(i_{v_1}, \ldots, i_{v_{p-1}}) + \sum_i \hat{\boldsymbol{\alpha}}_i \mathbf{y}_i \phi_p^v(d_i)$
    if $key > DP^k(i_{v_2}, \ldots, i_{v_p})$ **then**
      $DP^k(i_{v_2}, \ldots, i_{v_p}) \leftarrow key$
      $ID^k(i_{v_2}, \ldots, i_{v_p}) \leftarrow v_1$
    end if
  end for
end for
Return DP, ID

Algorithm 4.2 matrix_track(ID, DP): backward tracking of optimal string

Input: ID, DP
Initialize
$s \leftarrow \epsilon^l$, ϵ^l being an empty string of length l
$k \leftarrow 0$
Track backwards
$\mathbf{i} \leftarrow \arg\max_{\mathbf{i}=\{i_1,\ldots,i_{p-1}\}} DP^{l-p+1}(i_1, \ldots, i_{p-1})$
$s[l - p + 2 : l] \leftarrow \Sigma_{\mathbf{i}}$
$s[l - p + 1] \leftarrow ID^{l-p+1}(i_1, \ldots, i_{p-1})$
while $l - p - k > 0$ **do**
  $s[l - p - k] \leftarrow ID^{l-k-1}(i_{s_{l-p-k+1}}, \ldots, i_{s_{l-k-1}})$
  $k \leftarrow k + 1$
end while
Return s

4.3.2 Graph-represented data

Algorithms 4.1-4.3 are based on matrix-represented features in which the base features are represented by vectors in a numerical matrix. We can further increase the efficiency of the proposed algorithm by using graphs to store the features. We use

Algorithm 4.3 dpLPBoost(ϕ_p, $\mathbf{y}$, ν, l, A): dynamic programming LPBoost

Input: features ϕ_p, label $\mathbf{y}$, regularization parameter ν, length l, set of substrings $A = \bigcup \Sigma^p$
Initialize
$j \leftarrow 0$
$\beta \leftarrow 0$
$\boldsymbol{\alpha} \leftarrow (\frac{1}{m}, \ldots, \frac{1}{m})$
$D \leftarrow \frac{1}{m\nu}$
$S \leftarrow \emptyset$
while $\max_{f \in \mathcal{F}} \sum_{i=1}^{m} \hat{\boldsymbol{\alpha}}_i \mathbf{y}_i f(d_i) \geq \beta$ **do**
  $j \leftarrow j + 1$
  $DP, ID \leftarrow$ matrix_dp(ϕ_p, $\mathbf{y}$, $\boldsymbol{\alpha}$, l, A)
  $S \leftarrow S \cup$ matrix_track(ID, DP)
  $f_j(d_i) \leftarrow \max DP$
  $F_{ij} \leftarrow f_j(d_i)$
  Solve (4.19)
end while
$\mathbf{w} \leftarrow$ Lagrange multipliers from last LP solution
Return w

directed graphs to represent the feature space of p-mer based features. Each base feature is represented by a single transition or an edge in the graph, and each node represents a substring of length $p - 1$. The feature vector of a base feature is thus assigned as an attribute of the corresponding edge. Figure 4.1 illustrates a directed graph of a Markov string, where each $v_i \in \Sigma^{p-1}$, $x_i \in \Sigma$ and v_{i-1} transfers to the next node v_i by adding an extra symbol x_i at the end, such that $v_i = v_{i-1}[2 : p-1]x_i$. An example of a subgraph with all triplets that occur in the aforementioned sequence GACACAGACAC is shown in Figure 4.2. Please note that although Figure 4.2 consists of all triplets ($p = 3$) in the given sequence, other nodes and edges are also required to construct the features for $l > 3$, as longer subsequences may also contain triplets that do not appear in the sequence.

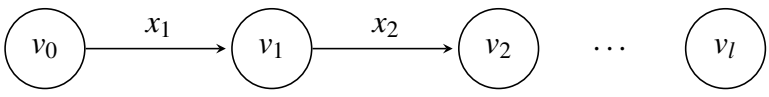


Figure 4.1: Graph representation of a Markov string

Although the graph-represented feature space has the same number of feature vectors and occupies similar memory space as the matrix-represented feature space,

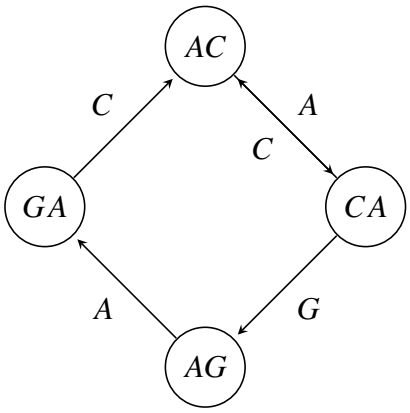


Figure 4.2: Graph representation of existing triplets in DNA sequence GACACAGACAC.

this approach potentially provides faster solutions and greater convenience from the structure of directed graphs. First, it is more straightforward to compute the transition probabilities in Fisher kernel features by rewriting equation (4.8) as

$$\mathcal{P}_{u\to x} = \frac{\mathrm{tf}(ux, d)}{\sum_{x\in \mathrm{ch}(u)} \mathrm{tf}(ux, d)} \tag{4.24}$$

where $\mathrm{ch}(u)$ denotes the set of all children of u. Second, in the dynamic programming approach, it is easier to look for the optimal string in the current iteration (length l') by traversing through all the parent nodes of every node and make use of the results at $l' - 1$ from the parent nodes. Finally, at each iteration, we can assign every node with an additional attribute that stores the particular parent node that maximizes (4.20) within all subsequences of lengths from p to l that end at this node. This offers future convenience when tracking backward to recover the optimal substrings after convergence.

Analogous to the matrix-represented approach, we propose similar algorithms to create feature columns that maximize the base learner and to track optimal substrings as in Algorithms 4.4 and 4.5. Since base features can be assigned as attributes in the input graph, additional feature matrices are not required. Instead, we denote $\phi((v_{j-1}, v_j), d_i)$ the base feature of transition $v_{j-1} \to v_j$ in the i-th sample. We also denote $G.V$ the set of vertices and $G.E$ the set of edges of graph G. Algorithms 4.3 remains the same, except that maxtrix_dp and matrix_track are replaced with Algorithms 4.4 and 4.5, respectively.

Algorithm 4.4 graph_dp($G, \mathbf{y}, \hat{\alpha}, l$): dynamic programming for graph-represented features

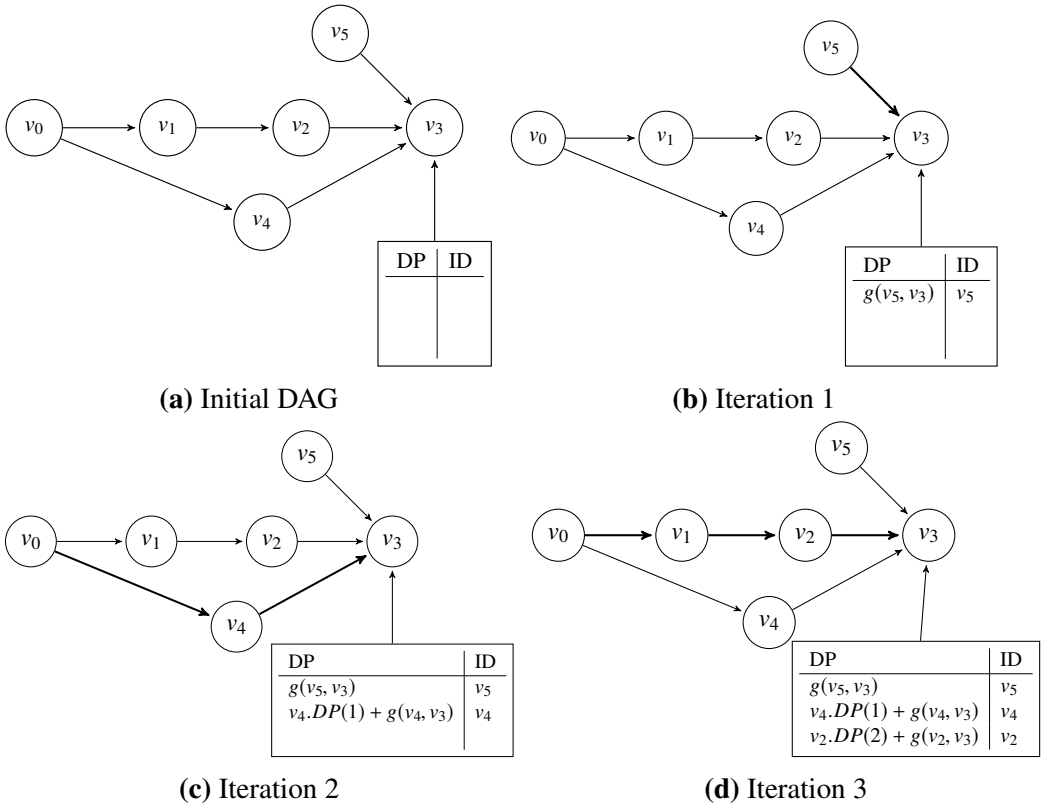
Input: directed graph G, labels $\mathbf{y}$, $\hat{\alpha}$ solved by last LP optimization, length l
Initialize
for $v \in G.V$ **do**
  $v.DP(0:l) \leftarrow 0$
  $v.ID(0:l) \leftarrow \epsilon$
end for
Dynamic programming
for $k = 1$ **to** l **do**
  for $(v_{j-1}, v_j) \in G.E$ **do**
    $key \leftarrow v_{j-1}.DP(k-1) + \sum_i \hat{\alpha}_i \mathbf{y}_i \phi((v_{j-1}, v_j), d_i)$
    if $key > v_j.DP(k)$ **then**
      $v_j.DP(k) \leftarrow key$
      $v_j.ID(k) \leftarrow v_{j-1}$
    end if
  end for
end for
Return G

Algorithm 4.5 graph_track(G, l): backward tracking of optimal path

Input: graph G, length l
Initialize
$k \leftarrow l$
$v \leftarrow \arg\max_{v \in G.V} v.DP(k)$
$P \leftarrow v$
Track backwards
while $k > 0$ **do**
  $v \leftarrow v.ID(k)$
  $P \leftarrow (v, P)$
  $k \leftarrow k - 1$
end while
Return P

Figure 4.3 illustrates the process of the dynamic programming approach on graph-represented features to look for subsequences corresponding to three transitions that ends at node v_3.

One additional advantage of the graph-based approach is that it offers further convenience and flexibility to edit the substrings in the graph and provides more biologically meaningful solutions to look for varying-length substrings, which will be discussed in detail in Chapter 5.



(a) Initial DAG **(b)** Iteration 1

(c) Iteration 2 **(d)** Iteration 3

Figure 4.3: An example of dynamic update for vertex v_3, with length of path fixed to 3.

4.4 Experimental results

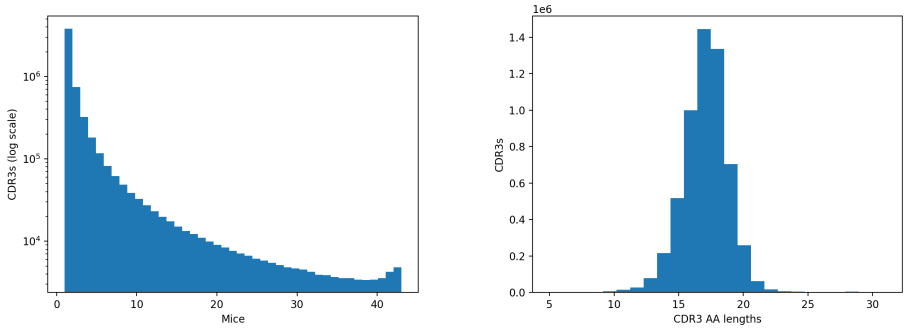
4.4.1 Data description

A total of 33 mice were immunized with complete Freund's adjuvant (CFA), an extract of *Mycobacterium tuberculosis*. CFA contains a complicated mixture of various antigens and is believed to induce widespread innate and adaptive immune responses. Ovalbumin (OVA), a commonly used protein to stimulate allergic responses in immunology studies, was also applied to some of the mouse samples. Among the immunized mouse set, five mice were also immunized with HSP60 (amino acid sequence VLGGGCALLRCIPALDSLTPANED, known as P277). In addition to the immunized mice, 10 control mice that were either untreated or injected with phosphate-buffered saline (PBS) were also analysed. CD4+ T cells were isolated from mouse spleens and CDR3β sequences were sampled from mice at various time points from Day 0 to Day 60. Sequences were processed by Decombinator [195]. Table 4.4 summarizes the details of the mouse dataset. The dataset

contains a total of 5,713,036 unique CDR3β sequences, with an average of 355,103 ± 201,429 unique CDR3βs per mouse. In each mouse sample, the actual number of CDRβs ranges from 136,434 to 12,718,758, with an average of 3,285,002 ± 2,573,324 CDRβs. Most CDR3βs are private (Figure 4.4a), which are found in only one mouse. The lengths of CDR3β sequences range from 5 to 31 amino acids, while as many as 78.5% CDR3β sequences consist of 16–19 amino acids. Figure 4.4b shows the distribution of lengths of CDR3βs in the dataset.

Table 4.4: Description of the mouse dataset.

	Day 0	Day 5	Day 7	Day 10	Day 14	Day 60	Total
CFA+OVA	0	3	3	0	3	6	15
CFA+PBS	0	3	2	0	3	5	13
CFA+P277	0	0	0	5	0	0	5
Untreated	8	0	0	0	0	0	8
PBS	0	1	0	0	0	1	2
Total	8	7	5	5	6	12	43



(a) **(b)**

Figure 4.4: Distribution of CDR3β sequences in the mouse dataset. (a). Histogram of CDR3βs in logarithm scale. Most CDR3βs are private and are not shared among mice. (b). Histogram of lengths of CDR3βs.

4.4.2 Results

The computational pipeline can be categorized into three steps: data processing, feature selection, and classification. In the data processing step, the feature space is constructed either with string kernel or Fisher kernel as introduced in Section 4.2. We set $p = 3$ and $l \in \{3, 5, 7\}$, corresponding to 1, 3, and 5 transitions on a graph.

The logarithm of transition probabilities is used instead of transition probabilities, as discussed in Section 4.2.2. To avoid zero log probabilities as denominators, we add 1 to both the numerator and denominator when computing transition probabilities. All data entries are preprocessed by ℓ_2 normalization. For simplicity, we do not take sequence alignment into account. Nodes in the graph are duplets and a transition represents a triplet.

In the feature selection step, we run Algorithm 4.3 but with a graph structure to dynamically select features of fixed lengths (3, 5, and 7, or 1, 3, and 5 transitions) with LPBoost. We then perform binary classification by either LPBoost or ℓ_2-regularized SVM with RBF kernel, using only the selected features to predict the immune status (control mice vs. immunized mice, or OVA mice vs. non-OVA mice). Nested leave-one-out cross-validation is used to tune parameters (ν in LPBoost and C and γ in RBF SVM) and leave-one-out validation accuracy is reported. We set $\nu \in \{0.1, 0.2, \cdots, 0.9\}$ in LPBoost. For SVM, we first test a larger range of parameters and then determine to use the following parameters in our experiment. We set box constraint $C \in \{1, 2, \cdots, 100\}$, $\sigma \in \{0.1, 0.2, \cdots, 10\}$ and compute $\gamma = \frac{1}{2\sigma^2}$ in the RBF kernel. All experiments are repeated on 11 subsets of randomly selected CDR3β sequences.

First, we examine the performance on classification between immunized and control mice using 50,000 randomly selected CDR3β sequences from each subset. The average accuracy is shown in Table 4.5. All methods report at least 90% accuracy using both string and Fisher kernel features. We then undertake the more challenging task of classifying between OVA mice and non-OVA mice. Two time points are considered: early (Days 5/7/10/14) and all (Days 5/7/10/14/60). Theoretically, early mice are more likely to experience more active immune responses compared to the late mice whose samples were taken 60 days after immunization. We test two types of CDR3βs: random 50,000 CDR3βs and top 5% CDR3βs. For top 5% CDR3βs, experiments are based on a single set, instead of 11 subsets of randomly selected CDR3βs. Features are again generated from string and Fisher kernel features and are ℓ_2-normalized. Tables 4.6 and 4.7 show the leave-one-out

Table 4.5: Mean classification accuracy on immunized and control mice with 50,000 random CDR3βs.

String/LP	String/LP-SVM	Fisher/LP	Fisher/LP-SVM
0.9302 ± 0.00	0.9556 ± 0.01	0.9260 ± 0.01	0.9302 ± 0.00

validation accuracy with top 5% and random 50,000 CDR3βs, respectively.

Table 4.6: Classification accuracy on immunized mice with top 5% CDR3βs.

(a) Early mice

	String/LP	String/LP-SVM	Fisher/LP	Fisher/LP-SVM
#tran = 1	0.5882	0.7647	0.6471	0.7059
#tran = 3	0.5294	0.6471	0.6471	0.7647
#tran = 5	0.5294	0.6471	0.5882	0.7647

(b) All immunized mice

	String/LP	String/LP-SVM	Fisher/LP	Fisher/LP-SVM
#tran = 1	0.5758	0.7879	0.6970	0.7273
#tran = 3	0.5152	0.5455	0.6970	0.7576
#tran = 5	0.6970	0.6970	0.7576	0.7273

Table 4.7: Mean classification accuracy on immunized mice with 50,000 random CDR3βs.

(a) Early mice

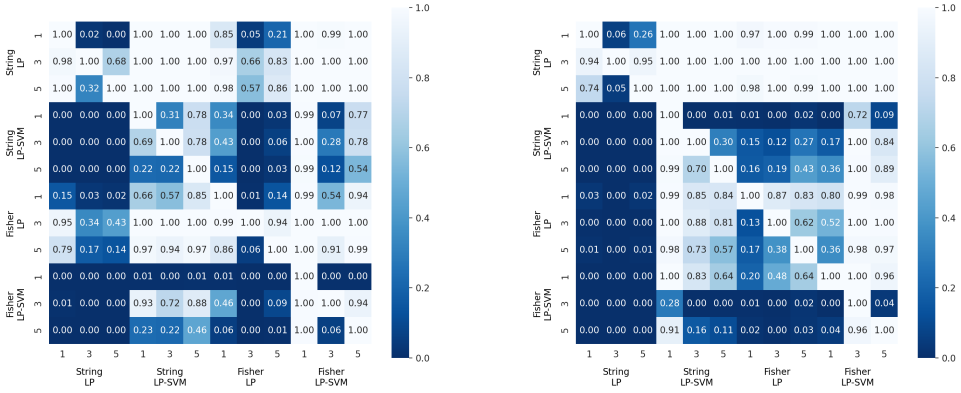
	String/LP	String/LP-SVM	Fisher/LP	Fisher/LP-SVM
#tran = 1	0.7059 ± 0.04	0.7754 ± 0.03	0.7487 ± 0.11	0.8396 ± 0.06
#tran = 3	0.6631 ± 0.04	0.7701 ± 0.03	0.6631 ± 0.08	0.7540 ± 0.04
#tran = 5	0.6631 ± 0.04	0.7807 ± 0.04	0.7112 ± 0.11	0.7914 ± 0.07

(b) All immunized mice

	String/LP	String/LP-SVM	Fisher/LP	Fisher/LP-SVM
#tran = 1	0.6722 ± 0.06	0.8072 ± 0.02	0.7410 ± 0.07	0.7741 ± 0.03
#tran = 3	0.6364 ± 0.03	0.7713 ± 0.03	0.7548 ± 0.04	0.8127 ± 0.03
#tran = 5	0.6667 ± 0.05	0.7658 ± 0.03	0.7576 ± 0.07	0.7851 ± 0.05

Not surprisingly, it is more difficult to distinguish mice immunized with different antigens than to distinguish between immunized and control mice. The use of top 5% CDR3βs does not bring any advantage. Indeed, performance on randomly selected CDR3βs is generally better than on top 5% CDR3βs. Such difference is

more significant when using early mice. To compare the performance of different features, we conducted a one-tailed Wilcoxon signed-rank test [196, 197] on the validation accuracies using randomly selected CDR3βs (Figure 4.5). Unfortunately, using longer subsequences does not induce higher classification accuracy in most experiments. Triplets (# tran = 1) already achieve similar or higher accuracy. Given the satisfactory performances of triplets, we further repeat the experiments with CDR3βs that occur only once within each mouse for all tasks. The average classification accuracy from 11 subsets of 50,000 randomly selected CDR3β singlets is reported in Table 4.8. Most experiments have increased or similar accuracy with CDR3β singlets, except for Fisher kernel features on early mice. Overall, Fisher kernel features perform better than string kernel features in most experiments. The higher accuracy achieved using singlets may indicate the extreme privacy of the T cell repertoire. We further report the selected triplets using random CDR3βs or CDR3β singlets in Appendix C.



(a) Early mice **(b)** All immunized mice

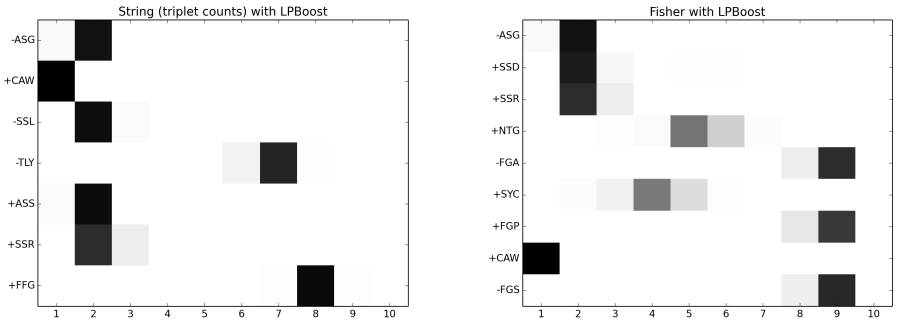
Figure 4.5: p-values of one-tailed Wilcoxon signed-rank test using randomly selected CDR3βs. Alternative hypothesis assumes the algorithms on the x axis (horizontal) achieve higher accuracies than those on the y axis (vertical). LPBoost in conjunction with SVM (LP-SVM) performs significantly better than LPBoost alone under most circumstances.

When leaving different mice out, the selected subsequences are slightly different. Figure 4.6 shows the relevant positions of the most commonly selected subsequences along CDR3β sequences. A "negative" subsequence is considered to

Table 4.8: Mean classification accuracy on immunized and control mice with CDR3βs singlets.

	String/LP	String/LP-SVM	Fisher/LP	Fisher/LP-SVM
Early	0.7701 ± 0.07	0.8396 ± 0.06	0.6631 ± 0.07	0.7433 ± 0.07
Immunized	0.8209 ± 0.03	0.8788 ± 0.04	0.8237 ± 0.04	0.9063 ± 0.04
All	0.9302 ± 0.00	0.9323 ± 0.01	0.9450 ± 0.01	0.9471 ± 0.01

be OVA-predictive, while a "positive" subsequence is considered to be relevant for classifying non-OVA mice. Surprisingly, those commonly selected subsequences are more likely to occur at the beginning or the end of CDR3βs. This may correspond to the germline sequences of V or J segments, although the true mechanism behind this is still unexplained. This result may also indicate the actual interacting regions between TCRβ and MHC-peptide complex.



(a) String kernel features **(b)** Fisher kernel features

Figure 4.6: Positions of subsequences from early mice

4.5 Conclusion

In this chapter, we present a novel machine learning pipeline to efficiently extract strings of fixed lengths from biological sequences, which are represented by subsequences and Fisher kernel features. We develop fast dynamic programming approaches to solve LPBoost using both matrix and graph-represented feature space. The proposed pipeline is applied to CD4+ CDR3β sequences to distinguish control mice and mice immunized with multiple antigens including CFA, OVA, and P277. The use of LPBoost in conjunction with SVM shows a significant advantage over

LPBoost alone in most experiments. This is a not surprising result. Previous studies have demonstrated that ℓ_1-regularized methods give sparse solutions at the cost of over shrinkage of weights, and it is suggested to fit unconstrained linear models such as OLS to the selected features [83]. Fitting Lasso with parameters that are optimal for prediction does not give model selection consistency [198]. Indeed, true features are usually a subset of features selected by Lasso when Lasso includes too many features, and this motivates a second step to fit constrained models to the selected features. Such a two-step process leads to relaxed Lasso [116] (also see Section 3.3.5) and other similar methods such as adaptive Lasso [114] and a combination of various algorithms [199]. Similar two-step approaches have also been applied to biology studies for biomarker selection [115, 200].

Our experimental results emphasize the conspicuous diversity and heterogeneity of the T-cell repertoire in the adaptive immune response, which is distinguished with only a small subset of short motifs. Both triplets and longer strings that consist of triplet-based features achieve satisfactory performances. Indeed, our previous studies have also shown exclusive advantages using triplets [71, 201, 202]. It may be surprising that the selected triplets occupy relatively specific locations along CDR3β sequences. Results show that OVA and CFA-classifying triplets occur mostly at the beginning or the end of sequences. Some popular triplets such as CAW and ASG, are likely to be found by the end of V and J regions, which may indicate that such regions are important in interacting with peptide fragments presented by MHC molecules. The specificity in locations also implies that the end of V and J regions may encode and determine antigen specificity of TCRs.

To summarize, in this chapter, novel dynamic programming approaches are presented to efficiently extract antigen-specific short motifs from mouse CDR3β sequences. Our work highlights the enormous diversity and privacy of T-cell repertoires. Experimental results provide evidence that short motifs that distinguish between CFA and OVA mice occupy rather specific locations along mouse CDR3β sequences, which indicates that such locations may be crucial in determining the diversity of TCRs. In the next chapter, we will extend the graph-based approach

to automatically determine the lengths required to extract contiguous substrings of varying lengths.

Chapter 5

AutoLPBoost: an automatic method to extend subsequences

In Chapter 4, we have presented efficient dynamic programming algorithms to select fixed-length subsequences using LPBoost. Experimental results on mouse CDR3β sequences have shown satisfactory performance in selecting CDR3β subsequences in various classification tasks. Longer subsequences are indeed the sum of base kernel features of triplets, which do not represent actual contiguous strings of the same lengths. Therefore, it may be difficult to interpret such subsequences from an immunology viewpoint. In this chapter, we extend our algorithms to look for optimal substrings while simultaneously extending the lengths of selected substrings, which has a potential to verify whether using varying-length substrings, rather than fixed-length triplets, improves the classification performances over using triplets alone. We start by introducing the method to extend selected substrings in Section 5.1.1. In Section 5.1.2, we show a more intelligent algorithm which extends and selects substrings from an empty graph. Experimental results on mouse CDR3β sequences are presented in Section 5.2. We conclude with our major contributions and findings in Section 5.3.

5.1 From fixed-length subsequences to varying-length subsequences

5.1.1 PairedLPBoost to extend subsequences on existing graphs

In Chapter 4, selected subsequences are represented by transitions or paths on a graph. Such subsequences represent the sum of either string or Fisher kernel features if they are longer than the base features. Therefore, it is questionable whether actual contiguous substrings from such "joint" motifs have the same or better effects as the sum of discrete substrings. For example, when CASSG, which sums up features indexed by CAS, ASS, and SSG, is selected, we are further interested in how contiguous CASSG performs. This leads to a new approach to merge nodes and split paths on a graph to create new nodes of longer substrings.

This new approach consists of three steps:

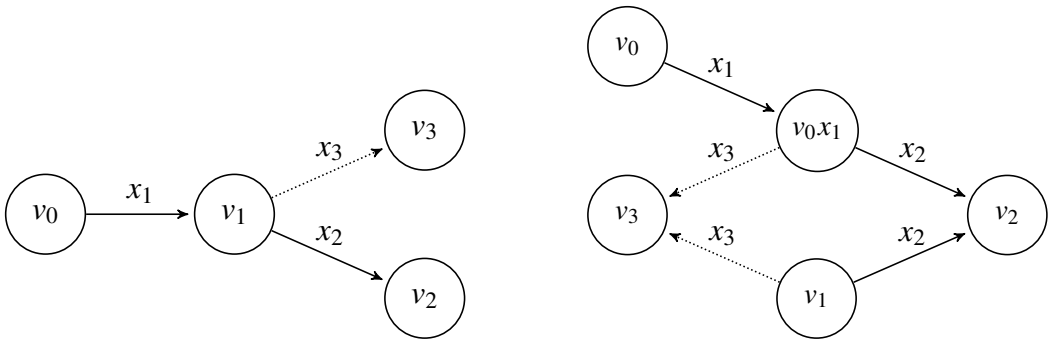
1. Select a pair of transitions ($v_0 \xrightarrow{x_1} v_1 \xrightarrow{x_2} v_2$) that are connected by an edge from the graph using Algorithms 4.4 and 4.5;

2. Merge the paired nodes from Step 1 to create a new node $v_0x_1 = v_0v_1[-1]$ of the substring represented by the transition from paired nodes;

3. Split the transitions from v_1 to its child node v_2 to represent $v_1 \xrightarrow{x_2} v_2$ and $v_0x_1 \xrightarrow{x_2} v_2$. Also update the transitions from v_1 and v_0x_1 to all other child nodes of v_1 as the edge $v_0 \xrightarrow{x_1} v_1$ is deleted.

We then select a single transition that satisfies the base learner of LPBoost and add the corresponding features to the LP columns for further optimization. Figure 5.1 shows how we select a pair of transitions and update the graph before picking up a single transition. For example, once $CA \xrightarrow{S} AS \xrightarrow{W} SW$ is selected, a new node CAS is created and the edge $CA \xrightarrow{S} AS$ is deleted. Instead, this transition is replaced by $CA \xrightarrow{S} CAS$. Therefore, transitions of longer substrings CASx ($CAS \xrightarrow{x} Sx$) are created, where x labels a transition from AS. Apparently, we also need to subtract the feature value of all $CASx$ from the original transitions $AS \xrightarrow{x} Sx$. It is then of

interest whether CASW can be selected. Algorithm 5.1 shows this select-merge-split approach to create longer substrings and the full LPBoost approach is given in Algorithm 5.2.



(a) Initial DAG **(b)** Split the graph

Figure 5.1: An example to split the graph. (a). A pair of transitions $v_0 \xrightarrow{x_1} v_1 \xrightarrow{x_2} v_2$ selected by LP base learner; (b). split the graph by adding a new node v_0x_1 from the transitions in (a). Note that all child nodes (v_3) of v_1 need to be updated even if not selected (shown in dashed arrows).

Algorithm 5.1 PairedLPBoost(G, K)

Input: directed graph G, hash table K of features
Update the graph
$v_1' \leftarrow v_0 + v_1[-1]$
G.add_edge(v_0, v_1')
$\phi(v_0, v_1') \leftarrow \phi(v_0, v_1)$
for $v_2' \in v_1.successors$ **do**
  G.add_edge(v_1', v_2')
  $\phi(v_1', v_2') \leftarrow \phi(v_1, v_2') - K(v_0v_1[-1]v_2[-1])$
end for
G.delete_edge(v_0, v_1)
Return G

5.1.2 AutoLPBoost to grow from empty graphs

Since Algorithm 5.2 generates new nodes and edges in the graph, it is possible to start from an empty graph and construct the graph while selecting useful transitions. An empty graph contains only an uninformative null symbol ϵ and $|\Sigma|$ edges that point from and to ϵ. Each edge represents the feature computed from a single symbol $x \in \Sigma$. In this approach, we also adapt the "select-merge-split" strategy as in Section 5.1.1 to create new nodes and edges in the graph. For example, if

Algorithm 5.2 graph_split(G, K): split the graph

Input: directed graph G, hash table K of features
Select a double transition
$G \leftarrow$ graph_dp($G, 2$)
$(v_0, v_1, v_2) \leftarrow$ graph_track($G, 2$)
$G \leftarrow$ PairedLPBoost(G, K)
Select a single transition
$G \leftarrow$ graph_dp($G, 1$)
$(v_0, v_1) \leftarrow$ graph_track($G, 1$)
Return $G, \phi(v_0, v_1)$

$\epsilon \xrightarrow{x_0} \epsilon \xrightarrow{x_1} \epsilon$ is selected, ϵ and x_0 is merged into new nodes, and the graph is split to create new edges $\epsilon \xrightarrow{x_0} x_0$ and $x_0 \xrightarrow{x_1} \epsilon$, which represent features indexed by x_0 and x_0x_1. Of course, the edge $\epsilon \xrightarrow{x_0} \epsilon$ is deleted. Starting from an empty graph or smaller graphs results in faster solutions as we do not need to traverse all edges in larger graphs. In this way, this approach is generally very fast and converges to very small graphs. Since this approach works with different edges that may be connected by the same pair of nodes, we redefine a unique edge with a pair of nodes v_0 and v_1 and a symbol x_1 that connects both nodes such that $(v_0, v_1, x_1) = v_0 \xrightarrow{x_1} v_1$. Algorithm 5.3 shows the pseudocode for updating the graph using this extended method. The complete optimization follows Algorithm 5.2, in which PairedLPBoost is replaced with AutoLPBoost.

5.2 Experimental results

We validate the performances of the proposed algorithms on the dataset used in Chapter 4: mouse CDR3β sequences sampled from 33 mice that were not immunized or immunized with various complex antigens (CFA, CFA+P277, and CFA+OVA) at various time points after immunization. In Chapter 4, the proposed algorithms provide a high classification accuracy when classifying between immunized and control groups, but a relatively lower accuracy when classifying between the mice immunized with different antigens (OVA-immunized or non-OVA-immunized). Therefore, in this chapter, we focus on the more challenging task to distinguish OVA-immunized and non-OVA-immunized mice. Given the satisfac-

Algorithm 5.3 AutoLPBoost(G, K)

Input: directed graph G, hash table K of features
$v_1' \leftarrow v_0 x_1$
if $v_1' \neq v_1$ **then**
  G.add_edge(v_0, v_1', x_1)
  $\phi(v_0, v_1', x_1) \leftarrow \phi(v_0, v_1, x_1)$
end if
for $v_2' \in v_1.successors$ **do**
  for $x_2' \in G(v_1, v_2')$ **do**
    $u' \leftarrow v_0 x_1 x_2'$
    $u \leftarrow u'[i : |u'|], i = \min\{i : u'[i : |u|'] \in G.V\}$
    G.add_edge$(v_0 x_1, u, x_2')$
    $\phi(v_0 x_1, u, x_2') \leftarrow K(v_0 x_1 x_2')$
    $\phi(v_1, v_2', x_2') \leftarrow \phi(v_1, v_2', x_2') - K(v_0 x_1 x_2')$
  end for
end for
G.delete_edge(v_0, v_1, x_1)
Return G

tory performances on CDR3β singlets, we focus our experiments using randomly selected 50,000 CDR3β sequences, which have been extensively studied in the last chapter. Similar to the experimental settings in Chapter 4, we generate 11 subsets of random 50,000 CDR3β sequences from each mouse. Two types of "starting graph" are considered. First, we examine LPBoost starting from a graph in which nodes are consisted of duplets using Algorithm 5.1. In this case, each transition connected by a pair of nodes represents a triplet. As discussed in Chapter 4, using triplets alone achieves high accuracy in most experiments. Therefore, it is questionable whether longer substrings of varying lengths can achieve similar or even higher accuracy. Second, we test the proposed algorithms with an empty graph, which consists a null node ϵ and edges that represent features of single amino acids. We apply Algorithm 5.3 to allow the graph to automatically grow and select optimal substrings simultaneously. The use of the empty graph saves significant computational time but requires little information at the starting point. SVM with an RBF kernel is applied on top of LPBoost, and parameters are tuned using nested leave-one-out cross-validation. Considering the small size of the dataset, we report the leave-one-out cross-validation accuracy in Table 5.1.

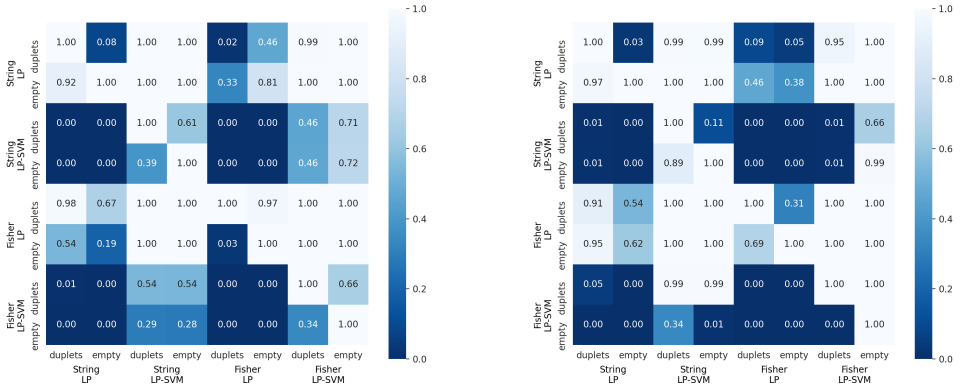
Table 5.1: Mean classification accuracy on immunized mice with 50,000 random CDR3βs.

(a) Early mice

	String/LP	String/LP-SVM	Fisher/LP	Fisher/LP-SVM
Duplets	0.6952 ± 0.08	0.8342 ± 0.07	0.6203 ± 0.07	0.8289 ± 0.06
Empty	0.6417 ± 0.08	0.8342 ± 0.03	0.6845 ± 0.05	0.8396 ± 0.03

(b) All immunized mice

	String/LP	String/LP-SVM	Fisher/LP	Fisher/LP-SVM
Duplets	0.6832 ± 0.10	0.7934 ± 0.03	0.6309 ± 0.04	0.7383 ± 0.03
Empty	0.6253 ± 0.04	0.7741 ± 0.03	0.6171 ± 0.05	0.8017 ± 0.03



(a) Early mice **(b)** All immunized mice

Figure 5.2: p-values of one-tailed Wilcoxon signed-rank test using randomly selected CDR3βs. Alternative hypothesis assumes the algorithms on the x axis (horizontal) achieve higher accuracies than those on the y axis (vertical).

We compared the performances of different algorithms and features with Wilcoxon signed-rank test, as shown in Figure 5.2. Compared with Table 4.7, both methods proposed in this chapter achieve similar or higher classification accuracy in most experiments, except for those using Fisher kernel features with LPBoost. Again, the use of SVM significantly improved the classification accuracy in all experiments, which is not a surprising result considering the possible over-shrinkage using ℓ_1-regularized methods and good performances of two-step methods in previous studies, as discussed in Section 4.5. Selected substrings are shown in Table D.1. In the second setting in which a graph of duplets is used, the majority of selected substrings are still triplets. In the second setting which starts from an empty graph,

it is surprising to find out that shorter substrings, including singlets and duplets, perform very well. The performance under this setting, however, relies on "localized" features where such features are selected only in a few subsets, instead of being selected by the majority of the 11 subsets. This is why we report a larger set of substrings in Table D.1, as only a few substrings are selected by the majority of the random subsets.

AutoLPBoost that starts from an empty graph has significant advantages in terms of memory. Table 5.2 shows the average numbers of nodes and edges in the final graphs after convergence. The graphs used under the first setting and in Chapter 4 start from 400 nodes and 8,000 edges which capture all possible triplets generated from 20 amino acids. However, AutoLPBoost starts from an empty graph with a null node and 20 edges, while finishes with approximately 15 nodes and less than 300 edges, creating a much smaller space to work with.

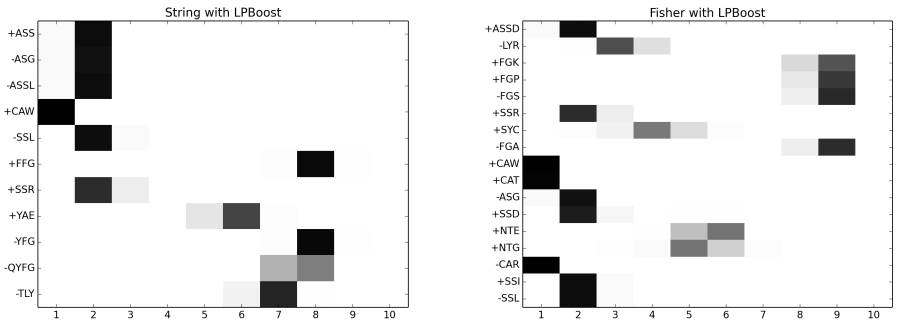
Table 5.2: Average number of nodes and edges in the graph using AutoLPBoost. Generated from an empty graph using early mice.

# Node/String	# Edge/String	# Node/Fisher	# Edge/Fisher
14.62	292.30	14.97	299.36

Figure 5.3 shows the relevant position of the selected substrings along CDR3β sequences using the early mice under the first setting. Similar to the results shown in Figure 4.6, most substrings occur towards the beginning or the end along CDR3β sequences.

5.3 Conclusion

In this chapter, we present two extended algorithms that are based on the dynamic programming algorithms in Chapter 4. The proposed algorithms, PairedLPBoost and AutoLPBoost, automatically add new nodes and edges to the graph as well as selecting optimal substrings that maximizes the base learner. This process has several distinct advantages over the algorithms presented in Chapter 4. First, both algorithms can extend substrings in the graph during the column generation process and select substrings of varying lengths. Unlike the algorithms introduced in Chap-



(a) String kernel **(b)** Fisher kernel

Figure 5.3: Positions of substrings from early mice

ter 4, in which the number of transitions or the length of subsequences needs to be predefined, the proposed algorithms do not have restrictions over the lengths of substrings. Second, the proposed algorithms combine substrings selected from the last iteration, leading to longer contiguous substrings in the graph. Such substrings are generally more biologically meaningful and are easier to interpret. Third, the proposed algorithms indeed select contiguous substrings from a potentially larger feature space, compared to the algorithms presented in Chapter 4. For example, if we use triplets as base features in Algorithm 4.4, the longest substring we can obtain is triplet, while with Algorithm 5.1, we can potentially obtain substrings from length 3 to infinity—although the algorithm usually arrives at the optimal solution before very long substrings are added to the graph. Finally, this grow-and-select strategy only requires very small graphs to start with, which significantly saves the memory space. Experimental results show that starting from an empty graph only results in a graph with less than 15 nodes and 300 edges after the optimization. One may argue that the features computed from longer substrings still need to be stored locally. However, such information can be stored in several hash tables in parallel, and is only retrieved when relevant substrings are added to the graph. Therefore, only small graphs are required under most circumstances at relatively low computational cost.

The proposed methods are validated on mice CDR3β sequences, which are also used for the experiments in Chapter 4. Experimental results show that with a

small number of short substrings, similar or higher performances can be achieved in most experiments. Substrings selected in this chapter are rather "localized": unlike some "popular" substrings which are favored by most random subsets in Chapter 4, different subsets have preferences over different sets of substrings. As a result, many substrings are selected by only a few random subsets of CDR3βs. This finding indeed strengthens the conclusion on the privacy of T-cell repertoires. However, it is still undetermined whether the selected substrings generalize well on larger datasets of mice CDR3βs in response to the same antigens.

The selected substrings are more likely to occur at both ends along a CDR3β sequence, which coheres with our findings in Chapter 4. Although this may result from the germ line sequences of V, D, and J regions, such result again suggests that there are structural constraints to generating functional CDR3β and indicates possible interacting regions between TCRβ sequences and MHC-peptide complex.

Similar to Chapter 4, the combination of LPBoost and SVM performs better in all experiments. As discussed in Chapter 4, similar two-step procedures have been applied to many studies to avoid weights estimated by ℓ_1-regularized methods to be biased towards zero. Therefore, we conclude that it is advisable to apply some two-step procedures to ensure both feature selection consistency as well as prediction accuracy in the proposed pipelines.

So far, our work has focused on identification of CDR3β subsequences. The extreme privacy of TCR and T-cell repertoire addresses a problem in identifying entire CDR3β or even TCRβ sequences: if most of the sequences are private and thus cannot be used for classification, does there exist a method to select public sequences that are found in multiple individuals? Compared to private sequences, public sequences have higher prediction power in diagnosis and can be used to design simpler, yet powerful diagnostic strategies. However, there are two major problems in identification of entire sequences: increased dimensionality and correlation between antigen-specific sequences. In the next chapter, we will readdress this problem and provide effective methods to study ultra-high dimensional correlated features.

Chapter 6

$\ell_{1,2}$-regularization in identifying entire sequences

6.1 Overview

In Chapters 4 and 5, we have presented novel computational pipelines to represent features with directed graphs and extract antigen-specific subsequences or substrings from mouse CDR3β sequences. The feature selection relies on LPBoost, an ℓ_1-regularized method. As introduced in Chapter 3, ℓ_1-regularized methods such as Lasso achieve sign consistency under stringent conditions including the irrepresentable condition, which is easily violated in practice. Previous studies showed that even if the strict irrepresentable condition is satisfied, the optimal regularization parameters for achieving sign consistency are not ideal for prediction [199, 203]. Compared to ℓ_1-regularized algorithms for regression purposes, much less has been studied regarding the model selection consistency for ℓ_1-regularized classification algorithms, such as ℓ_1-SVM, mainly because of the non-smooth loss function in the optimization problems. Although little is known about variable selection consistency for such algorithms, some theoretical as well as experimental work have shown that similar to Lasso and its generalizations, feature selection accuracy tends to be lower when the correlation between features is high [204, 205].

In this chapter, we explore the exclusive group Lasso introduced in Section 3.3.4 for both regression and classification. We examine the experimental con-

ditions for the exclusive group Lasso to select the correct features under various correlation settings. We present three efficient algorithms in Section 6.3 to efficiently solve the exclusive group Lasso. We also present novel approaches in Section 6.4 by making use of artificial features and stability selection to reduce the uncertainty from random group allocation. A novel pipeline is introduced in Section 6.5 for parameter tuning without nested cross-validation over a range of parameters. We test the performance and compare the results on both synthetic and real-world data using Lasso (ℓ_1-SVM) and elastic net in Section 6.6. Particularly, the proposed pipeline is applied to a more complex problem to infer entire TCRβ sequences that are associated with public T-cell responses to cytomegalovirus (CMV).

6.2 Constructing features

Our previous work on selecting short motifs from biological sequences is based on the string or Fisher kernel features, which measure the frequencies or weighted frequencies of non-contiguous subsequences or contiguous substrings. As the target of this chapter is to select entire biological sequences, the feature construction step is very straightforward: binary representation is used to convert the sequencing samples to numerical feature space. Obviously, as we move from substrings to entire sequences, the dimensionality of the feature space increases because of the extreme diversity and privacy of the T-cell repertoire. For example, the number of amino acid substrings is approximately 3 million even if we sum up all possible substrings from length 1 to 5, while in practice this number significantly reduces thanks to the dynamic programming algorithms and the absence of certain base features. However, as we shall see in Section 6.6, the training data used in this chapter consists of almost 90 million unique TCR β sequences, and this number is even larger if we consider $\alpha\beta$ pairs of TCR sequences or if we include more samples in the pool of training data. Therefore, although the feature construction step is much easier, the problem becomes more difficult and algorithms that are capable of efficient feature selection from ultra-high dimensional data are required.

6.3 Solving exclusive group Lasso

The exclusive group Lasso aims to solve the following optimization problem:

$$\min_{\mathbf{w}} f(\mathbf{w}) + \lambda \sum_{g \in \mathcal{G}} ||\mathbf{w}_g||_1^2 \tag{6.1}$$

In the original paper of exclusive group Lasso, the authors presented an iterative algorithm to solve the optimization problem. As introduced in Section 3.3.4, the algorithm iteratively solves the following problems.

$$\text{Solve } \mathbf{w}: \qquad \nabla_{\mathbf{w}} f(\mathbf{w}) + 2\lambda F\mathbf{w} = 0 \tag{6.2}$$

$$\text{Compute } F: \qquad F_{ii} = \left(\sum_{g \in \mathcal{G}} \frac{I_g^i ||\mathbf{w}_g||_1}{|\mathbf{w}_i|} \right), F_{ij} = 0, i \neq j \tag{6.3}$$

Although the proposed algorithm in [113] is very straightforward and can be applied to all types of convex loss functions for various regression and classification problems, solving the optimization problem is not an easy task. First, $f(\mathbf{w})$ may be not smooth and the computation of its derivative $\nabla_{\mathbf{w}} f(\mathbf{w})$ is not tractable. Second, even if $f(\mathbf{w})$ is convex and differentiable, solving (6.2) may be difficult, particularly because of the unknown loss function $f(\mathbf{w})$. Even in the simplest case when $f(\mathbf{w})$ is square loss, solving (6.2) involves heavy computation of the inverse of a possibly large matrix in every iteration. For other loss functions, some convex optimization solvers, such as cvxopt [206] and cvxpy [207, 208] in Python and CVX [209, 210] in Matlab, may be used to solve the optimization problem, but the optimization process is again very slow and does not adapt to high-dimensional data. In this section, we present three efficient and easy-to-implement approaches to solve the exclusive group Lasso for both classification and regression problems.

6.3.1 Transforming features and weights

Consider problem (6.1), which can be rewritten as

$$\min_{\mathbf{w}} f(X\mathbf{w}, \mathbf{y}) + \lambda \mathbf{w}^T F \mathbf{w} \tag{6.4}$$

where we replace $f(\mathbf{w})$ with $f(X\mathbf{w}, \mathbf{y})$ for future convenience. This modification does not make any changes to the problem itself. We notice that the second term $\mathbf{w}^T F\mathbf{w}$ can be rewritten as a quadratic form of $\sqrt{F}\mathbf{w}$ such that

$$\mathbf{w}F\mathbf{w} = \tilde{\mathbf{w}}^T\tilde{\mathbf{w}} \tag{6.5}$$

where $\tilde{\mathbf{w}} = \sqrt{F}\mathbf{w}$. In this way, we convert problem (6.4) to an ℓ_2-regularized problem, if we substitute $\mathbf{w}$ with $\tilde{\mathbf{w}}$ in the loss function. Let $\tilde{X} = X\sqrt{F}^{-1}$, and substitute X and $\mathbf{w}$ with $\tilde{X}$ and $\tilde{\mathbf{w}}$, respectively, the exclusive group Lasso can be written as

$$\min_{\tilde{\mathbf{w}}} f(\tilde{X}\tilde{\mathbf{w}}, \mathbf{y}) + \lambda\tilde{\mathbf{w}}^T\tilde{\mathbf{w}} \tag{6.6}$$

As F is a diagonal matrix, the computation of $\sqrt{F}$ and $\sqrt{F}^{-1}$ is very easy. Problem (6.6) can be easily solved by ℓ_2-regularized solvers in standard packages or libraries. For example, if $f(\tilde{X}\tilde{\mathbf{w}}, \mathbf{y})$ is the square loss, solving (6.6) is equivalent to solving a ridge regression problem, hinge loss leads to a standard SVM, and logistic loss requires logistic regression with an ℓ_2 penalty. The procedure to solve the exclusive group Lasso is as follows:

1. Update F with $\mathbf{w}$ via (6.3);
2. Update $\tilde{X}$ with X and F via $\tilde{X} = X\sqrt{F}^{-1}$;
3. Update $\tilde{\mathbf{w}}$ with $\tilde{X}$ by solving problem (6.6);
4. Update $\mathbf{w}$ with $\tilde{\mathbf{w}}$ and F via $\mathbf{w} = \sqrt{F}^{-1}\tilde{\mathbf{w}}$.

We summarize the pseudocode in Algorithm 6.1.

As both features and weights are iteratively reweighted during the optimization process, we can impose further restrictions to the weights and features. This is a very useful property in biology studies. For example, in immunology, as effector cells proliferate in the adaptive immune response to the cognate antigens, it is likely that the antigen-specific TCRs are more expanded in the samples that undergo immune response to the counter-specific antigens, compared to those that do

Algorithm 6.1 reweightExcl($X, \mathbf{y}, \lambda, I, \mathcal{G}$): Exclusive group Lasso with a reweighting scheme

Input: features space X, labels $\mathbf{y}$, regularization parameter λ, group indicator matrix I, groups $\mathcal{G}$
$\mathbf{w} \leftarrow \frac{1}{n}$
while not converged **do**
  $F_{ii} \leftarrow \frac{1}{|\mathbf{w}_i|} \sum_{g \in \mathcal{G}} I_g^i ||\mathbf{w}_g||_1$
  $\tilde{X} \leftarrow X\sqrt{F}^{-1}$
  $\tilde{\mathbf{w}} \leftarrow \arg\min_{\tilde{\mathbf{w}}} f(\tilde{X}\tilde{\mathbf{w}}, \mathbf{y}) + \lambda \tilde{\mathbf{w}}^T \tilde{\mathbf{w}}$
  $\mathbf{w} \leftarrow \sqrt{F}^{-1} \tilde{\mathbf{w}}$
end while
return $\mathbf{w}$

not. Therefore, it is more biologically meaningful to look for TCRs that are more related to the infected samples in immunological studies. This is easy to implement in Algorithm 6.1 as we only need to add constraints to $\tilde{\mathbf{w}}$ to allow only positive or negative values by adding an additional intercept term when solving (6.6) in the optimization process. The nonzero weights for the other class are set to small numbers to avoid numerical problems. In this way, features that are positively related to one class are selected while the rest of the features are filtered out during the optimization. We will see how exclusive group Lasso with this modification outperforms ℓ_1-SVM and elastic net in real-world immunology studies in Section 6.6. Algorithm 6.2 summaries the pseudocode of the approach which adds restrictions to weights and features.

Algorithm 6.2 restrictedExcl($X, \mathbf{y}, \lambda, I, \mathcal{G}, c$): exclusive group Lasso with restrictions on weights

Input: features space X, labels $\mathbf{y}$, regularization parameter λ, group indicator matrix I, groups $\mathcal{G}$, class to be restricted $c \in \{-1, +1\}$
$\mathbf{w} \leftarrow \frac{1}{n}$
while not converged **do**
  $\mathbf{w}_{i:\text{sign}(\mathbf{w}_i)=c} \leftarrow \epsilon$, where ϵ is a very small number
  Run a single iteration of Algorithm 6.1
end while
return $\mathbf{w}$

6.3.2 Iterative algorithm

In the previous section, we presented an algorithm to solve the exclusive group Lasso with standard ℓ_2-regularized packages. The major challenge of the above algorithms comes from large intermediate matrices during the optimization process, leading to an increased computational burden, particularly when $m \ll n$. In Section 3.2.3.2, we have reviewed how Lasso and its generalizations, such as elastic net, group Lasso, and sparse group Lasso, are solved efficiently. As discussed earlier, Lasso-like algorithms are particularly suitable for coordinate descent methods—coordinate descent algorithms are applied to Lasso and elastic net while group Lasso and sparse group Lasso enjoy efficient solutions from block coordinate descent methods.

Recalling the general conditions for coordinate descent algorithms to converge in problems $\min_x f(x) + \sum_i h_i(x_i)$, it is not difficult to see that such conditions stand for the exclusive group Lasso, as long as we restrict the loss function to be convex and differentiable. Therefore, hinge loss violates the conditions, while many other types of loss functions, including square loss, squared hinge loss, and logistic loss, still satisfy the conditions for coordinate descent to converge. In this way, the exclusive group Lasso with aforementioned loss functions can be solved with cyclic block coordinate descent methods.

In this section, we focus on exclusive group Lasso with square loss because square loss is extensively used in Lasso-type problems so that exclusive group Lasso with square loss is more comparable with Lasso and its generalizations. Furthermore, solving exclusive group Lasso involves finding the inverse of very large matrices under high dimensional settings. SVD in ridge regression as used in Algorithm 6.1 can improve the efficiency but still requires $O(mn^2)$ or $O(m^2n)$ time, whichever is smaller. In this section, we show a block coordinate descent approach to solve the exclusive group Lasso with standard Lasso packages efficiently.

Rewrite problem (6.1) with square loss, we have

$$\min_{\mathbf{w}} ||X\mathbf{w} - \mathbf{y}||_2^2 + \lambda \sum_{g \in \mathcal{G}} ||\mathbf{w}_g||_1^2 \tag{6.7}$$

The problem can be solved by block coordinate descent methods as individual groups naturally correspond to blocks in the algorithm. Consider a single iteration when we aim to solve problem (6.7) with all blocks fixed except for the group or block g, if we take the derivative w.r.t. $\mathbf{w}_g$ and set the derivative to 0, we have

$$2X_g^T(X_g\mathbf{w}_g + X_{-g}\mathbf{w}_{-g} - \mathbf{y}) + 2\lambda||\mathbf{w}_g||_1\mathbf{s}_g = 0 \tag{6.8}$$

where $\mathbf{s}_g$ is the subgradient of $||\mathbf{w}_g||_1$.

There is no straightforward way to solve this problem because of the $\ell_{1,2}$-norm regularization term, but we can consider an alternative approach to iteratively solve it via Lasso. If we represent $||\mathbf{w}_g||_1$ with z, and replace one of $||\mathbf{w}_g||_1$ in the quadratic form, we have

$$\min_{\mathbf{w}} ||X_g\mathbf{w}_g + X_{-g}\mathbf{w}_{-g} - \mathbf{y}||_2^2 + \lambda z||\mathbf{w}_g||_1 \tag{6.9}$$

for every $g \in \mathcal{G}$. Let $\lambda' = \lambda z$, the above expression can be rewritten as

$$\min_{\mathbf{w}} ||X_g\mathbf{w}_g + X_{-g}\mathbf{w}_{-g} - \mathbf{y}||_2^2 + \lambda'||\mathbf{w}_g||_1 \tag{6.10}$$

In this way, the problem converts to Lasso problem with a new regularization parameter λ'. Apparently, it is not possible to simply solve (6.10), as λ' contains $\mathbf{w}_g$. Instead, we can iteratively solve (6.10) with Lasso under different λ', and check if $\mathbf{w}'_g = \frac{\lambda'}{\lambda}$. The solution is found when $\mathbf{w}_g = \mathbf{w}'_g$.

Theoretically, this approach needs to examine infinite λ' values to obtain an accurate estimation of $\mathbf{w}_g$. However, we can make use of bisection algorithms. In practice, only several values of λ' are required before convergence. The bisection algorithm is given in Algorithm 6.3. The complete algorithm to solve the exclusive group Lasso is summarized in Algorithm 6.4.

6.3.3 A faster algorithm with active set

So far, two different approaches to solve the exclusive group Lasso have been presented. Algorithm 6.1 is generally very fast but apparently, the computational time is mostly restricted by the input feature space. Like most algorithms, for extremely

Algorithm 6.3 Bisection($X_g, \mathbf{y}_g, \lambda, \lambda_1, \lambda_2$): Bisection algorithm to solve $\mathbf{w}_g$

Input: features of current group X_g, labels $\mathbf{y}_g$, regularization parameter λ, lower bound λ_1, upper bound λ_2
$\lambda' \leftarrow \frac{(\lambda_1+\lambda_2)}{2}$
$\mathbf{w}_g \leftarrow \text{Lasso}(X_g, \mathbf{y}_g, \lambda')$
if converged **then**
  return $\mathbf{w}_g, \lambda'$
else if $\lambda' < \lambda||\mathbf{w}_g||_1$ **then**
  return Bisection($X_g, \mathbf{y}_g, \lambda, \lambda_1, \lambda'$)
else
  return Bisection($X_g, \mathbf{y}_g, \lambda, \lambda', \lambda_2$)
end if

Algorithm 6.4 iterExcl($X, \mathbf{y}, \lambda, \lambda_1, \lambda_2, \mathcal{G}$): an iterative approach in solving exclusive group Lasso

Input: features space X, labels $\mathbf{y}$, regularization parameter λ, lower bound λ_1, upper bound λ_2, groups $\mathcal{G}$
$\mathbf{w} \leftarrow \mathbf{0}$
while not converged **do**
  for all $g \in \mathcal{G}$ **do**
    $\mathbf{y}_g \leftarrow \mathbf{y} - X_{-g}\mathbf{w}_{-g}$
    $\mathbf{w}_g \leftarrow \text{Bisection}(X_g, \mathbf{y}_g, \lambda, \lambda_1, \lambda_2)$
  end for
end while
return $\mathbf{w}$

high-dimensional problems, Algorithm 6.1 is rather slow and sometimes may cause memory problems, especially when $m \ll n$ and X is not sparse. Note that the exclusive group Lasso returns sparse feature selection where the majority of features receive zero weights. As λ becomes larger, the solution becomes more sparse. Furthermore, in the process of the optimization, it is unlikely that the estimated weight of a feature becomes nonzero again once it receives zero. Therefore, during the optimization, features with zero weights do not necessarily need to be included in the next iterations. However, since this algorithm is essentially solved by ℓ_2-regularized solvers, the algorithm does not make any use of the sparsity of weights during the optimization.

Algorithm 6.4, on the other hand, provides efficient solutions. The computational time largely depends on the number of groups. Solving the subproblems

with Lasso is very fast, considering the relatively small size of the features that are required in each subproblem. However, since the block coordinate descent loops through groups of features, smaller subproblems correspond to larger group numbers, which generally leads to a slower convergence rate. For example, for a 1000×500 dataset, where X, $\mathbf{w}$, and ϵ are independently generated from standard normal distributions, running Algorithm 6.4 with 50 groups is approximately 2.6 times faster than using 100 groups. Solving the same problem with Algorithm 6.1 is 7 times faster than Algorithm 6.4 (50 groups), while the ratio becomes nearly $\frac{1}{600}$ when the number of features increases to 5000. The latter case benefits from the advantages of sparsity-inducing Lasso solver and block coordinate descent. As a result, it is natural to combine both algorithms to make full use of their advantages with only the active set of features.

The active set of features is the set of all nonzero features. Its complement, the set of all zero features, is unlikely to receive nonzero estimations for weights again in the optimization procedure. The idea of using the active set is simple: after several iterations, the algorithm runs only with features from the active set. We only need to add an additional iteration with the full set of features after convergence using the active set. This is to ensure that the algorithm converges to the optimal solution. To solve the exclusive group Lasso, we first run Algorithm 6.4 for a full cycle with all groups, then we can apply Algorithm 6.1 only to the active set of features until convergence. These steps are repeated iteratively until an overall convergence is reached. In practice, the algorithm converges to the optimal solution within a few iterations. The procedure is summarized below.

1. Run a single iteration over all groups with Algorithm 6.4;
2. Run Algorithm 6.1 until convergence using only the features with nonzero weights from Step 1;
3. Check if overall convergence is achieved by an additional cycle of Step 1. Repeat Step 1 and 2 if not converged.

Running an additional cycle for convergence check may be redundant in most

cases. Alternatively, this can be done by a quick check on equation (6.8). The complete algorithm of this approach is given in Algorithm 6.5.

Algorithm 6.5 activeExcl(X, $\mathbf{y}$, λ, λ_1, λ_2, I, $\mathcal{G}$): exclusive group Lasso with active set

Input: features space X, labels $\mathbf{y}$, regularization parameter λ, lower bound λ_1, upper bound λ_2, group indicator matrix I, groups $\mathcal{G}$
$\mathbf{w} \leftarrow \mathbf{0}$
while not converged overall **do**
  Run a single iteration over all groups of Algorithm 6.4
  Run Algorithm 6.1 on features with nonzero weights until Algorithm 6.1 converges
end while
return $\mathbf{w}$

6.4 Random group allocation

The purpose of applying the exclusive group Lasso to biological problems is to select all correlated features without a natural group structure. Although ℓ_1-regularized methods such as Lasso does not perform well on correlated features, the grouping scheme and ℓ_2-norm at the inter-group level ensures that correlated features can be selected given they are assigned to different groups. Therefore, group allocation is very crucial in exclusive group Lasso. To our knowledge, previous studies have not presented practical guidelines on how to partition groups for the exclusive group Lasso, which makes it very impractical in real-world problems.

As discussed in Chapter 3, several methods have been commonly used in group Lasso when the data does not have a known group structure. A possible approach is to group features based on the correlation between the features, or between the features and the labels. A potential problem is that in the biological context, many biology patterns function in very complex biochemical pathways. For example, genes have different functions when they act together with different sets of genes; the effect of a single SNP may be different or function at weaker level compared to the joint effects when it cooperates with other SNPs; a TCR may bind to multiple antigens; one sequence may not be related to a clinical outcome, but is strongly correlated to another sequence which is related to the same outcome. Therefore, many biological features, in particular ultra-high dimensional features such as TCRs, are

unlikely to have a clear group structure by correlation. In addition, grouping biological features by their correlation may overlook some important information or lead to inaccurate group structure. Another possible approach is to cluster the features by unsupervised clustering algorithms such as k-means and hierarchical clustering. However, such algorithms can be considerably affected by initial states and thus leads to inaccurate results. Moreover, the computational cost for some clustering algorithms under high-dimensional settings can be very high.

Therefore, in this thesis, we exploited random group allocation, which can be used in more general cases since it partitions groups efficiently without any prior knowledge on biology. However, randomly allocating groups suffers from the risk of not selecting all relevant features, if some of them are correlated and coexist in the same group. In addition, irrelevant features are likely to be selected if no relevant features are in the same group. Figure 6.1 shows the probability of relevant or irrelevant features being selected, with and without other relevant features in the same group. The probability is computed by an average over a wide range of λs 100 toy datasets containing 150 irrelevant features and 50 relevant features. For each dataset, exclusive group Lasso with 70 random groups is used, since in practice the number of groups should be slightly larger than the number of relevant features. On the one hand, using a large group number is likely to select more irrelevant features. When the number of groups equals the number of features, the exclusive group Lasso is equivalent to ridge regression, in which all features are selected. On the other hand, if the number of groups is too small, relevant features will inevitably fall into the same groups and compete with one another. Therefore, Figure 6.1 shows an intuition of the empirical probabilities of both relevant and irrelevant features under random groups when the group size is almost ideal to allow most relevant features to be allocated to different groups and most groups to contain at least one relevant feature.

It is clear from Figure 6.1 that, on the one hand, relevant features within the same group compete with one another and may not all be selected while, on the other hand, irrelevant features stand out if their group does not contain any relevant

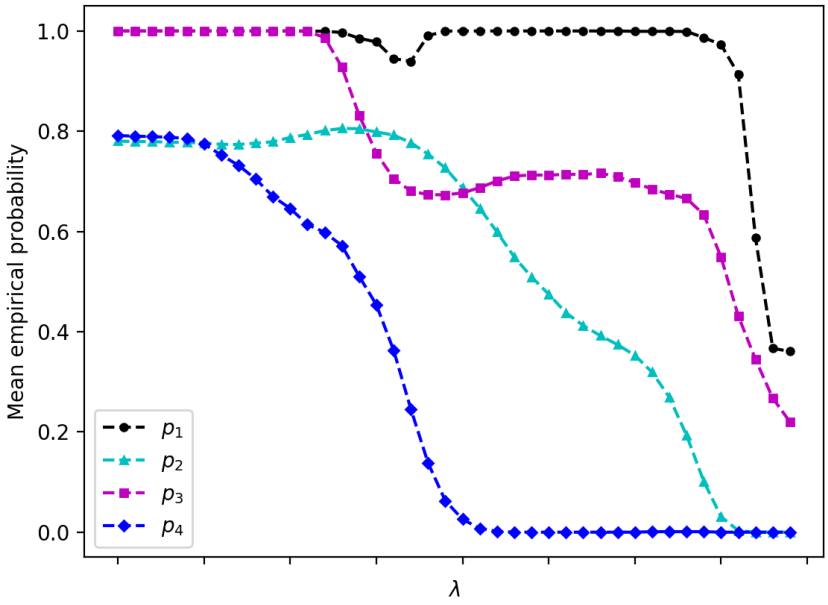


Figure 6.1: Empirical probabilities under various conditions. p_1: Prob[a relevant feature is selected | there are no other relevant features in the same group]; p_2: Prob[an irrelevant feature is selected | there are no relevant features in the same group]; p_3: Prob[a relevant feature is selected | there are other relevant features in the same group]; p_4: Prob[an irrelevant feature is selected | there are at least one relevant feature in the same group].

features. To compensate for the risks from random group allocation, we present two methods in this section: a novel approach using artificial features and a method that combines random group allocation with stability selection.

6.4.1 Artificial features

The ℓ_1 regularization inside groups and ℓ_2 regularization across different groups ensure sparse feature selection at intra-group level in exclusive group Lasso. Unlike the group Lasso, which either selects or deselects entire groups at the same time, the exclusive group Lasso chooses sparse features from every group. When a group does not contain any relevant features, the algorithm still chooses features from this group because of the ℓ_2 regularization at inter-group level. On the other hand, if a group contains at least one relevant feature, the relevant features in this group are selected at high probabilities. Of course, when there are too many relevant

features in one group competing with each other, we again lose the advantage of the exclusive group Lasso and the algorithm tends to perform similarly to Lasso, randomly choosing one or a few relevant features from this group.

Therefore, for the exclusive group Lasso to select correct features, an important condition is that a group must contain only a few, preferably one, relevant features. That is to say, for exclusive group Lasso, the perfect group number is n_S, where we inherit the notation to denote S the support of $\mathbf{w}$ and we let $n_S = |S|$ represent the number of relevant features. Each group contains exactly one relevant feature, plus several irrelevant features. We call this type of allocation the "fixed" group allocation. Practically, when $n_S \ll n$, the exclusive group Lasso is able to select all relevant features with fixed groups even if relevant features are highly correlated. We further examine the performance of the exclusive group Lasso with fixed groups compared to random groups in Section 6.6.

It is not hard to see that although fixed group allocation performs well even when correlations among relevant features are high, this type of group allocation is not possible in practice as it is impossible to know what features are relevant. In contrast to fixed groups, random group allocation is perhaps the most straightforward way to partition groups. However, as discussed earlier in this section, random groups may end up with thorny groups that consist of only irrelevant features or too many relevant features. When the number of relevant features n_S is much smaller than the total number of features n, it is obvious that many groups may only contain irrelevant features. Recognizing biology sequences involves notoriously massive feature spaces which consist of millions of sequences, while perhaps only a few hundreds of them are associated with the research problem. Hence, most groups may end up with a large amount of noisy and useless features and it is not possible for the exclusive group Lasso to select relevant features while excluding the irrelevant ones. Likewise, other high-dimensional feature selection problems with many noisy features such as image classification or text classification also encounter the same problem. Including too many relevant features in one group incurs risks of not being able to select all relevant features because they are competing in one

group. We further discuss the solution to this problem in the next section and in this section, we mainly focus on the problem of including only irrelevant features in a single group.

The remedy to be proposed is in fact very simple and straightforward: manually add a "faked" relevant feature into each group and deceive the algorithm into selecting this "faked" feature when there is no relevant feature in this group. Such "faked" feature is called an "artificial feature." Relevant features can still be selected if the artificial features are generated with proper distributions. Since we know exactly which features are the artificial ones, it is very easy to exclude these artificial features after convergence. Such an approach can reduce the false discovery rate (FDR), as we will show in Section 6.6.

For a regression problem, this approach consists of the following steps.

1. Randomly generate i.i.d. variables $X' \in R^{m \times ns}$, $\mathbf{w}' \in R^{ns}$;

2. Compute new labels $\mathbf{y}' = \mathbf{y} + X'\mathbf{w}'$;

3. Add a feature from X' into each group;

4. Find solution with new features $[X, X']$ and new labels $\mathbf{y}'$;

5. Remove selected artificial features after convergence.

The critical part of this approach is to choose the distribution for generating artificial X' and $\mathbf{w}'$, since too distinct features may conquer the subspace of grouped features and the algorithm may only select the artificial features even when the artificial features and relevant features coexist in the same group. Meanwhile, too feeble features and weights cannot stand out from irrelevant features and this cannot prevent the algorithm from choosing irrelevant features when there are no other relevant features in the same group. In practice, we find Gaussian distributed X' and $\mathbf{w}'$, such as $X'_{ij} \sim \mathcal{N}(\mu_1, \sigma_1^2)$, $\mathbf{w}'_j \sim \mathcal{N}(\mu_2, \sigma_2^2)$, generally achieve good performance. $\mathbf{w}'$ of fixed values also works well in our experiments. Section 6.5 describes a distribution-insensitive approach, which does not require very accurate distributions

of artificial features and weights. Usually, a pre-estimation over the input data is sufficient to set up the artificial features and weights.

For a classification problem, this procedure is slightly different from that in regression. This is because in classification, the labels need to remain unchanged and it is not possible to create new labels by adding the contribution from "faked" X and $\mathbf{w}$. Therefore, in order to distinguish X' from irrelevant features, we let the artificial features corresponding to each class follow different distributions. In this way, the above approach is modified to

1. Generate $X' \in R^{m \times ns}$, where rows of positive samples $\mathbf{x}'_{(+)} \sim \mathcal{N}(\mu_1, \Sigma_1)$, and rows of negative samples $\mathbf{x}'_{(-)} \sim \mathcal{N}(\mu_2, \Sigma_2)$, diagonal covariance matrices are used in our experiments;

2. Add a feature column from X' to each group;

3. Find solution with new features $[X, X']$ and $\mathbf{y}$;

4. Remove selected artificial features after convergence.

Similar to the regression approach, choosing proper distributions for generating artificial features is a very crucial step of this method but again, some pre-estimation steps can be used.

6.4.2 Stability selection

Stability selection was introduced in Section 3.5. Stability selection provides stable feature selection by repeatedly subsampling from the input data. We adapt the idea of subsampling $\frac{1}{2}$ data points and instead use exclusive group Lasso with a different random group allocation at each iteration of stability selection. We summarize stability selection in conjunction with random group allocation in Algorithm 6.6. As mentioned in Section 3.5, in our experiments, we use single λ value where $\Lambda = \{\lambda\}$.

6.5 Model selection

An obvious disadvantage of the proposed methods is the number of parameters to be determined, including regression parameter λ, selection threshold π, the number

Algorithm 6.6 stabExcl($X, \mathbf{y}, \Lambda, N, \mathcal{G}$): Stability selection

Input: input data X, labels $\mathbf{y}$, regularization parameters $\Lambda = \{\lambda_1, \lambda_2, \cdots\}$, number of iterations N, group $\mathcal{G}$
for $\lambda \in \Lambda$ **do**
  for $i = 1 : N$ **do**
    Subsample from X without replacement to generate subsets $X_{I_i}, \mathbf{y}_{I_i}$, where $I_i \subseteq \{1, \cdots, m\}, |I_i| = \frac{m}{2}$
    Run exclusive group Lasso with $X_{I_i}, \mathbf{y}_{I_i}$ to select features $\hat{S}_i^\lambda = \{j \mid \hat{\mathbf{w}}_j \neq 0\}$
    Shuffle $\mathcal{G}$ if using random group allocation
  end for
  $\Pi_j^\lambda \leftarrow \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}\{j \in \hat{S}_i^\lambda\}$
end for
$\hat{S}^{stable} \leftarrow \{j : \max_\lambda \hat{\Pi}_j^\lambda \geq \pi\}$

of groups $|\mathcal{G}|$, as well as distributions of artificial features X' and artificial weights $\mathbf{w}'$ if using artificial features. In practice, the number of groups is often predefined on subjective grounds. Selection probability threshold can be determined as a fixed threshold by cross-validation, or by clustering such as k-means where $k = 2$. However, the number of groups directly affects the number of selected features: larger $|\mathcal{G}|$ often results in more features being selected because of the inter-group level ℓ_2 norm. In the most extreme case when $|\mathcal{G}| = n$, each group contains one feature and the exclusive group Lasso is equivalent to ridge regression where all features are selected. We therefore propose a model selection pipeline to determine parameters without an exhaustive cross-validation procedure. The model selection pipeline consists of four steps as follows.

1. Apply Lasso to get an estimation of the structure of the input features. Regularization parameter λ is determined by cross-validation. Let S be the set of nonzero features selected by Lasso. Apply OLS only on the features selected by Lasso for an estimation of weights $\hat{\mathbf{w}}_S$. Set the number of groups $|\mathcal{G}| = |S|$. Generate artificial features $X'_{ij} \sim \mathcal{N}(0, 1)$ and artificial weights $\mathbf{w}'_j \sim \mathcal{N}(\bar{\mathbf{w}}_S, \sigma^2)$, where $\bar{\mathbf{w}}_S$ is the mean of $\mathbf{w}_S$, σ^2 is a arbitrarily small number. Compute artificial labels $\mathbf{y}' = X'\mathbf{w}' + \tilde{\mathbf{y}}$, where $\tilde{\mathbf{y}}$ are randomly shuffled labels. The use of shuffled labels implies no features other than the artificial features are real. Set $\gamma = \frac{|\mathcal{G}|}{n}$.

2. Run exclusive group Lasso over an extensive range of λ using X' and $\mathbf{y}'$. Select λ^* when $\gamma n = |\mathcal{G}|$ non-artificial features are selected. The purpose of this step is to select a λ^* that introduces some error in exclusive group Lasso feature selection. Such λ^* is considered to be "safe", since it is small enough to include all relevant features as well as some, but not too many, irrelevant features.

3. Run stability selection for only a few iterations using exclusive group Lasso with $|\mathcal{G}|$ and λ^* from Steps 1 and 2, but now without shuffled labels. Keep a feature if its selection probability is higher than π^*, where π^* is a small number.

4. Set $|\mathcal{G}|$ where each group contains n_g features. Run stability selection with λ^* using the features that are selected from Step 3. Since most irrelevant features have been filtered out in Step 3, the distribution over selection probabilities is more distinct between relevant and irrelevant features. Apparently, the algorithm is more efficient than the full algorithm which runs stability selection on the complete feature space.

The model selection only exploits two undetermined parameters: π^* in Step 3 and n_g in Step 4. In practice, these two parameters are set to some small values: $\pi^* \in \{0.2, 0.3\}$ and $n_g \in \{2, 3\}$. We will discuss in Section 6.6 the feature selection performance, with and without this model selection strategy.

6.6 Experimental results

In this section, we validate the performance of the exclusive group Lasso using both synthetic and real-world datasets. Various correlation settings are used in synthetic experiments to test the feature selection performance of the exclusive group Lasso, Lasso, and elastic net in Section 6.6.1, while in Section 6.6.2, we show the experimental results to identify CMV-associated TCRβs in public T-cell responses.

6.6.1 Synthetic datasets

6.6.1.1 Data generation

The synthetic data is generated as follows. $\mathbf{x}_{(i)} \sim \mathcal{N}(0, \Sigma), \epsilon_i \sim \mathcal{N}(0, \sigma^2), i = 1, \cdots, m$, with Σ being the covariance matrix which will be described later. Weight vector $\mathbf{w} = [s_1 w_S, s_2 w_S, \cdots, s_{n_S} w_S, 0, \cdots, 0]^T$ in which the first n_S entries are some scalar w_S with random signs such that $s_i \in \{-1, +1\}$, and the other entries being 0. Therefore, the first n_S features are relevant and the last $n - n_S$ features are irrelevant. Noise follows a Gaussian distribution such that $\epsilon_i \sim \mathcal{N}(0, \sigma^2)$ for some variance σ^2. Under linear assumption, we compute $\mathbf{y} = X\mathbf{w} + \epsilon$.

To validate the performance on different covariances, four types of Σ are considered.

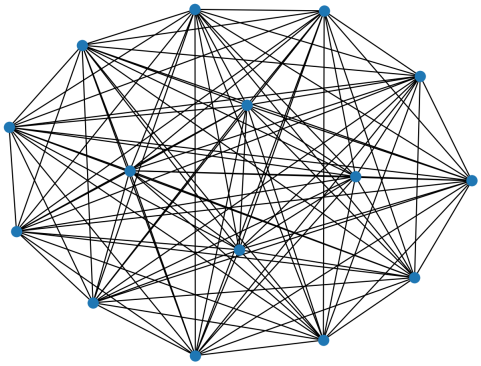
1. Example 1 (pairwise correlation): relevant features are pairwise correlated at some correlation level ρ, while irrelevant features are not correlated either with relevant features, or other irrelevant features;

2. Example 2 (random pairwise correlation): relevant features are pairwise correlated at ρ, while some irrelevant features are correlated with randomly selected relevant features at a lower correlation level $\frac{\rho}{2}$;

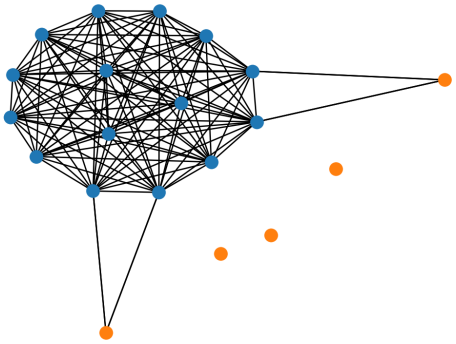
3. Example 3 (blocked diagonal correlation): relevant features are partitioned into equal-sized groups. Features in the same group are pairwise correlated at ρ, while features from different groups are not correlated. If we put features from the same group adjacent to each other, the covariance matrix Σ is an all-zero matrix, except for the blocks along the diagonal;

4. Example 4 (Erdos-Renyi correlation): relevant features are correlated according to the connectivity of an Erdos-Renyi random graph, where features that are connected in the graph are correlated at ρ. Similar to Examples 1 and 3, irrelevant features are neither correlated with relevant features, or other irrelevant features.

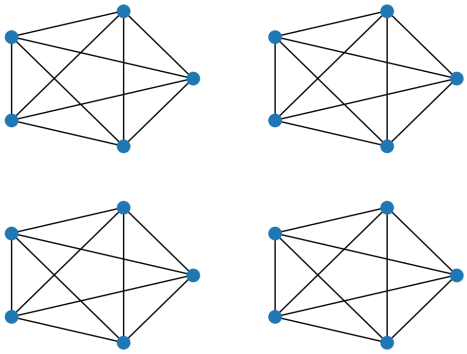
Figure 6.2 illustrates the four types of covariance Σ.



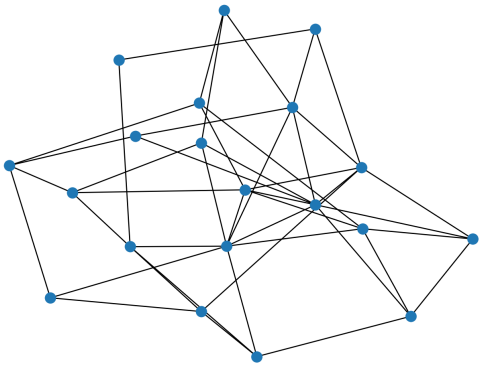
(a) Example 1: pairwise correlation



(b) Example 2: random pairwise correlation



(c) Example 3: blocked diagonal correlation



(d) Example 4: Erdos-Renyi correlation

Figure 6.2: Graph representation of covariance matrix with correlated features being represented by connected nodes in the graph. Blue: relevant features, orange: irrelevant features. Irrelevant features are not shown in the graph if all of them are isolated. (a). Pairwise correlation: 15 relevant features are pairwise correlated; (b). random pairwise correlation: 15 relevant features are pairwise correlated, 2 of 5 irrelevant features are correlated with some random relevant features; (c). blocked diagonal correlation: 4 groups of 5 relevant features are pairwise correlated; (d). Erdos-Renyi correlation: relevant features are correlated based on the connectivity of an Erdos-Renyi graph with 20 nodes and $\frac{4}{19}$ connection probability.

6.6.1.2 Results

Correlation level

We first test the performance of vanilla exclusive group Lasso and Lasso under different correlation levels to validate the advantage of the exclusive group Lasso on highly correlated features. We test the performance on features generated from Example 3. We create $m = 1000$ samples with $n = 200$ features. The number of blocks is set to 2. Figure 6.3 shows the FDR when all relevant features are

discovered by the model. When correlation level ρ is small, the exclusive group Lasso and Lasso perform similarly, while as ρ increases, Lasso only discovers the correct features at the cost of including other irrelevant features.

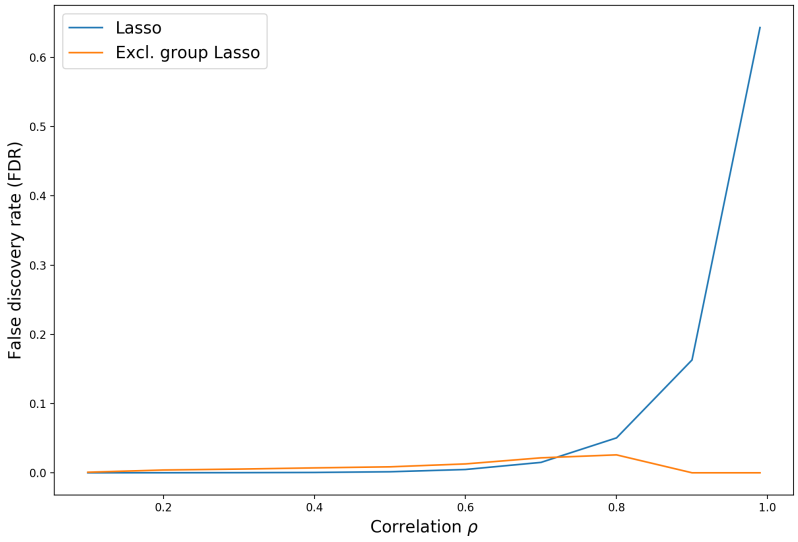


Figure 6.3: False discovery rate of Lasso and exclusive group Lasso under different correlation levels.

Stability selection

We then explore the use of stability selection all types of correlation. We set $m = 100, n = 100, n_S = 30$. For Examples 1, 2, and 3, ρ is set to 0.6. We use this value for ρ because $\rho = 0.6$ is the maximum correlation for Example 2 under such settings. We allow an additional 20 irrelevant features to correlated with relevant features: each of these irrelevant features correlate with two randomly selected relevant features at correlation 0.3. For Example 3, we use 5 blocks, each of which consists of 6 relevant features. For Example 4, $\rho = 0.3$ because 0.3 is the maximum value to ensure the covariance matrix is positive definite. The connection probability is set to $\frac{5}{29}$ to mimic the structure of Example 3. We fix the variance of the noise $\sigma^2 = 1$ under all circumstances. We test the performance of Lasso, elastic net, and the exclusive group Lasso on 50 randomly generated datasets under different $w_S = \{0.1, 0.3, 0.5\}$, with or without stability selection.

We allow 50 iterations in stability selection. For the exclusive group Lasso, we

test two types of group allocation: fixed and random groups. Fixed groups allocate one relevant feature into each group, and the number of groups is thus $|\mathcal{G}| = n_S$. Random groups allocate features randomly. In this experiment, we use $|\mathcal{G}| = 50$ at all times. To validate the feature selection results, we adopt F measure which is commonly used in information retrieval. F measure is computed by

$$2 * \frac{precision * recall}{precision + recall} \tag{6.11}$$

with $precision = \frac{n_S^*}{n^*}, recall = \frac{n_S^*}{n_S}$, n_S being the number of relevant features, n_S^* being the number of relevant features that have been selected, and n^* being the number of all features that have been selected.

Table 6.1 shows the average F measure for the feature selection results from the above experiment. In almost all experiments, using stability selection outperforms using the selection algorithm alone. Not surprisingly, exclusive group Lasso with fixed group allocation outperforms all other methods in most experiments, regardless of correlation types. Exclusive group Lasso with random group allocation still achieves similar or better performance than elastic net and Lasso. Although random group allocation influences the performance of the exclusive group Lasso, results show that stability selection can help random group allocation achieve similar or even better feature selection results than fixed group allocation without stability selection. This verifies the effectiveness of exclusive group Lasso with stability selection and may indicate that in real-world problems, where the information on true features is not available, this pipeline performs as well as exclusive group Lasso with fixed group allocation. As w_S increases, the learning problem tends to become easier as the signal is larger to detect. Figure 6.4 compares the effect of w_S on Example 1 and 2 with stability selection, where selection probabilities of relevant and irrelevant features are determined using k-means with 2 clusters. See Table E.1 for the results using a fixed threshold without k-means.

In the above experiments, we test the performance when $m = n$ under different correlation types. We further examine the results when $m < n$ where

Table 6.1: Feature selection results with or without stability selection.

(a) Without stability selection

	w_S	Lasso	Elastic net	Excl. (fixed)	Excl. (random)
Example 1	0.1	0.4362	0.5939	0.8817	0.8793
	0.3	0.5362	0.8189	0.9726	0.9446
	0.5	0.5995	0.8483	0.9595	0.9892
Example 2	0.1	0.4086	0.5806	0.8560	0.6977
	0.3	0.5140	0.8423	0.9814	0.9347
	0.5	0.5615	0.8620	0.9672	0.9108
Example 3	0.1	0.4446	0.4964	0.5816	0.4760
	0.3	0.5398	0.6461	0.7317	0.5486
	0.5	0.6200	0.6983	0.7629	0.6483
Example 4	0.1	0.4663	0.4462	0.4413	0.4556
	0.3	0.5193	0.5489	0.5813	0.5272
	0.5	0.5566	0.5863	0.6137	0.5430

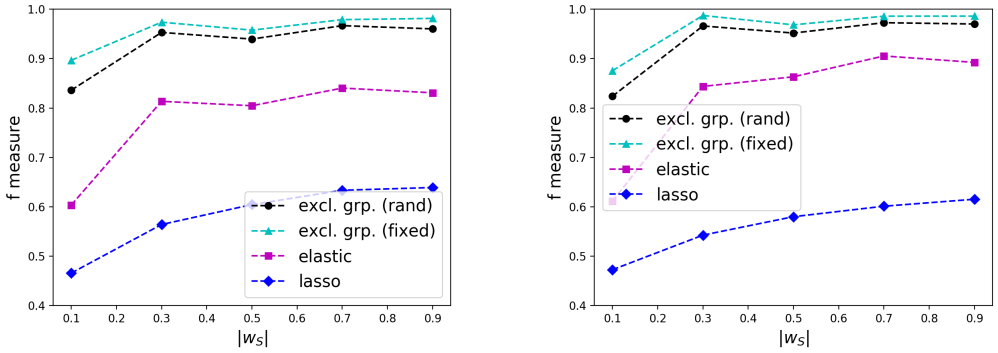
(b) With stability selection

	w_S	Lasso	Elastic net	Excl. (fixed)	Excl. (random)
Example 1	0.1	0.4659	0.6063	0.8966	0.8354
	0.3	0.5640	0.8139	0.9735	0.9530
	0.5	0.6055	0.8046	0.9575	0.9398
Example 2	0.1	0.4727	0.6103	0.8754	0.8238
	0.3	0.5427	0.8437	0.9869	0.9661
	0.5	0.5800	0.8632	0.9684	0.9517
Example 3	0.1	0.4695	0.4891	0.5836	0.5418
	0.3	0.5586	0.6456	0.7660	0.6830
	0.5	0.5587	0.6452	0.7657	0.6809
Example 4	0.1	0.4764	0.4748	0.5130	0.4915
	0.3	0.6649	0.6474	0.6849	0.6157
	0.5	0.7656	0.7247	0.7407	0.6410

$m = 300, n = 1500, n_S = 30$, and $w_S = 0.5$. Table 6.2 shows the average F measure on 30 randomly generated datasets using stability selection with k-means.

Artificial features

In the above experiments, all methods perform better in Examples 1 and 2, while in the other two settings, the performances are not very distinguishable between exclusive group Lasso and other methods, even Lasso. This is probably because the



(a) Example 1 **(b)** Example 2

Figure 6.4: F measure under different values of w_S with stability selection. As w_S increases, all algorithms tend to perform better (higher F measure) because of higher signal values. Exclusive group Lasso (fixed and random groups) outperform both elastic net and Lasso under all w_S values in both examples.

Table 6.2: Feature selection results with stability selection ($m = 300, n = 1500, n_S = 30$).

	Lasso	Elastic net	Excl. (fixed)	Excl. (random)
Example 1	0.2342	0.8804	0.9908	0.9840
Example 2	0.2380	0.8513	0.9587	0.9486
Example 3	0.2795	0.6275	0.6856	0.6376
Example 4	0.4196	0.6813	0.7187	0.5902

large noise level, as well as the small size of both samples and features results in a more noisy input data space than that of the other two examples. To better detect the difference between Lasso and exclusive group Lasso under these two types of correlation, we consider a slightly more challenging task. We set $m = 1000, n = 350, n_S = 50$, which is indeed more constrained for Lasso to select correct features than the above experiments. In addition, we set $\sigma^2 = 0.1$ for a higher signal-to-noise ratio. However, we increase the correlation level ρ to 0.99 for Example 3, which is very challenging to Lasso as discussed at the beginning of this section. ρ is set to 0.25 for Example 4 to ensure that the covariance matrix is positive definite. Furthermore, we adjust the signs of the relevant features. In previous experiments, sign($\mathbf{w}_S$) is determined randomly, and as a result, the numbers of positive weights and negative weights are not equal. In this experiment, however, we restrict the number of positive and negative weights to be equal. Therefore, 25 relevant features

have positive weights while the other 25 relevant features have negative weights. Such modification on weights represents a very challenging task for Lasso, where the irrepresentable condition is easily violated. We set $w_S = 0.2$ in both examples. For Example 3, we set the number of blocks to 10 so that each block contains 5 correlated features. Similarly, the connection probability in Example 4 is set to $\frac{4}{49}$ to simulate the structure of the blocked diagonal type correlation. We use $|\mathcal{G}| = 70$ groups in random group allocation. For better comparison, we add artificial features in this experiment, where the artificial features and weights are generated such that $X'_{ij} \sim \mathcal{N}(0, 1)$, $\mathbf{w}'_j \sim \mathcal{N}(0, 0.05)$ (Example 3) or $\mathbf{w}'_j \sim \mathcal{N}(0, 0.1)$ (Example 4). All experiments are performed with stability selection of 50 perturbations on 100 randomly generated datasets.

Table 6.3 shows the mean selection probabilities of both relevant and irrelevant features over 100 random datasets for Examples 3 and 4. In Example 3, relevant features are less likely to be selected using Lasso, compared with exclusive group Lasso. Lasso performs well in Example 4, possibly because of the low correlation level. Compared to random group allocation, the introduction of artificial features greatly reduces the selection probabilities of irrelevant features, without losing the advantage on relevant features.

Table 6.3: Mean selection probabilities of relevant and irrelevant features over 100 randomly generated datasets using Lasso, exclusive group Lasso with fixed groups (Excl. fixed), random groups (Excl. random), and random groups with artificial features (Excl. randArt).

	Example 3		Example 4	
	Relevant	Irrelevant	Relevant	Irrelevant
Lasso	0.6141	0.0080	0.9997	0.1255
Excl. (fixed)	0.9987	0.0027	1.0000	0.0351
Excl. (random)	0.8113	0.1152	0.9996	0.5272
Excl. (randArt)	0.8226	0.0966	0.9997	0.4615

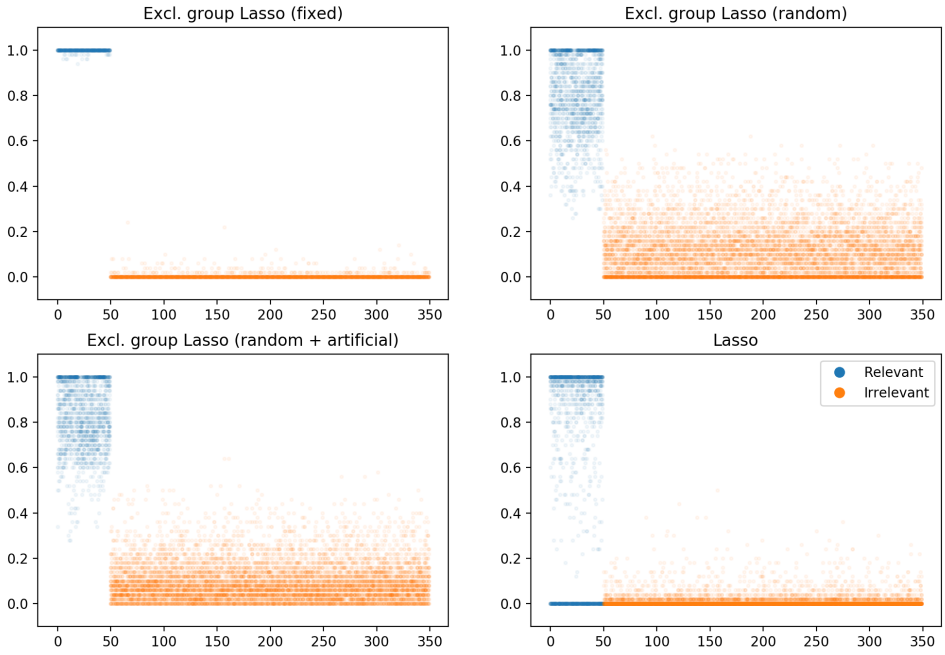
Figure 6.5 illustrates the selection probabilities from the first 30 randomly generated datasets using Lasso and exclusive group Lasso with both fixed and random group allocations, with or without artificial features. The figures are plotted with low opacity for clearer illustration. In both examples, using artificial features re-

duces the selection probabilities of irrelevant features, which agrees with the results shown in Table 6.3. It is clear that Lasso cannot distinguish between relevant and irrelevant features in Example 3 as some relevant features are barely selected. Figures F.1 and F.2 compare the selection probability using one random dataset, which give more clear intuition about how different methods perform.

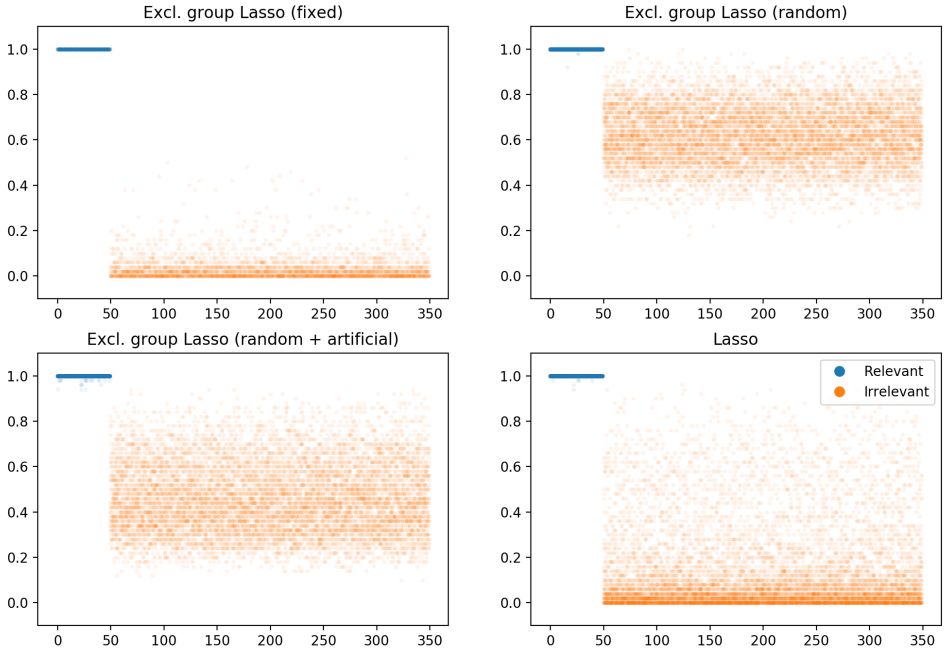
One may argue that under such circumstances artificial features are not needed given that exclusive group Lasso performs quite well with random groups alone. It is true that in the above experiments, relevant and irrelevant features are already distinguishable with random group allocation. However, there are several advantages to include artificial features. First, the use of artificial features makes the selection probability more distinguishable by selecting less irrelevant features without affecting the selection on relevant features, which may help the learning problem in some real-world applications when data is much more noisy than the synthetic data. Second, when jointly used with stability selection, the use of artificial features reduces the number of iterations required for stability selection. Figure 6.6 shows the average selection probability after 5 and 20 iterations in stability selection in Example 3 in the above experiment.

Model selection

We then verify the effectiveness of the proposed model selection approach. We randomly generate 30 datasets with 1,000 samples and 1,000 features using all four correlation types. For Examples 1 and 3, $\rho = 0.99$. For the other correlation types, ρ is set to the maximum value to ensure a valid covariance matrix. $w_S = 0.1$ for all relevant features. Ideally, the number of groups $|\mathcal{G}|$ should be greater than the number of relevant features n_S so that relevant features do not coexist in the same group. Therefore, to better validate the performance of the proposed feature selection pipeline, we test different levels of sparsity where $n_S \in \{20, 50, 100, 200, 500\}$. The number of irrelevant features that are correlated to relevant features in Example 2 equals to $\frac{n_S}{2}$. The number of blocks in Example 3 is set to $\frac{n_S}{10}$, except when $n_S = 20$ and 50, 5 blocks are used. Connection probabilities of Erdos-Renyi graphs in Example 4 are set to mimic the connections in Example 3. For all experiments,

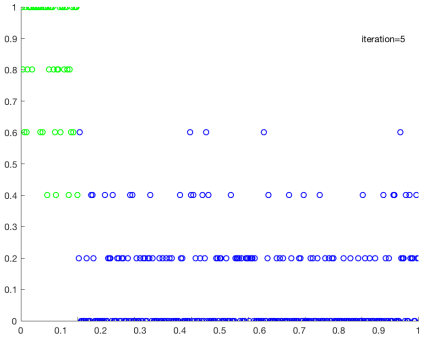


(a) Example 3

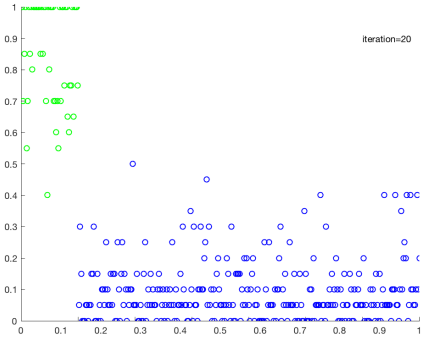


(b) Example 4

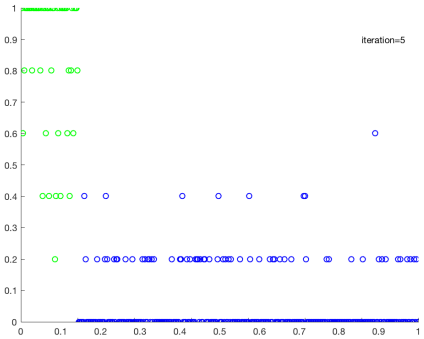
Figure 6.5: Selection probabilities of the first 30 random datasets using Lasso and exclusive group Lasso with fixed and random group allocation, with and without artificial features.



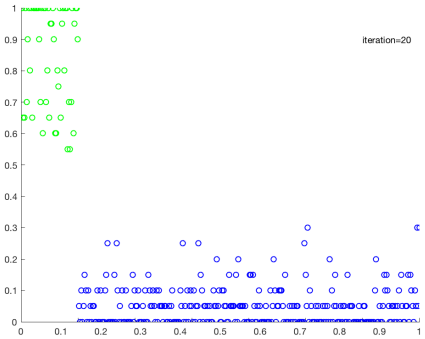
(a) 5 iterations, without artificial features.



(b) 20 iterations, without artificial features.



(c) 5 iterations, with artificial features.



(d) 20 iterations, with artificial features.

Figure 6.6: Changes of selection probabilities during stability selection. Green: relevant features, blue: irrelevant features. (a)-(b): without artificial features; (c)-(d): with artificial features.

we set $\pi^* = 0.3$ and $n_g = 2$.

Table 6.4 compares the mean selection probabilities of relevant and irrelevant features using Lasso, exclusive group Lasso without the model selection pipeline (Excl. random), and exclusive group Lasso with model selection pipeline (Excl. ms). In most experiments, relevant features generally have similar or higher selection probabilities and irrelevant features have lower selection probabilities using the proposed model selection pipeline.

We also compare the F measure in Table 6.5. Relevant and irrelevant features are determined using k-means on the selection probabilities. Since k-means may not perform well with imbalanced data, while features selected in Step 3 are likely to be significantly dominated by relevant features, we add some zeros to the results. The number of zeros is set to be half of the selected features in Step 3. Exclusive

group Lasso achieves similar accuracy with or without the proposed model selection pipeline in Examples 1–3, which proves the effectiveness of the proposed method. However, it is notable that the proposed model selection pipeline does not perform well in Example 4. This is perhaps because of the high noise and low correlation levels. Indeed, we notice that higher correlation has a "clustering effect" for correlated feature selection, while the true mechanism behind this is still unexplained. Fixed threshold without k-means (Table 6.6) is a potential solution to this particular experiment, which requires an additional parameter to determine.

Table 6.4: Mean selection probabilities of relevant and irrelevant features using Lasso, exclusive group Lasso without model selection pipeline (Excl. random), and with model selection pipeline (Excl. ms).

	n_s	ρ	Lasso		Excl. (random)		Excl. (ms)	
			Relevant	Irrelevant	Relevant	Irrelevant	Relevant	Irrelevant
Example 1	20	0.99	0.3573	0.0042	1.0000	0.0009	0.9997	0.0000
	50	0.99	0.3138	0.0036	0.9965	0.0000	1.0000	0.0000
	100	0.99	0.0285	0.0000	1.0000	0.0771	1.0000	0.0001
	200	0.99	0.0151	0.0000	0.9994	0.1819	0.9960	0.0000
	500	0.99	0.0350	0.0000	0.9976	0.4782	0.9806	0.0000
Example 2	20	0.60	0.8787	0.0043	0.9968	0.0026	0.9869	0.0026
	50	0.60	0.8518	0.0004	0.9986	0.0004	0.9583	0.0015
	100	0.50	0.5537	0.0448	0.9923	0.0002	0.8738	0.0037
	200	0.50	0.4498	0.0604	0.9509	0.0620	0.8173	0.0031
	500	0.40	0.3227	0.0060	0.9174	0.4011	0.7670	0.0097
Example 3	20	0.99	0.3920	0.0058	0.8369	0.0014	0.8564	0.0019
	50	0.99	0.3537	0.0054	0.7287	0.0050	0.7998	0.0019
	100	0.99	0.0540	0.0000	0.6895	0.0575	0.7507	0.0009
	200	0.99	0.0354	0.0000	0.8810	0.1753	0.6925	0.0002
	500	0.99	0.0267	0.0000	0.9125	0.4544	0.5889	0.0007
Example 4	20	0.40	0.5508	0.0071	0.4229	0.0010	0.7412	0.0138
	50	0.30	0.4111	0.0068	0.4671	0.0104	0.6389	0.0236
	100	0.15	0.1121	0.0288	0.3292	0.0544	0.7284	0.0624
	200	0.15	0.0098	0.0000	0.0046	0.0000	0.6970	0.1710
	500	0.15	0.0895	0.0225	0.7132	0.5400	0.6831	0.3513

Computational time

We further examine the computational time using the exclusive group Lasso and the elastic net. We do not include Lasso since previous experiments show Lasso does not perform well on correlated features. We generate either 1,000 or 10,000 samples with various numbers of features (n) and relevant features (n_S) as in Example 1. We set $|\mathbf{w}_S| = 0.3$ with random signs and $\rho = 0.9$. Exclusive group Lasso is run with fixed group allocation on 10 λs and the elastic net on a parameter grid of 10 αs and 5 ℓ_1 ratios, using Scikit-learn [211] in Python 2.7. The computational

Table 6.5: Average F measure using Lasso, exclusive group Lasso without model selection pipeline (Excl. random), and with model selection pipeline (Excl. ms). Selection probabilities are distinguished using k-means.

	n_s	ρ	Lasso	Excl. (random)	Excl. (ms)
Example 1	20	0.99	0.7312	1.0000	1.0000
	50	0.99	0.7749	1.0000	1.0000
	100	0.99	0.8120	1.0000	1.0000
	200	0.99	0.8100	0.9993	0.9997
	500	0.99	0.6694	0.9985	0.9936
Example 2	20	0.60	0.9773	1.0000	0.9984
	50	0.60	0.9861	1.0000	1.0000
	100	0.50	0.9916	1.0000	0.9995
	200	0.50	0.9894	0.9991	0.9987
	500	0.40	0.9714	0.9937	0.9937
Example 3	20	0.99	0.7043	1.0000	0.9569
	50	0.99	0.7033	1.0000	0.9921
	100	0.99	0.6565	1.0000	0.9721
	200	0.99	0.6407	0.9998	0.9529
	500	0.99	0.6667	0.9928	0.8975
Example 4	20	0.40	0.7787	0.9004	0.7372
	50	0.30	0.7835	0.8827	0.7445
	100	0.15	0.8490	0.8907	0.7604
	200	0.15	0.8404	0.8363	0.6839
	500	0.15	0.7524	0.7962	0.7375

Table 6.6: Average F measure of Example 4 using Lasso, exclusive group Lasso without model selection pipeline (Excl. random), and with model selection pipeline (Excl. ms). Selection probabilities are distinguished using fixed threshold without k-means.

n_s	ρ	Lasso	Excl. (random)	Excl. (ms)
20	0.40	0.7754	0.8592	0.8216
50	0.30	0.8535	0.9042	0.8398
100	0.15	0.8884	0.8929	0.8802
200	0.15	0.8400	0.8263	0.8082
500	0.15	0.8176	0.7883	0.7979

time (CPU time in seconds) over the grid of parameters is reported in Table 6.7. When n_S is reasonably small, the exclusive group Lasso outperforms elastic net. As n_S increases, the exclusive group Lasso loses some of its advantages as larger n_S requires more groups.

Table 6.7: Computational time (s) of exclusive group Lasso and elastic net

m	n	n_S	Excl. grp. Lasso	Elastic net
1000	5000	100	44.64	130.12
1000	5000	500	224.99	223.51
1000	5000	1000	511.57	270.93
1000	10000	100	53.08	129.31
1000	10000	500	244.06	415.83
1000	10000	1000	579.52	486.85
1000	20000	100	82.13	575.63
1000	50000	100	980.87	1426.86
10000	20000	100	1429.97	5536.68
10000	50000	100	3947.49	13695.28

6.6.2 Real-world datasets

So far, we have validated the performance of exclusive group Lasso from various angles using synthetic datasets under different correlation settings. The reason we turn to exclusive group Lasso in the first place was to extract entire antigen-specific TCR sequences from RNA-seq data. Public T-cell responses, in which the same TCR sequences in multiple individuals are observed in response to the same antigen epitope, have been actively studied in immune T-cell repertoire problems. Previous studies have shown that public TCRs are highly likely to exist in the naive T-cell repertoire at any time [48], and therefore, such TCRs are more likely to be detected in the repertoires of subjects that are exposed to the cognate antigen. There are many studies on public T-cell responses and public TCRs that respond to various infectious diseases such as influenza, Epstein-Barr virus, HIV, and CMV. CMV infects up to 70% human adults [212]. Studies showed that at least 60% of US population has been exposed to CMV [213]. The high infection rate of CMV therefore provides high statistical power in this study.

Table 6.8: Description of the CMV dataset

		Cohort 1	Cohort 2
CMV staus	Positive	289	51
	Negative	352	69
	Unknown	25	0
HLA status	Known	626	0
	Unknown	40	120
Total		666	120

Our methods presented in Chapters 4 and 5 although perform well when extracting CDR3β subsequences in our experiments, do not give satisfactory results in this problem because of the correlation between TCRs that respond to the same antigen. A potential remedy is exclusive group Lasso, which shows good performance in synthetic datasets under different correlation settings. To identify public TCRs in response to CMV, we apply the exclusive group Lasso to a published dataset [214] which is described below.

6.6.2.1 Data description

The dataset consists of TCR β sequences (TCRβs) sequenced from CD8+ naive and memory T cells from two independent cohorts. The rearranged CDR3 region of TCRβs from 666 healthy bone marrow donors were immunosequenced for Cohort 1, with 641 samples being either CMV seropositive (CMV+) or CMV seronegative (CMV-). An independent Cohort 2 with 120 samples were also sequenced, all with known CMV status. A unique TCRβ is defined as a unique combination of a CDR3 sequence, a V gene, and a J gene. Cohort 1 consists of a total of 89,840,865 unique TCRβs, while Cohort 2 contains 20,829,966 unique TCRβs. The majority of TCRβs were private or found in a few samples (Figure 6.7). Training and testing were performed on Cohort 1 and 2, respectively. Details regarding both cohorts are summarized in Table 6.8.

In [214], the authors applied a one-tailed Fisher exact test [215] to identify CMV-associated TCRβs. As a result, 164 unique TCRβs were identified, and the CMV-associated TCRβs were applied to a probabilistic model for classification.

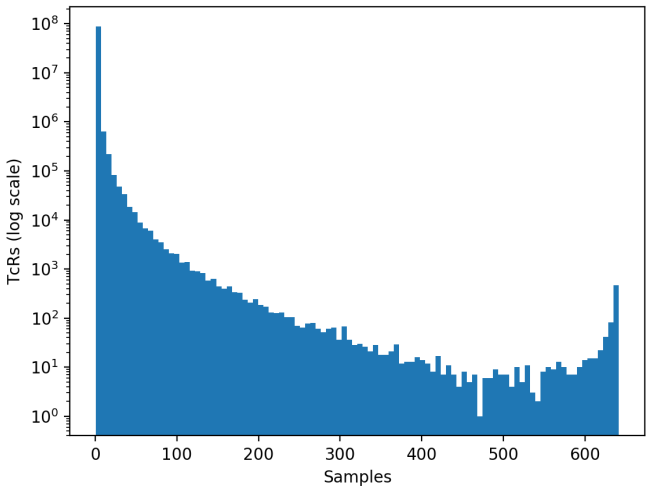


Figure 6.7: Histogram of TCRβs in Cohort 1 (training set) in logarithm scale. The majority of TCRβs only existed in a few samples.

The correlation between the 164 TCRβs is shown in Figure 6.8. A prediction accuracy of 0.89 on Cohort 2 was reported in that paper.

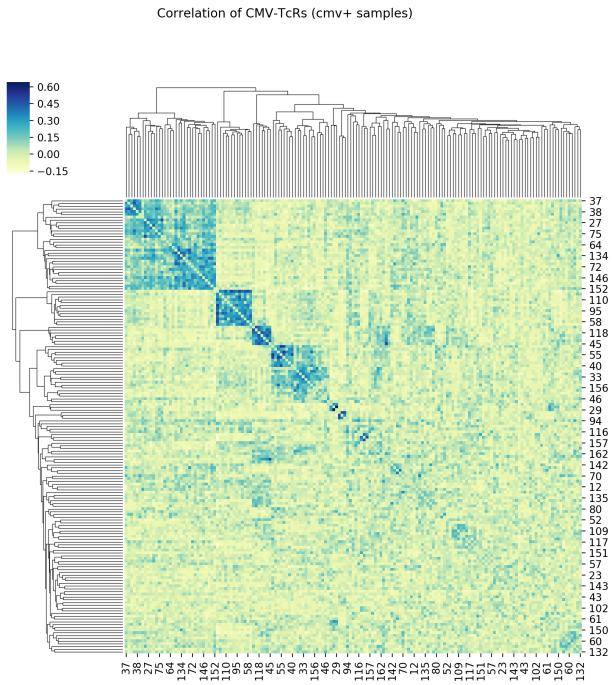


Figure 6.8: Clustermap of the correlation between 164 CMV-associated TCRβs from CMV+ samples. TCRβs were identified in [214]. Diagonal of the correlation matrix was set to 0.

In addition to this dataset, we test the proposed methods on an independent dataset, with sequences collected from subjects in very different geographical locations [216]. This dataset (Cohort 3) consists of 33 samples dominated by CMV- subjects (24), compared with CMV+ subjects (9). Whilst both CD4+ and CD8+ naive and memory T cells were collected for Cohort 1 and Cohort 2, Cohort 3 only included CD4+ memory T cells. A total of 67 CMV-associated TCRβs identified by [214] were also discovered in Cohort 3. With the reported 164 TCRβs and the probabilistic model in [214], an accuracy of 0.8 was achieved for Cohort 3.

6.6.2.2 Results

We apply the exclusive group Lasso to both datasets to identify TCRβs that are associated with the public T-cell responses to CMV. We inherit the definition in [214] in which two TCRβs are considered to be identical if they have the same CDR3, V gene, and J gene. We generate artificial features for CMV+ and CMV- samples separately such that $\mathbf{x}_{(+)} \sim \mathcal{N}(0.1, 0.5), \mathbf{x}_{(-)} \sim \mathcal{N}(0, 1)$, where $\mathbf{x}_{(+)}$ and $\mathbf{x}_{(-)}$ denote the artificial features of CMV+ and CMV- samples, respectively. The artificial features are added to $|\mathcal{G}| = 200$ randomly allocated groups, each of which contains a single artificial feature. Since CMV-specific TCRβs are likely to expand upon exposure to CMV, we restrict the selected TCRβs to be positively related to CMV+ samples as in Algorithm 6.2. In addition to exclusive group Lasso, we also test two other methods, ℓ_1-regularized SVM and elastic net with hinge loss, on the CMV dataset to compare with exclusive group Lasso. To improve the efficiency and lower the risk of memory problems, we only experiment with TCRβs that occur in at least 10 samples for feature selection, which result in 241,008 unique TCRβs in Cohort 1. ℓ_1-regularized SVM, elastic net, and exclusive group Lasso are only used for TCR selection, while the classification task is performed by SVM with a linear or RBF kernel. Stability selection with 30 iterations is also applied to all feature selection methods.

Similar to [214], the proposed pipeline is applied to Cohort 1 for training and Cohort 2 for testing. Parameters are selected by a 10-fold cross-validation on Cohort 1. To remove the possible effects of geographical location, we apply the trained

model with selected TCRs on a completely independent dataset (Cohort 3) published by [216]. As 35 samples were published for 33 patients in Cohort 3 and this cohort is used for testing only, our predictions include the additional measurements for duplicated patients without affecting the trained model. Accuracy on Cohort 2 and Cohort 3 is reported in Table 6.9.

Table 6.9: Cross-validation accuracy on Cohort 1 and test accuracy on Cohorts 2 and 3. Models are trained on Cohort 1. Classification results are based on a two-step procedure, in which TCRβs are selected by 1) ℓ_1-SVM; 2) elastic net (elastic); 3) exclusive group Lasso (excl. grp.), and SVM with linear or RBF kernel is fitted on the selected TCRβs for classification. Both linear and RBF SVM predict all 35 samples in Cohort 3 as CMV- with features selected by elastic net. Performances are even worse with Lasso. Although linear and RBF SVM result in the same accuracy in Cohort 3 with features selected by Lasso, the actual prediction are not consistent: 31 out of 35 samples share the same prediction between linear and RBF SVM.

		ℓ_1-SVM	Elastic	Excl. grp.
Cohort 1	SVM (linear)	0.8799 ± 0.01	0.7443 ± 0.04	0.9282 ± 0.02
	SVM (RBF)	0.9049 ± 0.03	0.7084 ± 0.04	0.9329 ± 0.01
Cohort 2	SVM (linear)	0.6667	0.6750	0.8750
	SVM (RBF)	0.6500	0.6583	0.8833
Cohort 3	SVM (linear)	0.6286	0.7429	0.8571
	SVM (RBF)	0.6286	0.7429	0.8857

Although the highest accuracy with SVM is 0.88, which is slightly lower than 0.89 as in [214], a higher accuracy of 0.90 can be achieved with the published model in [214]. A significantly higher prediction accuracy is achieved for Cohort 3, compared with a classification accuracy of 0.80 if using the published model with the 164 TCRβs from [214].

A total of 47 TCRβs are identified from Cohort 1 by exclusive group Lasso, and 21 of them also exist in Cohort 3. Tables G.1 and H.1 summarize the TCRβs discovered from both cohorts. As shown in Table 6.9, we achieve similar or even better results in both Cohorts 2 and 3, but with much less TCRβs. Correlation between the selected TCRβs is shown in Figure 6.9.

We compare the CMV-associated TCRβs with previously published work [51–53, 217–247], which were also summarized in [214]. Among all 1054 previously

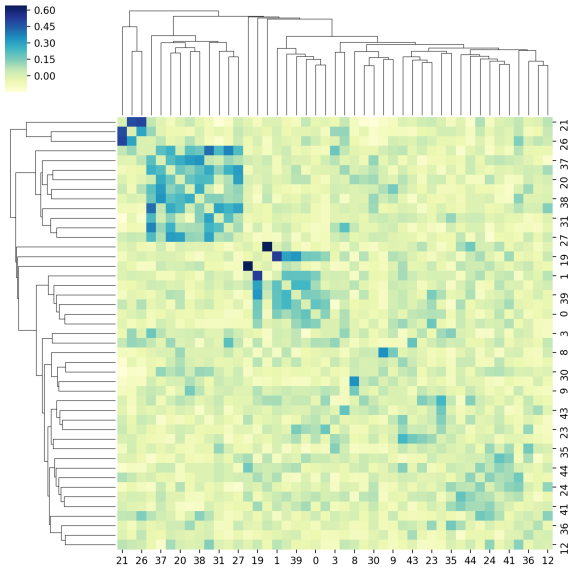


Figure 6.9: Clustermap of the correlation between 47 CMV-associated TCRβs from CMV+ samples. Diagonal of the correlation matrix was set to 0.

published TCRβs in response to CMV, three of them are also identified by the proposed method (shown in Table 6.10). All three TCRβs are observed from subjects in both Cohort 1 and 2, but are absent from Cohort 3 samples, possibly because of the small sample size or potential differences in T-cell repertoires from distinct geographical locations. The small overlapped TCRβs indicate that the identified TCRβs are not more abundant in CMV+ samples in the broader context. This is not a surprising result since the majority of T-cell responses and TCRs are private. Besides, previously published studies did not specifically focus on TCRβs that are more abundant in CMV+ subjects, while both our study and [214] emphasized these TCRβs. From this viewpoint, previously published TCRβs may react to antigenic epitopes from pathogens other than CMV because of cross reactivity, in which TCRs recognize more than one MHC-peptide complex structure.

Table 6.10: Common TCRβs with previously published CMV-TCRβs.

CDR3	V gene	J gene
CASSPQRNTEAFF	TCRBV04-03*01	TCRBJ01-01*01
CASSLAPGATNEKLFF	TCRBV07-06*01	TCRBJ01-04*01
CASSLIGVSSYNEQFF	TCRBV07-09*	TCRBJ02-01*01

6.7 Conclusion

In this chapter, we focus on the identification of entire TCRβ sequences that are associated with public T-cell responses to CMV-infected subjects. We apply the exclusive group Lasso to ascertain CMV-associated TCRβs sequenced from CD4+/CD8+ naive and memory T cells. The proposed methods recognize 47 TCRβs which are abundant in CMV seropositive patients and are less likely to be cross reactivated to other antigens. Our methods and results also show some success in another independent cohort which is geographically remote with possibly different genetic patterns. Success in this cohort also indicates that CMV-seropositive-associated TCRβs present in CD4+ memory T cell repertoire with satisfactory prediction power. Only a few of the selected TCRβs are reported by some previous studies. Potential explanations include the extreme privacy of T-cell repertoires, cross-reactivity in T-cell responses, as well as the extra restrictions concerning TCRβs that are positively related to CMV.

To our knowledge, the exclusive group Lasso has been mostly used for feature selection on uncorrelated features, while no prior studies have applied it to correlated feature selection. In this chapter, we extensively explore the exclusive group Lasso under several different correlation settings and verify its potential to select highly correlated features. As a group-based algorithm, group allocation plays an important role in the exclusive group Lasso. Without appropriate groups, exclusive group Lasso is rather impractical in real-world scenarios. Previous studies have not addressed this point or offer practical guidelines on how to allocate groups in exclusive group Lasso. In our studies, we provide simple but powerful methods to increase the performance of exclusive group Lasso. Experimental results in synthetic data show that with stability selection and artificial features, exclusive group Lasso under random group allocation can achieve satisfactory performances with a small number of iterations. In addition, we present a model selection procedure without parameter tuning. The proposed model selection pipeline works well under most correlation settings, but still needs some improvements in determining the threshold of selection probabilities: k-means may fail to select all relevant features,

while fixed threshold requires an additional parameter to determine. However, even if k-means is not used to separate the selection probabilities, the model selection pipeline still has advantages since many other parameters, such as λ and $|\mathcal{G}|$, do not need to be predefined. Furthermore, novel algorithms to efficiently solve exclusive Lasso are developed in an easy-to-implement fashion, which show a computational advantage over elastic net. The application of exclusive group Lasso to real-world data to detect CMV-associated TCRβs provides potential diagnostic strategies for CMV seropositive patients, as well as shedding light on identifying antigen-specific TCRβs and other patterns such as gene expression of various diseases in broader immunological applications.

Chapter 7

Conclusion

The aim of this thesis has been to identify biological patterns in response to various antigens in adaptive immune responses. To achieve our goal, we develop fast dynamic programming algorithms to solve LPBoost, an ℓ_1-regularized method, on both matrix and graph-represented feature space. We use both string kernel and Fisher kernel to work with sequencing data. The proposed algorithms are improved to select both fixed-length and varying-length substrings from potentially high-dimensional feature space. Aware of the limitations of ℓ_1-regularization on feature selection from correlated feature space, we further apply the exclusive group Lasso, an $\ell_{1,2}$-regularized method, and develop various fast and stable optimization algorithms. In addition, to address the problem of group allocation in exclusive group Lasso, which has largely been ignored by previous studies, we propose novel pipelines to control the error rate of feature selection from randomly allocated groups as well as selecting best models without an exhaustive model selection procedure. Experimental results on synthetic experiments show that the exclusive group Lasso, in conjunction with the proposed algorithms and pipelines, has advantages over Lasso and elastic net under most settings.

Our results on real-world datasets show that the proposed methods can select both short subsequences and entire sequences to classify immunization in response to different antigens and diseases. Experimental results indicate that a small set of short amino acid motifs in CDR3β sequences is sufficient to classify OVA- and non-OVA-immunized mice samples. Furthermore, the selected subsequences are mainly

discovered at specific positions along mouse CDR3β sequences, which may shed some light on learning both physical and chemical interactions between TCR and MHC-peptide complex. Our work also suggests that the status of CMV infection from a geographically isolated population can be predicted by a small subset of public TCRβ sequences, proving that immunological phenotypes such as the infection status of CMV control the TCR repertoire.

Due to the lack of available data, it is questionable whether the selected patterns, particularly subsequences in response to OVA and other antigens in the mouse experiment, generalize well to other mouse experiments. However, both methods on selecting antigen-specific subsequences and entire sequences have the potential to be generalized to predict other antigens and provide machine learning pipelines to broad applications in designing strategies for diagnosis, treatment, and prognosis of various diseases. Moreover, our study addresses a very important task in machine learning: feature selection from ultra-high dimensional space, with an additional emphasis on correlated features. The proposed feature selection algorithms can be applied to various fields to efficiently reduce the dimensionality of the input features to select uncorrelated or correlated features.

There remain a number of experiments and improvements that can be done in our studies and the general field of feature selection from high-dimensional biological datasets.

In regard to the data presented in this thesis, future experiments involve collecting more samples and sequences. More sequences per sample, which can be achieved by sequencing from more blood or tissue samples, can allow us to detect more public and private sequences for a better grasp of the learning problem. A greater number of samples can help to address the concern of generalization of selected sequences on the larger population. However, this is often limited by cost and available funding sources.

Aware of the limitations of the typically small sample size of most biological data, we believe future research directions include data-efficient algorithms to work with small-sample datasets. Transfer learning with deep Bayesian hierarchical

modelling has received success in small-sample problems and we believe this can provide potentially powerful tools for learning and inferring patterns from biological datasets. A major reason for doing so is that although most biological datasets are of rather small sample sizes, a great number of datasets that address the same or similar problems have been published. Therefore, storing knowledge from existing datasets for making predictions on new datasets concerning similar problems may provide more predictive and generalized power to various problems. Such approaches are limited by the negative learning effects which are often observed in transfer learning studies: the use of deep Bayesian hierarchical modelling may help address this concern.

Appendix A

The immune system

A.1 The innate immune system

The innate immune system provides fast and broad responses against many common pathogens with the help of the physical barriers, secretions, and broad cellular responses. The physical barrier, including skin and mucous membranes, provides effective protection over a large area: the average surface area covered by adult human skin is approximately 2 square meters while the area covered by mucous membranes is approximately 400 square meters [248]. Secretions such as tears and saliva can snare and destroy foreign invaders. The general immune response devastates invaders via a series of biochemical reactions that will be further discussed in this section.

A.1.1 Induced cellular innate immunity

The innate immune system has evolved to recognize common invaders through receptors such as pattern recognition receptors (PRRs) to evoke immediate responses. Pattern recognition receptors are found in many cells of the innate immune system, such as macrophages, dendritic cells, neutrophils, and NK cells, as well as members of the adaptive immune system, including B cells and T cells. They are also found in non immune cells such as mucosal epithelial cells and vascular endothelial cells [249]. Pattern recognition receptors can recognize two groups of molecules: pathogen-associated molecular patterns and damage-associated molecular patterns, which are associated with extracellular and intracellular pathogens or

debris from cell damage and death, respectively. One important subset of PRRs is toll-like receptors (TLRs) [250], which identify diverse microbes via different ligands[251, 252]. For example, TLR3 recognizes double-stranded RNA on certain types of viruses, TLR4 recognizes lipopolysaccharide (LPS) on Gram-negative bacteria and mannans on fungi, TLR5 can bind to bacteria via flagellin, a structural protein of bacteria flagella, and TLR7 identifies single-stranded RNA of many RNA viruses. Later in the section, how PRRs and TLRs aid other players in the immune system to defend against infectious pathogens and cellular debris will be presented.

A.1.2 Leukocytes

Leukocytes, or white blood cells, are an important part of the immune system and help the body defend itself against infectious pathogens. They are produced and derived from hematopoietic stem cells in bone marrow (Figure A.1), and are categorized into three cell types: monocytes, which can differentiate into macrophages and certain types of dendritic cells; granulocytes, which include neutrophils, eosinophils, and basophils; and lymphocytes, including B cells and T cells that are involved in adaptive immune response as well as natural killer cells (NK cells). In this section, we focus on two cell types that play important roles in both innate and adaptive immune responses: macrophages and dendritic cells. We close this section with a brief introduction of other types of leukocytes.

A.1.2.1 Macrophages

Phagocytes are a group of white blood cells that engulf and digest foreign invaders, cellular debris, tumor cells and anything else that needs to be cleared from the human body. This engulfing process is called phagocytosis. Phagocyte types in humans and other animals include non-professional and professional phagocytes, the latter of which have pattern recognition receptors on the cell surface for more effective phagocytosis. Professional phagocytes include various types of white blood cells such as macrophages, neutrophils, mast cells, and dendritic cells [254].

Macrophages are very versatile [255–257]. They collect cellular debris, present antigens to T cells, and kill pathogenic invaders. Macrophage phagocytosis

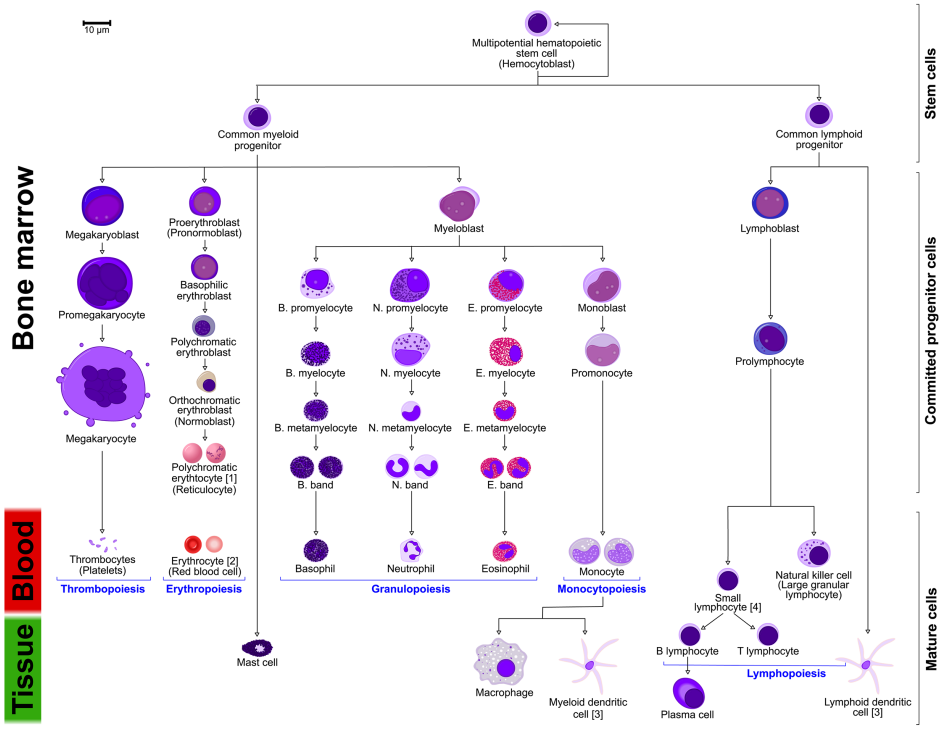
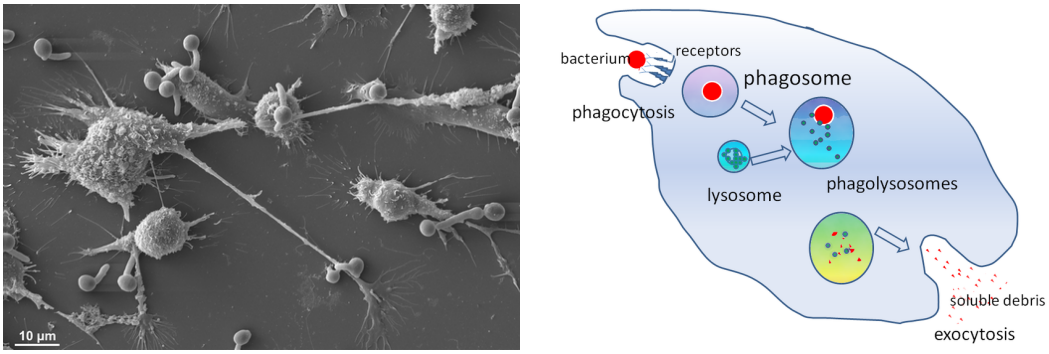


Figure A.1: Blood cells derived from hematopoietic stem cells [253].

is very similar to the mechanism employed by amoebas and some other unicellular organisms. When macrophages encounter its target, it first engulfs the target into its phagosome—a vesicle inside macrophages. The lysosome is then fused with the phagosome, releasing lethal enzymes and chemicals to destroy the invader inside the phagosome. Figure A.2 shows phagocytosis by macrophages.



(a) **(b)**

Figure A.2: Macrophages. (a). Election micrograph of a macrophage interacting with Candida albicans [258]; (b). simplified phagocytosis by a macrophage [259].

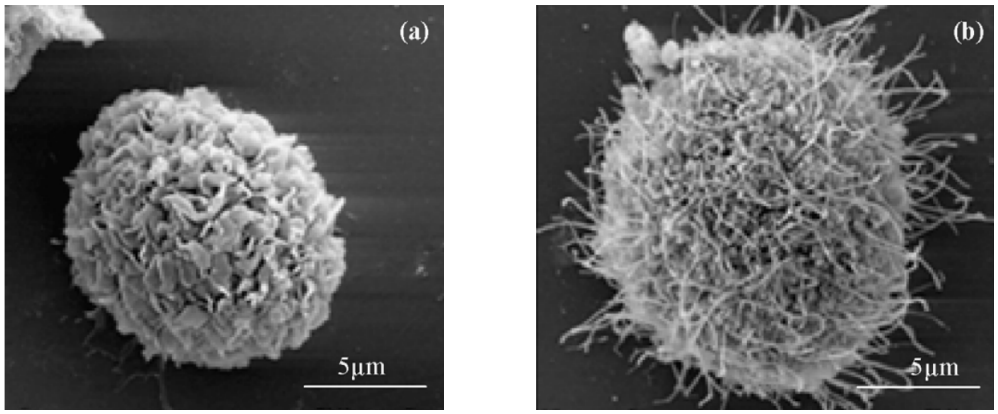
Macrophages need to be activated to present antigens or kill infectious in-

vaders. Unactivated macrophages primarily engulf dead cells to remove debris. A number of cytokines, such as interferon gamma (IFN-γ), can activate macrophages. Once activated, macrophages upregulate the expression of MHC-II molecules and present the pathogen proteins that have been digested via phagocytosis to T cells. This process is known as antigen presentation, which has been discussed in Section 2.1.1. Macrophages can be hyperactivated by molecules such as LPS, a common component found on the membrane of many bacteria [260]. Macrophages can also react to mannose, a carbohydrate present on the cell surface of many common pathogens [261]. In the hyperactive state, macrophages grow larger to provide more effective phagocytosis [248]. Lysosomes and the production of hydrogen peroxide inside hyperactivated macrophages also increase[262]. In addition, macrophages under this state produce various cytokines such as tumor necrosis factor (TNF) and interleukin-2 (IL-2), and are involved in many other immunological processes.

A.1.2.2 Dendritic cells

Dendritic cells (DCs) are an important part of the mammalian immune system. Although they are not involved in antigen-specific reactions, they present antigens to T cells in the adaptive immune system and build a bridge between the innate and the adaptive immune systems.

The most common division of dendritic cells are myeloid and lymphoid DCs, which evolve from myeloid and lymphoid precursors, respectively (see Figure A.1). Both types of dendritic cells are of hematopoietic origin—myeloid DCs developed from monocytes, which can also evolve into macrophages depending on the chemical signal. Lymphoid progenitors become NK cells, B and T lymphocytes, and lymphoid DCs. Dendritic cells are often found in tissues receiving stimuli from the external environment such as skin, stomach, and intestines [263–265]. Immature dendritic cells also exist in the blood and activated dendritic cells move to the nearby lymph nodes to participate in the adaptive immune response. Figure A.3 shows electron microscopy photos of porcine immature and CD40L-stimulated dendritic cells. The main function of dendritic cells is acting as antigen-presentation cells, which has been introduced in detail in Section 2.1.1.



(a) **(b)**

Figure A.3: Porcine dendritic cells from electron microscopy [266]. (a). Immature-DC; (b). DC after 48 hours of stimulation with human CD40L-transfected L-cells.

A.1.2.3 Other leukocytes

Other leukocytes, such as neutrophils, eosinophils, basophils, and natural killer cells (NK cells), also play important roles in the innate immune system and are actively involved in the adaptive immune response. Neutrophils are the most predominant white blood cells. There are approximately 20 billion neutrophils in the human bloodstream, constituting 60-70% of white blood cells [252]. Neutrophils are highly motile—they can exit the blood and become fully activated in 30 minutes [248]. The functions of neutrophils include phagocytosis as professional phagocytes, producing various signaling cytokines including TNF superfamily proteins and chemokine [267], inflict direct harm to microbes with various complexes such as NADPH oxidase [268], and release hydrolytic enzymes to cause liquefactive necrosis around infectious tissues. Unlike macrophages which can live for months, neutrophils only have a very short life span of 1-2 days [269].

Natural killer cells are a type of cytotoxic lymphocyte that are mostly found in blood and blood-rich organs such as spleen and liver. NK cells can detect stressed virus-infected cells that do not present class I MHC molecules and destruct the target cells with various enzymes such as granzyme B, a protease found in NK cells and killer T cells. Therefore, NK cells play a pivotal role because it can recognize and devastate infected cells that cannot be identified by T cells.

Other leukocytes such as eosinophils and basophils are in very small numbers.

Eosinophils make up approximately 1-3% of human leukocytes [270]. They are responsible for the immune responses to parasitic infections, allergies, and certain infections in vertebrates. Basophils are the rarest leukocytes (less than 1% in peripheral blood) [271] and are mainly related to allergic responses.

A.1.3 Complement system

The complement system consists of more than 30 glycoproteins that are present in tissue, fluids, and blood [252, 272]. It plays a central role in the innate immune system and is recruited by the adaptive immune system through antibodies. The complement system can be activated by three different pathways, all of which converge to form C3 convertase which cleaves C3 into C3a and C3b. Based on the particular activation pathway (alternative, classical or lectin), C3b interacts with different protein complexes to generate C5 convertase which cleaves C5 into C5a and C5b to initialize the membrane attack complex (MAC), a complex component consisting of C5b, C6, C7, C8, and multiple C9. It can be inserted into cell membranes of infectious invaders and create pores, resulting in cell lysis.

A.1.3.1 Alternative pathway

In the alternative pathway, C3 is cleaved into two parts: a smaller C3a and a larger C3b. C3b can bind to amino and hydroxyl groups found on the surfaces of many invaders. Any C3b that fails to bind to one of these groups within 60 seconds is neutralized by binding to a water molecule. Activated C3b can further interact with factor B, while factor B is activated and clipped by factor D to form Bb. This results in C3bBb, a C3 convertase that accelerates the process by cleaving more C3 molecules into C3a and C3b directly. C3bBb also interacts with C3b to form a C5 convertase for MAC.

A.1.3.2 Classical pathway

The complement system is recruited by antibodies through the classical pathway. C1 exists as a complex consisting of C1q, C1r, and C1s, and C1q must bind to at least 2 Fc regions on antibodies that are already bound to antigens. C1s then cleaves C4 to C4a and C4b, which binds to C2, and C2 is further cleaved into C2b and C2a,

forming a C3 convertase C4b2a. C3b that is cleaved from C3 by C3 convertases can bind to C4b2a to form C4b2a3b, acting as a C5 convertase for initiating MAC.

A.1.3.3 Lectin pathway

Similar to the classical pathway, the lectin pathway also exploits C2 and C4 to form C3 convertases. Instead of through the C1 complex, the lectin pathway is activated upon binding to mannose, a carbohydrate molecule found in many bacteria, viruses, and fungi, through mannose-binding lectin. Mannose-binding lectin then combines with mannose-binding lectin-associated serine protease, which is structurally similar to C1s in the classical pathway, initiating the complement system by cleavage on C2 and C4. The remainder of the process follows the classical pathway. Figure A.4 compares the three pathways in activating the complement system.

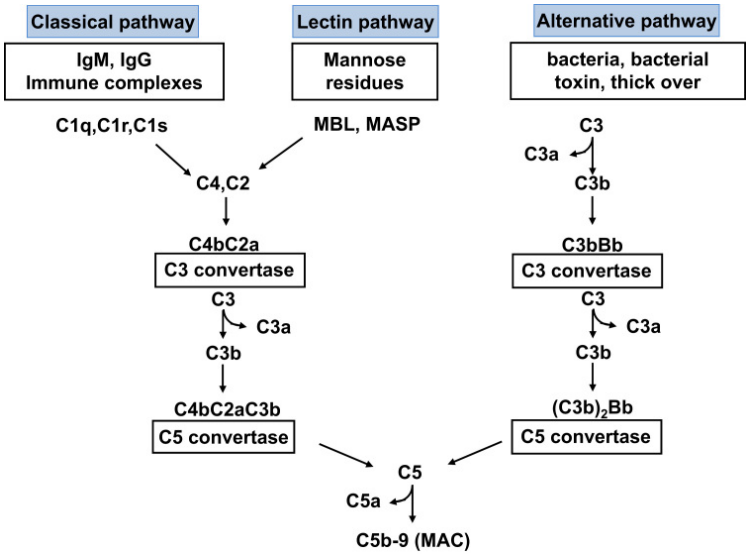


Figure A.4: Complement system activation pathways [273].

A.1.3.4 Complement functions

Complement proteins have diverse functions in both the innate and the adaptive immune systems. They assist professional phagocytes through opsonization via C3b and C4b, as well as generating MAC with C5b, C6, C7, C8, and C9 to cause bacterial cell lysis. Complement proteins such as C3b can bind to B cells to promote the production of antibodies. In addition, C1q and MBL induce T cell activation and C3b and C4b and their fragments bind to antigen-immune complexes to enhance

immunological memory of B cells and T cells.

A.2 The adaptive immune system

The adaptive immune system, or the acquired immune system, is a highly specific subsystem with specialized cells to eliminate infectious pathogens. It creates an immunological memory of a specific pathogen after initial exposure, evoking an enhanced response to the same pathogen in subsequent infections. The adaptive immune system includes humoral immunity, which is mediated by extracellular macromolecules such as antibodies produced by B cells in body fluids, and cell-mediated immunity, which involves various types of T cells. Chapter 2 has introduced cell-mediated immunity, which is most relevant to this thesis. In this section, we mainly focus on humoral immunity of the adaptive immune system.

A.2.1 Humoral immunity

A.2.1.1 B cells

B cells, or B lymphocytes, are a type of leukocyte (see Figure A.1). The main function of B cells in the adaptive immune system is to secrete antibodies, but they also assist other immune system components by presenting antigens to T cells and producing cytokines. B cells can bind specific antigens via B-cell receptors (BCRs) on the membrane. Three gene regions in BCRs, named variable (V), joining (J) and diversity (D), can recognize antigens. These gene regions are spliced and recombined through V(D)J recombination to create enormous BCR diversity–in fact, B cells and BCRs are so diverse that they could potentially recognize every pathogen on earth. The procedure of V(D)J recombination in BCRs is very similar to that in T cell receptors (TCRs), discussed in Section 2.1.2.

B cells need to be activated before responding to pathogens. B cell activation usually requires two steps: cross linking and co-stimulatory signaling. Depending on whether helper T cells are involved, B cell activation can be divided into two types: T cell-dependent activation and T cell-independent activation.

T cell-dependent activation

In T cell-dependent activation, the co-stimulatory signal follows the ligation with

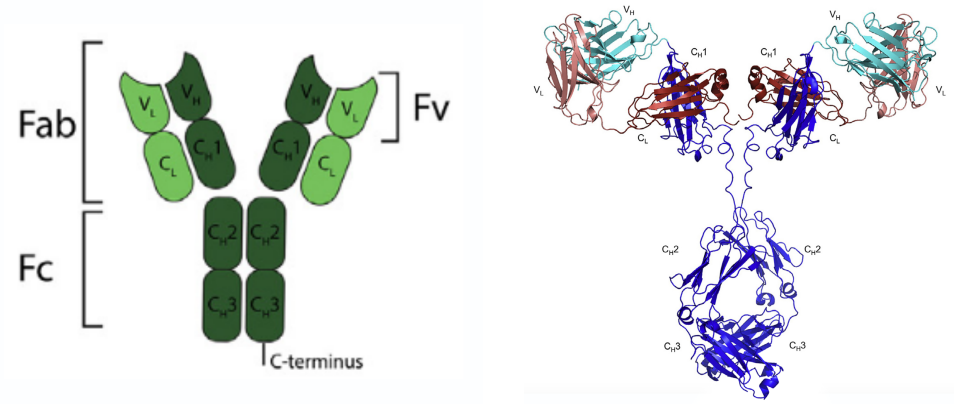
helper T cells. After BCRs bind to an antigen, CD40L on the surface of a helper T cell ligates CD40 on the B cell surface, and this interaction between CD40 and CD40L can supply co-stimulatory signals to activate B cells. In other words, in T cell-dependent activation, B cells and helper T cells work cooperatively when there is an ongoing infection. Since helper T cells only work with protein antigens presented by class II MHC molecules, T cell-dependent activation is limited to these antigens.

T cell-independent activation

In T cell-independent activation, the co-stimulatory signal comes from the interaction between PRRs on B cell surfaces and target molecules on the target antigen. As introduced in Section A.1.1, PRRs directly recognize and bind to invaders. If simultaneously the BCRs are cross linked, B cells are then activated and start to function. Since little or no help from helper T cells is required, T cell-independent activation proceeds much faster than the other activation pathway. Unlike T cell-dependent activation that only works with protein antigens, this activation pathway reacts not only to proteins, but also carbohydrates and lipids, which are found on the cell surface of many common invaders, but rarely found on human cells.

A.2.1.2 Antibodies

Activated B cells produce and release antibodies, or immunoglobulins (Ig). Antibodies are similar to BCRs, but lack the adhesion molecule so they are released freely from B cells. Antibodies recognize antigens via a region called the fragment antigen-binding (Fab) region, while antibodies use the fragment crystallizable (Fc) region to communicate with other members of the immune system (Figure A.5). Each antibody binds to a specific epitope on its cognate antigen via a paratope that is located on the tip of Fab regions. An antibody can therefore neutralize the target antigen or tag the invading microbe for phagocytosis or other action by different components of the immune system. Most antibodies are one of four types: IgM, IgG, IgA, and IgE, all with different functions. Antibody genes can re-organize to change the heavy chain Fc fragment from one antibody isotype to another via the class switching process.



(a) (b)

Figure A.5: Structure of antibodies. (a). Schematic diagram of antibody structure. Dark green: heavy chains, light green: light chains [274]; (b). structure of human IgG [275].

IgM IgM has a half-life of approximately one day. It helps to activate the classical pathway in the complement system. Complement proteins can combine and form a larger protein C1 that binds to the C1-inhibitor to prevent the complement system from spontaneous action. However, if two or more C1 complexes are in close proximity, the C-inhibitors are removed and C1 complexes can begin a series of biochemical reactions that lead to C3 convertase production. In the classical pathway, after IgM binds to an antigen, C1 molecules bind to the Fc regions of IgM and produce C3 convertase to switch on the complement system. This mechanism ensures that the complement system is capable of recognizing and responding to bacteria that have evolved complement protein-resistant surfaces.

IgG IgG is the most abundant antibody in blood—it constitutes approximately 75% of antibodies circulating in the bloodstream. IgG has a relatively long-half life of approximately 3 weeks and can be passed from mother to fetus to provide longer protection before the fetus can produce IgG itself. There exist four subtypes of IgG: IgG1, IgG2, IgG3, and IgG4. IgG binds to antigens via their Fab regions and interacts with other members throughout their Fc regions, which are structurally different between different types of IgG and function differently. For example, IgG1 opsonizes microbes for phagocytosis, IgG2 responds to polysaccharides secreted by many bacteria and fungi, IgG3 provides a link between pathogens and NK cells,

while IgG4 is often involved in long-term response to non-infectious antigens [248, 276]. Therefore, IgG acts like a bridge between antigen and the innate immune system, recognizing the invader and engaging its collaborator to remove the antigen.

IgA While IgG is the most common antibody in blood, IgA is the predominant antibody in the human body: more than 80% of B cells beneath the surface of mucous membranes produce IgA. The most prevalent form of IgA is dimeric form, which enables IgA to move across the intestinal wall and survive exposure to acids and enzymes in the gastrointestinal tract. The IgA dimer has four Fc regions and thus can agglutinate pathogens in the mucus layer to be excreted from the body.

IgE IgE is only found in mammals and the main function is to trigger immune response to parasites. While IgE binds to antigens via its Fab region, its Fc region mainly binds to the Fc receptors on the surface of mast cells and other types of cells in response to parasitic infections.

Appendix B

Amino acid

Table B.1: Standard amino acid abbreviations

Amino acid	3-letter abbreviation	1-letter abbreviation
Alanine	Ala	A
Arginine	Arg	R
Asparagine	Asn	N
Aspartic acid	Asp	D
Cysteine	Cys	C
Glutamine	Gln	Q
Glutamic acid	Glu	E
Glycine	Gly	G
Histidine	His	H
Isoleucine	Ile	I
Leucine	Leu	L
Lysine	Lys	K
Methionine	Met	M
Phenylalanine	Phe	F
Proline	Pro	P
Serine	Ser	S
Threonine	Thr	T
Tryptophan	Trp	W
Tyrosine	Tyr	Y
Valine	Val	V

Appendix C

CFA/OVA/P277-associated CDR3β substrings (fixed length)

Table C.1: Triplets selected by majority vote from 11 subsets of 50,000 random CDR3βs to identify early, all immunized, or all mice using LPBoost and SVM.

Early		All immunized		All	
String	Fisher	String	Fisher	String	Fisher
+ASS	+CAW	+ASS	+CAE	+FFG	+ASG
+CAW	+FGP	+CAW	+CAP	+LYF	+CAE
+FFG	+NTG	+SSY	+CAT		+CAR
+SSR	+SSD	+TQY	+CAW		+FGE
-ASG	+SSR	-ARN	+FGE		+FGH
-SSL	-ASG	-CSA	+NTG		+FGK
-TLY	-FGA	-EQY	+SSD		+FGS
	-FGS	-GTG	+SSR		+GAR
		-SGD	+SYT		-ASQ
		-SSL	-ASG		-FGP
			-FGK		
			-SSL		

Table C.2: Triplets selected by majority vote from 11 subsets of 50,000 random CDR3β singlets to identify early, all immunized, or all mice using LPBoost.

Early		All immunized		All	
String	Fisher	String	Fisher	String	Fisher
+ASS	+CAW	+ASS	+ASS	+ASG	+ASG
+CAW	+FGE	+CAW	+CAW	+CAR	+CAR
+FFG	+FGK	+FGE	+FGE	+LYF	-CAW
-ASG	+SSI	+SSQ	+FGK	+PLY	-FGK
-CAS	+SSQ	+VFF	+SSA	+SPL	-FGP
-GGG	-ASC	-ASG	+SSQ	-ASS	-SSL
-SSL	-ASG	-CAS	+SSY	-CAW	
-TLY	-GGG	-GAG	-ASC	-FGK	
		-GGG	-ASG	-YEQ	
		-SSL	-ASR		
			-FGA		
			-GAR		
			-GGG		

Appendix D

CFA/OVA/P277-associated CDR3β substrings (varying length)

Table D.1: Substrings selected by at least one subset from 11 subsets of 50,000 distinct CDR3s to identify early and all immunized mice using LPBoost. Empty: start from an empty graph; duplets: start from a graph with duplets as nodes. Substrings are sorted by the frequencies being selected.

Empty				Duplets			
Early		All immunized		Early		All immunized	
String	Fisher	String	Fisher	String	Fisher	String	Fisher
-G	-G	+T	-G	+FFG	-ASG	-SSL	+CAE
+N	+N	-G	+T	+ASS	-FGS	-SGD	-FGK
+S	-L	+W	+W	-ASG	+SSR	+TQY	+SSD
+W	+R	-GK	-GK	-TLY	+SSD	-CSA	+CAP
+R	+I	-L	-V	-YFG	+CAW	+ASS	+CAT
+Q	+W	+C	-GS	-ASSL	+FGP	-ARN	-ASG
-V	+Q	-GS	-Q	+SSR	-FGA	-GTG	+CAW
-L	-V	+CA	+K	+CAW	+NTG	+FGE	+SYT
+T	+S	+N	-GT	-SSL	+FGK	-EQY	+SSR
-SG	-GD	+R	-L	+YAE	+SSI	-ASG	+NTG
+F	-SG	+E	+D	-QYFG	+CAT	+YFG	-LYR
+A		+Y	+I		+SYC	+CAW	+FGE

-GD	-A	+R	-SSL	+SSY	-GAR
	-GT	+N	-CAR	+TGQ	-VFG
	-K	-A	+NTE	+SYT	-SSL
	-S	-P	-LYR	+AWS	+SSY
	+S	+GQ	+ASSD	+SSD	+CAR
	+Q	+C		+GQL	-FGH
	-P	-GD		+VRN	-FGP
	-Q	+Y		-GKG	-FGS
	+GQ	+TV		+LYF	-CAY
	-V	+S		+RNV	+VTG
	+D	-CS		-AEQ	-SSLA
	-SL	-GA		-GAR	+GQS
	-KG	+E		-SSQ	-SSQ
	-F	-S		-QFF	+CAL
	-GKG	+CD		+VFF	-SLA
	+A	+QS		+SYA	+FGA
	+WG	-GG		-CGA	+CARS
	-SG	+H		+ASSY	-CAR
	-CS	+AW		-TEV	+SSI
		+QL		+QLY	-SSV
		+TD		+GSG	
		-SL		+SSR	
				-AET	

Appendix E

Mean F measure

Table E.1: Average F measure using fixed threshold to distinguish selection probabilities of relevant and irrelevant features.

	w_S	Lasso	Elastic net	Excl. (fixed)	Excl. (random)
Example 1	0.1	0.6058	0.7543	0.9114	0.8668
	0.3	0.7294	0.8943	0.9783	0.9635
	0.5	0.7471	0.8781	0.9657	0.9517
Example 2	0.1	0.5929	0.7436	0.8919	0.8423
	0.3	0.7416	0.9323	0.9913	0.9765
	0.5	0.7376	0.9209	0.9720	0.9624
Example 3	0.1	0.5022	0.5332	0.6331	0.5798
	0.3	0.6286	0.6860	0.8067	0.7135
	0.5	0.6989	0.7282	0.8308	0.7265
Example 4	0.1	0.5016	0.5069	0.5656	0.5259
	0.3	0.7196	0.6776	0.7259	0.6449
	0.5	0.8092	0.7581	0.7760	0.6779

Appendix F

Selection probabilities of one dataset

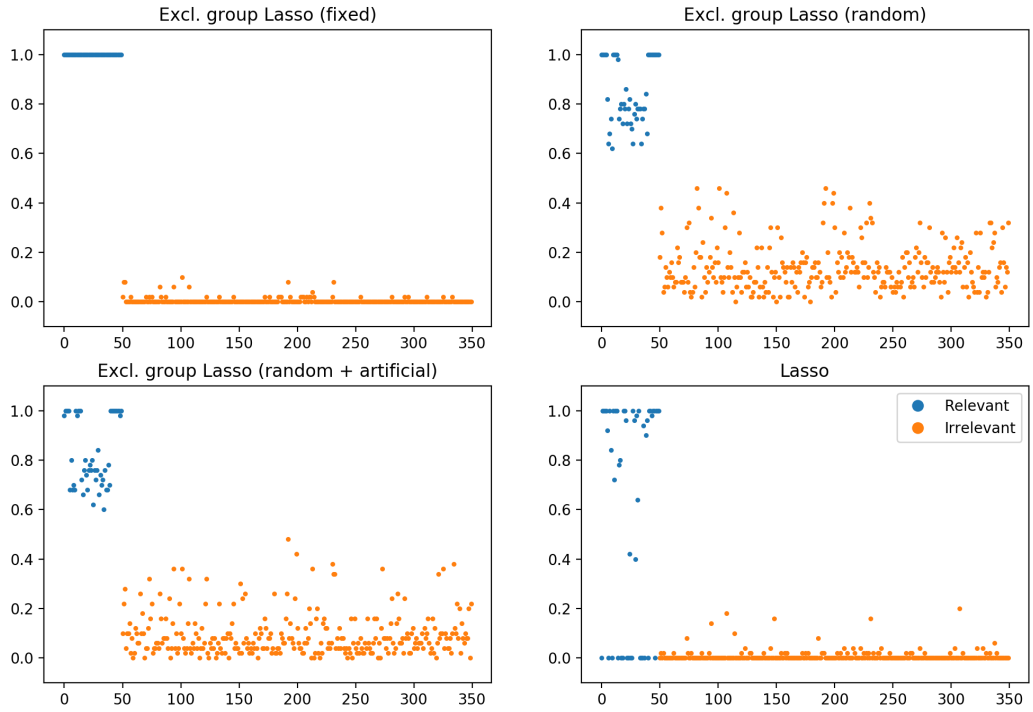


Figure F.1: Selection probabilities using Lasso and exclusive group Lasso with Example 3.

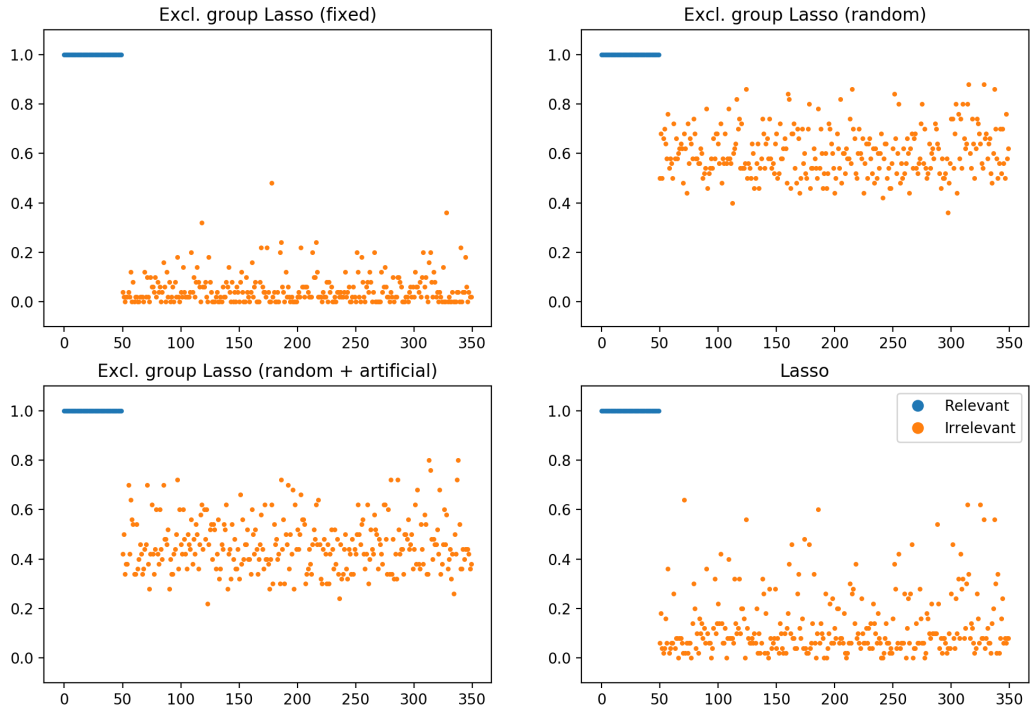


Figure F.2: Selection probabilities using Lasso and exclusive group Lasso with Example 4.

Appendix G

CMV-associated TCRβs in the US cohort

Table G.1: CMV-associated TCRβs identified by exclusive group Lasso with artificial features and stability selection.

CDR3	V gene	J gene
CASSGDRLYEQYF	TCRBV02-01*01	TCRBJ02-07*01
CASSPQRNTEAFF	TCRBV04-03*01	TCRBJ01-01*01
CASSLGHRDPNTGELFF	TCRBV05-01*01	TCRBJ02-02*01
CASSLVIGGDTEAFF	TCRBV05-01*01	TCRBJ01-01*01
CASSPDRVGQETQYF	TCRBV05-01*01	TCRBJ02-05*01
CASSLDRDEQYF	TCRBV05-04*01	TCRBJ02-07*01
CASSLLWDQPQHF	TCRBV05-05*01	TCRBJ01-05*01
CASSLRREKLFF	TCRBV05-06*01	TCRBJ01-04*01
CASSLVAGGRETQYF	TCRBV05-06*01	TCRBJ02-05*01
CASRPTGYEQYF	TCRBV06-01*01	TCRBJ02-07*01
CASSEARTRAFF	TCRBV06-01*01	TCRBJ01-01*01
CASSTGTSGSYEQYF	TCRBV06-01*01	TCRBJ02-07*01
CASSEIPNTEAFF	TCRBV06-04*	TCRBJ01-01*01
CASSYVRTGGNYGYTF	TCRBV06-05*01	TCRBJ01-02*01
CASSLEAEYEQYF	TCRBV07-02*01	TCRBJ02-07*01
CASSRLAGGTDTQYF	TCRBV07-03*01	TCRBJ02-03*01

CASSLAPGATNEKLFF	TCRBV07-06*01	TCRBJ01-04*01
CASSRGRQETQYF	TCRBV07-06*01	TCRBJ02-05*01
CASSLQGADTQYF	TCRBV07-08*01	TCRBJ02-03*01
CASSFPTSGQETQYF	TCRBV07-09*	TCRBJ02-05*01
CASSHRDRNYEQYF	TCRBV07-09*	TCRBJ02-07*01
CASSLIGVSSYNEQFF	TCRBV07-09*	TCRBJ02-01*01
CASSAGQGVTYEQYF	TCRBV09-01*	TCRBJ02-07*01
CASSGQGAYEQYF	TCRBV09-01*	TCRBJ02-07*01
CASSESGHRNQPQHF	TCRBV10-02*01	TCRBJ01-05*01
CASSSGQVQETQYF	TCRBV11-02*02	TCRBJ02-05*01
CASSLGGAGDTQYF	TCRBV12*	TCRBJ02-03*01
CASSQTGGRNQPQHF	TCRBV12*	TCRBJ01-05*01
CASSQNRGQETQYF	TCRBV14-01*01	TCRBJ02-05*01
CASSQVPGQGDNEQFF	TCRBV14-01*01	TCRBJ02-01*01
CATSRDSQGSYGYTF	TCRBV15-01*01	TCRBJ01-02*01
CATSRDTQGSYGYTF	TCRBV15-01*01	TCRBJ01-02*01
CATSRGTVSYEQYF	TCRBV15-01*01	TCRBJ02-07*01
CASSPPGQGSDTQYF	TCRBV18-01*01	TCRBJ02-03*01
CASSIGPLEHNEQFF	TCRBV19-01*	TCRBJ02-01*01
CASSIWGLDTEAFF	TCRBV19-01*	TCRBJ01-01*01
CASSPAGLNTEAFF	TCRBV19-01*	TCRBJ01-01*01
CASSTTGGDGYTF	TCRBV19-01*	TCRBJ01-02*01
CSASPGQGASYGYTF	TCRBV20*	TCRBJ01-02*01
CATSDGDEQFF	TCRBV24*	TCRBJ02-01*01
CATSDGETQYF	TCRBV24*	TCRBJ02-05*01
CASSPGDEQFF	TCRBV25-01*01	TCRBJ02-01*01
CASSPGDEQYF	TCRBV25-01*01	TCRBJ02-07*01
CASSIEGNQPQHF	TCRBV28-01*01	TCRBJ01-05*01
CASSLPSGLTDTQYF	TCRBV28-01*01	TCRBJ02-03*01
CASSPPSGLTDTQYF	TCRBV28-01*01	TCRBJ02-03*01

CSVRDNFNQPQHF	TCRBV29-01*01	TCRBJ01-05*01

Appendix H

CMV-associated TCRβs in the Belgium cohort

Table H.1: Overlapped TCRβs between TCRβs discovered by exclusive group Lasso and in Cohort 3.

CDR3	V gene	J gene
CASSLGHRDPNTGELFF	TCRBV05-01*01	TCRBJ02-02*01
CASSLDRDEQYF	TCRBV05-04*01	TCRBJ02-07*01
CASSLRREKLFF	TCRBV05-06*01	TCRBJ01-04*01
CASSLVAGGRETQYF	TCRBV05-06*01	TCRBJ02-05*01
CASRPTGYEQYF	TCRBV06-01*01	TCRBJ02-07*01
CASSYVRTGGNYGYTF	TCRBV06-05*01	TCRBJ01-02*01
CASSLEAEYEQYF	TCRBV07-02*01	TCRBJ02-07*01
CASSRLAGGTDTQYF	TCRBV07-03*01	TCRBJ02-03*01
CASSRGRQETQYF	TCRBV07-06*01	TCRBJ02-05*01
CASSGQGAYEQYF	TCRBV09-01*	TCRBJ02-07*01
CASSESGHRNQPQHF	TCRBV10-02*01	TCRBJ01-05*01
CASSLGGAGDTQYF	TCRBV12*	TCRBJ02-03*01
CASSQNRGQETQYF	TCRBV14-01*01	TCRBJ02-05*01
CASSPPGQGSDTQYF	TCRBV18-01*01	TCRBJ02-03*01
CASSIWGLDTEAFF	TCRBV19-01*	TCRBJ01-01*01

CASSPAGLNTEAFF	TCRBV19-01*	TCRBJ01-01*01
CATSDGETQYF	TCRBV24*	TCRBJ02-05*01
CASSIEGNQPQHF	TCRBV28-01*01	TCRBJ01-05*01
CASSLPSGLTDTQYF	TCRBV28-01*01	TCRBJ02-03*01
CASSPPSGLTDTQYF	TCRBV28-01*01	TCRBJ02-03*01
CSVRDNFNQPQHF	TCRBV29-01*01	TCRBJ01-05*01

Bibliography

[1] C. A. Janeway, P. Travers, M. Walport, and M. J. Shlomchik, *Immunobiology*. New York: Garland Science, 5 ed., 2001.

[2] B. Alberts, A. Johnson, J. L. abd Martin Raff, K. Roberts, and P. Walter, *Molecular Biology of the Cell*. New York: Garland Science, 4 ed., 2002.

[3] R. Medzhitov, "Recognition of microorganisms and activation of the immune response," *Nature*, vol. 449, no. 7164, pp. 819–826, 2007.

[4] O. Takeuchi and S. Akira, "Pattern recognition receptors and inflammation," *Cell*, vol. 140, no. 6, pp. 805–820, 2010.

[5] T. Kawai and S. Akira, "The role of pattern-recognition receptors in innate immunity: update on Toll-like receptors," *Nature Immunology*, vol. 11, no. 5, pp. 373–384, 2010.

[6] R. M. Chicz, R. G. Urban, W. S. Lane, J. C. Gorga, L. J. Stern, D. A. A. Vignali, and J. L. Strominger, "Predominant naturally processed peptides bound to HLA-DR1 are derived from MHC-related molecules and are heterogeneous in size," *Nature*, vol. 358, pp. 764–768, 1992.

[7] P. M. van Endert, R. Tampé, T. H. Meyer, R. Tisch, J.-F. Bach, and H. O. McDevitt, "A sequential model for peptide binding and transport by the transporters associated with antigen processing," *Immunity*, vol. 1, no. 6, pp. 491–500, 1994.

[8] A. J. Gentles, A. M. Newman, C. L. Liu, S. V. Bratman, W. Feng, D. Kim, V. S. Nair, Y. Xu, A. Khuong, C. D. Hoang, M. Diehn, R. B. West, S. K.

Plevritis, and A. A. Alizadeh, "The prognostic landscape of genes and infiltrating immune cells across human cancers," *Nature Medicine*, vol. 21, pp. 938–945, 2015.

[9] N. E. McCarthy and M. Eberl, "Human $\gamma\delta$ T-cell control of mucosal immunity and inflammation," *Frontiers in Immunology*, vol. 9, no. 985, 2018.

[10] C. T. Morita, R. A. Mariuzza, and M. B. Brenner, "Antigen recognition by human $\gamma\delta$ T cells: pattern recognition by the adaptive immune system," *Springer Seminars in Immunopathology*, vol. 22, no. 3, pp. 191–217, 2000.

[11] M. Attaf, E. Huseby, and A. K. Sewell, "$\alpha\beta$ T cell receptors as predictors of health and disease," *Cellular & Molecular Immunology*, vol. 12, no. 4, pp. 391–399, 2015.

[12] J. L. Xu and M. M. Davis, "Diversity in the CDR3 region of VH is sufficient for most antibody specificities," *Immunity*, vol. 13, no. 1, pp. 37–45, 2000.

[13] M. M. Davis and P. J. Bjorkman, "T-cell antigen receptor genes and T-cell recognition," *Nature*, vol. 334, no. 6181, pp. 395–402, 1988.

[14] E. P. Rock, P. R. Sibbald, M. M. Davis, and Y. H. Chien, "CDR3 length in antigen-specific immune receptors," *Journal of Experimental Medicine*, vol. 179, no. 1, pp. 323–328, 1994.

[15] T. P. Arstila, A. Casrouge, V. Baron, J. Even, J. Kanellopoulos, and P. Kourilsky, "A direct estimate of the human $\alpha\beta$ T cell receptor diversity," *Science*, vol. 286, no. 5441, pp. 958–961, 1999.

[16] H. S. Robins, P. V. Campregher, S. K. Srivastava, A. Wacher, C. J. Turtle, O. Kahsai, S. R. Riddell, E. H. Warren, and C. S. Carlson, "Comprehensive assessment of T-cell receptor β-chain diversity in $\alpha\beta$ T cells," *Blood*, vol. 114, no. 19, pp. 4099–4107, 2009.

[17] K. Natarajan, J. Jiang, N. A. May, M. G. Mage, L. F. Boyd, A. C. McShan, N. G. Sgourakis, A. Bax, and D. H. Margulies, "The role of molecular flexibility in antigen presentation and T cell receptor-mediated signaling," *Frontiers in Immunology*, vol. 9, no. 1657, 2018.

[18] P. Parham, *The Immune System*. Garland Science, 2 ed., 2004.

[19] C. Heufler, F. Koch, U. Stanzl, G. Topar, M. Wysocka, G. Trinchieri, A. Enk, R. M. Steinman, N. Romani, and G. Schuler, "Interleukin-12 is produced by dendritic cells and mediates T helper 1 development as well as interferon-gamma production by T helper 1 cells," *European Journal of Immunology*, vol. 26, no. 3, pp. 659–668, 1996.

[20] E. L. Rael and R. F. Lockey, "Interleukin-13 signaling and its role in asthma," *World Allergy Organization Journal*, vol. 4, no. 3, pp. 54–64, 2011.

[21] F. Bonnefoy, M. Couturier, A. Clauzon, J.-P. Rémy-Martin, B. Gaugler, P. Tiberghien, W. Chen, P. Saas, and S. Perruche, "TGF-β-exposed plasmacytoid dendritic cells participate in Th17 commitment," *The Jounal of Immunology*, vol. 186, no. 11, pp. 6157–6164, 2011.

[22] M. A. Cooper and W. M. Yokoyama, "Memory-like responses of natural killer cells," *Immunological Reviews*, vol. 235, no. 1, pp. 297–305, 2010.

[23] M. S. Asano and R. Ahmed, "CD8 T cell memory in B cell-deficient mice," *Journal of Experimental Medicine*, vol. 183, no. 5, pp. 2165–2174, 1996.

[24] F. M. Burnet, "A modification of Jerne's theory of antibody production using the concept of clonal selection," *Australian Journal of Science*, 1957.

[25] M. K. Jenkins, H. H. Chu, J. B. McLachlan, and J. J. Moon, "On the composition of the preimmune repertoire of T cells specific for peptide-major histocompatibility complex ligands," *Annual Review of Immunology*, vol. 28, pp. 275–294, 2009.

[26] G. Lythe, R. E. Callard, R. L. Hoare, and C. Molina-Paris, "How many TCR clonotypes does a body maintain?," *Journal of Theoretical Biology*, vol. 389, pp. 214–224, 2016.

[27] 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 & Immunity*, vol. 17, no. 3, pp. 153–164, 2016.

[28] M. M. Davis and P. J. Bjorkman, "T-cell antigen receptor genes and T-cell recognition," *Nature*, vol. 334, no. 6181, pp. 395–402, 1988.

[29] E. Hodges, M. T. Krishna, C. Pickard, and J. L. Smith, "Diagnostic role of tests for T cell receptor (TCR) genes," *Journal of clinical pathology*, vol. 56, no. 1, pp. 1–11, 2003.

[30] J. Nikolich-Zugich, M. K. Slifka, and I. Messaoudi, "The many important facets of T-cell repertoire diversity," *Nature Reviews Immunology*, vol. 4, pp. 123–132, 2004.

[31] D. J. Laydon, C. R. M. Bangham, and B. Asquith, "Estimating T-cell repertoire diversity: limitations of classical estimators and a new approach," *Philosophical Transactions of the Royal Society B*, vol. 370, 2014.

[32] T. P. Arstila, A. Casrouge, V. Baron, J. Even, J. Kanellopoulos, and P. Kourilsky, "A direct estimate of the human $\alpha\beta$ T cell receptor diversity," *Science*, vol. 286, no. 5441, pp. 958–961, 1999.

[33] C. Keşmir, J. A. Borghans, and R. J. de Boer, "Diversity of human $\alpha\beta$ T cell receptors," *Science*, vol. 288, no. 5469, pp. 1135–1135, 2000.

[34] T. Cukalac, W.-T. Kan, P. Dash, J. Guan, K. M. Quinn, S. Gras, P. G. Thomas, and N. L. La Gruta, "Paired TCR$\alpha\beta$ analysis of virus-specific CD8+ T cells exposes diversity in a previously defined a 'narrow' repertoire," *Immunology & Cell Biology*, vol. 93, no. 9, pp. 804–814, 2015.

[35] R. L. Warren, J. D. Freeman, T. Zeng, G. Choe, S. Munro, R. Moore, J. R. Webb, and R. 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*, vol. 21, no. 5, pp. 790–797, 2011.

[36] R. Cibotti, J. P. Cabaniols, C. Pannetier, C. Delarbre, I. Vergnon, J. M. Kanellopoulos, and P. Kourilsky, "Public and private V beta T cell receptor repertoires against hen egg white lysozyme (HEL) in nontransgenic versus HEL transgenic mice," *The Journal of experimental medicine*, vol. 180, no. 3, pp. 861–872, 1994.

[37] P. Boudinot, S. Boubekeur, and A. Benmansour, "Rhabdovirus infection induces public and private T cell responses in teleost fish," *The Journal of Immunology*, vol. 167, no. 11, pp. 6202–6209, 2001.

[38] H. Li, C. Ye, G. Ji, and J. Han, "Determinants of public T cell responses," *Cell research*, vol. 22, no. 1, pp. 33–42, 2012.

[39] W. S. DeWitt 3rd, A. Smith, G. Schoch, J. A. Hansen, F. A. Matsen 4th, and P. Bradley, "Human T cell receptor occurrence patterns encode immune history, genetic background, and receptor specificity," *eLife*, vol. 7, p. e38358, 2018.

[40] A. Sainz-Perez, A. Lim, B. Lemercier, and C. Leclerc, "The T-cell receptor repertoire of tumor-infiltrating regulatory T lymphocytes is skewed toward public sequences," *Cancer Research*, vol. 72, no. 14, pp. 3557–3569, 2012.

[41] P. Bousso, A. Casrouge, J. D. Altman, M. Haury, J. Kanellopoulos, J.-P. Abastado, and P. Kourilsky, "Individual variations in the murine T cell response to a specific peptide reflect variability in naive repertoires," *Immunity*, vol. 9, no. 2, pp. 169–178, 1998.

[42] 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 research*, vol. 24, no. 10, pp. 1603–1612, 2014.

[43] H. S. Robins, S. K. Srivastava, P. V. Campregher, C. J. Turtle, J. Andriesen, S. R. Riddell, C. S. Carlson, and E. H. Warren, "Overlap and effective size of the human CD8+ T cell receptor repertoire," *Science translational medicine*, vol. 2, no. 47, pp. 47ra64–47ra64, 2010.

[44] V. Venturi, M. F. Quigley, H. Y. Greenaway, P. C. Ng, Z. S. Ende, T. McIntosh, T. E. Asher, J. R. Almeida, S. Levy, D. A. Price, M. P. Davenport, and D. C. Douek, "A mechanism for TCR sharing between T cell subsets and individuals revealed by pyrosequencing," *The Journal of Immunology*, vol. 186, no. 7, pp. 4285–4294, 2011.

[45] A. Murugan, T. Mora, A. M. Walczak, and C. G. Callan, "Statistical inference of the generation probability of T-cell receptors from sequence repertoires," *Proceedings of the National Academy of Sciences*, vol. 109, no. 40, pp. 16161–16166, 2012.

[46] W. Ndifon, H. Gal, E. Shifrut, R. Aharoni, N. Yissachar, N. Waysbort, S. Reich-Zeliger, R. Arnon, and N. Friedman, "Chromatin conformation governs T-cell receptor Jβ gene segment usage," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 109, no. 39, pp. 15865–15870, 2012.

[47] 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 of the United States of America*, vol. 103, no. 49, pp. 18691–18696, 2006.

[48] V. Venturi, D. A. Price, D. C. Douek, and M. P. Davenport, "The molecular basis for public T-cell responses?," *Nature Reviews Immunology*, vol. 8, pp. 231–238, 2008.

[49] P. A. Moss, R. J. Moots, W. M. Rosenberg, S. J. Rowland-Jones, H. C. Bodmer, A. J. McMichael, and J. I. Bell, "Extensive conservation of α and β chains of the human T-cell antigen receptor recognizing HLA-A2 and influenza A matrix peptide," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 88, no. 20, pp. 8987–8990, 1991.

[50] P. J. Lehner, E. C. Wang, P. A. Moss, S. Williams, K. Platt, S. M. Friedman, J. I. Bell, and L. K. Borysiewicz, "Human HLA-A0201-restricted cytotoxic T lymphocyte recognition of influenza A is dominated by T cells bearing the V beta 17 gene segment," *The Journal of experimental medicine*, vol. 181, no. 1, pp. 79–91, 1995.

[51] N. Khan, M. Cobbold, R. Keenan, and P. A. H. Moss, "Comparative analysis of CD8+ T cell responses against human cytomegalovirus proteins pp65 and immediate early 1 shows similarities in precursor frequency, oligoclonality, and phenotype," *Journal of Infectious Diseases*, vol. 185, no. 8, pp. 1025–1034, 2002.

[52] D. A. Price, J. M. Brenchley, L. E. Ruff, M. R. Betts, B. J. Hill, M. Roederer, R. A. Koup, S. A. Migueles, E. Gostick, L. Wooldridge, A. K. Sewell, M. Connors, and D. C. Douek, "Avidity for antigen shapes clonal dominance in CD8+ T cell populations specific for persistent DNA viruses," *Journal of Experimental Medicine*, vol. 202, no. 10, pp. 1349–1361, 2005.

[53] M. P. Weekes, M. R. Wills, K. Mynard, A. J. Carmichael, and J. G. P. Sissons, "The memory cytotoxic T-lymphocyte (CTL) response to human cytomegalovirus infection contains individual peptide-specific CTL clones that have undergone extensive expansion in vivo," *Journal of Virology*, vol. 73, no. 3, pp. 2099–2108, 1999.

[54] V. P. Argaet, C. W. Schmidt, S. R. Burrows, S. L. Silins, M. G. Kurilla, D. L. Doolan, A. Suhrbier, D. J. Moss, E. Kieff, T. B. Sculley, and I. S. Misko, "Dominant selection of an invariant T cell antigen receptor in re-

sponse to persistent infection by Epstein-Barr virus," *The Journal of experimental medicine*, vol. 180, no. 6, pp. 2335–2340, 1994.

[55] N. E. Annels, M. F. C. Callan, L. Tan, and A. B. Rickinson, "Changing patterns of dominant TCR usage with maturation of an EBV-specific cytotoxic T cell response," *The Journal of Immunology*, vol. 165, no. 9, pp. 4831–4841, 2000.

[56] A. Lim, L. Trautmann, M.-A. Peyrat, C. Couedel, F. Davodeau, F. Romagné, P. Kourilsky, and M. Bonneville, "Frequent contribution of T cell clonotypes with public TCR features to the chronic response against a dominant EBV-derived epitope: Application to direct detection of their molecular imprint on the human peripheral T cell repertoire," *The Journal of Immunology*, vol. 165, no. 4, pp. 2001–2011, 2000.

[57] J. J. Miles, D. Elhassen, N. A. Borg, S. L. Silins, F. E. Tynan, J. M. Burrows, A. W. Purcell, L. Kjer-Nielsen, J. Rossjohn, S. R. Burrows, and J. McCluskey, "CTL recognition of a bulged viral peptide involves biased TCR selection," *The Journal of Immunology*, vol. 175, no. 6, pp. 3826–3834, 2005.

[58] G. M. A. Gillespie, G. Stewart-Jones, J. Rengasamy, T. Beattie, J. J. Bwayo, F. A. Plummer, R. Kaul, A. J. McMichael, P. Easterbrook, T. Dong, E. Y. Jones, and S. L. Rowland-Jones, "Strong TCR conservation and altered T cell cross-reactivity characterize a B*57-restricted immune response in HIV-1 infection," *The Journal of Immunology*, vol. 177, no. 6, pp. 3893–3902, 2006.

[59] X. G. Yu, M. Lichterfeld, S. Chetty, K. L. Williams, S. K. Mui, T. Miura, N. Frahm, M. E. Feeney, Y. Tang, F. Pereyra, M. X. Labute, K. Pfafferott, A. Leslie, H. Crawford, R. Allgaier, W. Hildebrand, R. Kaslow, C. Brander, T. M. Allen, E. S. Rosenberg, P. Kiepiela, M. Vajpayee, P. A. Goepfert, M. Altfeld, P. J. R. Goulder, and B. D. Walker, "Mutually exclusive T-cell receptor induction and differential susceptibility to human immunodeficiency

virus type 1 mutational escape associated with a two-amino-acid difference between HLA class I subtypes," *Journal of virology*, vol. 81, no. 4, pp. 1619–1631, 2007.

[60] D. A. Price, S. M. West, M. R. Betts, L. E. Ruff, J. M. Brenchley, D. R. Ambrozak, Y. Edghill-Smith, M. J. Kuroda, D. Bogdan, K. Kunstman, N. L. Letvin, G. Franchini, S. M. Wolinsky, R. A. Koup, and D. C. Douek, "T cell receptor recognition motifs govern immune escape patterns in acute SIV infection," *Immunity*, vol. 21, no. 6, pp. 793–803, 2004.

[61] D. A. Price, T. E. Asher, N. A. Wilson, M. C. Nason, J. M. Brenchley, I. S. Metzler, V. Venturi, E. Gostick, P. K. Chattopadhyay, M. Roederer, M. P. Davenport, D. I. Watkins, and D. C. Douek, "Public clonotype usage identifies protective Gag-specific CD8+ T cell responses in SIV infection," *Journal of Experimental Medicine*, vol. 206, no. 4, pp. 923–936, 2009.

[62] J. S. Menezes, P. van den Elzen, J. Thornes, D. Huffman, N. M. Droin, E. Maverakis, and E. E. Sercarz, "A public T cell clonotype within a heterogeneous autoreactive repertoire is dominant in driving EAE," *The Journal of clinical investigation*, vol. 117, no. 8, pp. 2176–2185, 2007.

[63] N. Fazilleau, C. Delarasse, I. Motta, S. Fillatreau, M.-L. Gougeon, P. Kourilsky, D. Pham-Dinh, and J. M. Kanellopoulos, "T cell repertoire diversity is required for relapses in myelin oligodendrocyte glycoprotein-induced experimental autoimmune encephalomyelitis," *The Journal of Immunology*, vol. 178, no. 8, pp. 4865–4875, 2007.

[64] C. L. O'Keefe, R. M. Sobecks, M. Wlodarski, A. Rodriguez, K. Bell, E. Kuczkowski, B. J. Bolwell, and J. P. Maciejewski, "Molecular TCR diagnostics can be used to identify shared clonotypes after allogeneic hematopoietic stem cell transplantation," *Experimental Hematology*, vol. 32, no. 10, pp. 1010–1022, 2004.

[65] L. K. Ely, K. J. Green, T. Beddoe, C. S. Clements, J. J. Miles, S. P. Bottomley,

D. Zernich, L. Kjer-Nielsen, A. W. Purcell, J. McCluskey, J. Rossjohn, and S. R. Burrows, "Antagonism of antiviral and allogeneic activity of a human public CTL clonotype by a single altered peptide ligand: Implications for allograft rejection," *The Journal of Immunology*, vol. 174, no. 9, pp. 5593–5601, 2005.

[66] M. Khosravi-Maharlooei, A. Obradovic, A. Misra, K. Motwani, M. Holzl, H. R. Seay, S. DeWolf, G. Nauman, N. Danzl, H. Li, S.-H. Ho, R. Winchester, Y. Shen, T. M. Brusko, and M. Sykes, "Crossreactive public TCR sequences undergo positive selection in the human thymic repertoire," *The Journal of clinical investigation*, vol. 129, no. 6, pp. 2446–2462, 2019.

[67] T. Hopf, L. Colwell, R. Sheridan, B. Rost, C. Sander, and D. Marks, "Three-dimensional structures of membrane proteins from genomic sequencing," *Cell*, vol. 149, no. 7, pp. 1607–1621, 2012.

[68] R. Brenke, D. R. Hall, G.-Y. Chuang, S. R. Comeau, T. Bohnuud, D. Beglov, O. Schueler-Furman, S. Vajda, and D. Kozakov, "Application of asymmetric statistical potentials to antibody-protein docking," *Bioinformatics*, vol. 28, no. 20, pp. 2608–2614, 2012.

[69] N. De Neuter, W. Bittremieux, C. Beirnaert, B. Cuypers, A. Mrzic, P. Moris, A. Suls, V. Van Tendeloo, B. Ogunjimi, K. Laukens, and P. Meysman, "On the feasibility of mining CD8+ T cell receptor patterns underlying immunogenic peptide recognition," *Immunogenetics*, vol. 70, no. 3, pp. 159–168, 2018.

[70] N. Thakkar and C. Bailey-Kellogg, "Balancing sensitivity and specificity in distinguishing TCR groups by CDR sequence similarity," *BMC bioinformatics*, vol. 20, no. 1, p. 241, 2019.

[71] N. Thomas, K. Best, M. Cinelli, S. Reich-Zeliger, H. Gal, E. Shifrut, A. Madi, N. Friedman, J. Shawe-Taylor, and B. 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*, vol. 30, no. 22, pp. 3181–3188, 2014.

[72] V. Greiff, C. R. Weber, J. Palme, U. Bodenhofer, E. Miho, U. Menzel, and S. T. Reddy, "Learning the high-dimensional immunogenomic features that predict public and private antibody repertoires," *The Journal of Immunology*, vol. 199, no. 8, pp. 2985–2997, 2017.

[73] A.-M. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes (New Methods for the Determination of the Orbits of Comets)*. Paris: F. Didot, 1805.

[74] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical Learning with Sparsity: the Lasso and Generalizations*. Chapman and Hall, 2015.

[75] A. N. Tikhonov, "Solution of incorrectly posed problems and the regularization method," *Soviet Mathematics*, vol. 4, pp. 1035–1038, 1963.

[76] D. L. Phillips, "A technique for the numerical solution of certain integral equations of the first kind," *Journal of the ACM*, vol. 9, no. 1, pp. 84–97, 1962.

[77] A. E. Hoerl, "Application of ridge analysis to regression problems," *Chemical Engineering Progress*, vol. 58, no. 3, pp. 54–59, 1962.

[78] A. E. Hoerl and R. W. Kennard, "Ridge regression: biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.

[79] E. M. L. Beale, M. G. Kendall, and D. W. Mann, "The discarding of variables in multivariate analysis," *Biometrika*, vol. 54, no. 3/4, pp. 357–366, 1967.

[80] R. R. Hocking and R. N. Leslie, "Selection of the best subset in regression analysis," *Technometrics*, vol. 9, no. 4, pp. 531–540, 1967.

[81] R. Tibshirani, "Regression shrinkage and selection via the Lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[82] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least angle regression," *Annals of Statistics*, pp. 407–499, 2004.

[83] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. Springer, 2 ed., 2009.

[84] W. J. Fu, "Penalized regressions: the bridge versus the Lasso," *Journal of Computational and Graphical Statistics*, vol. 7, no. 3, pp. 397–416, 1998.

[85] J. Friedman, T. Hastie, H. Hofling, and R. Tibshirani, "Pathwise coordinate optimization," *The Annals of Applied Statistics*, vol. 1, no. 2, pp. 302–332, 2007.

[86] T. T. Wu and K. Lange, "Coordinate descent algorithms for Lasso penalized regression," *The Annals of Applied Statistics*, vol. 2, no. 1, pp. 224–244, 2008.

[87] J. Friedman, T. Hastie, and R. Tibshirani, "Regularization paths for generalized linear models via coordinate descent," *Journal of Statistical Software*, vol. 33, no. 1, pp. 1–21, 2010.

[88] N. Parikh and S. Boyd, "Proximal algorithms," *Foundations and Trends in Optimization*, vol. 1, no. 3, pp. 127–239, 2014.

[89] S. Mosci, L. Rosasco, M. Santoro, A. Verri, and S. Villa, "Solving structured sparsity regularization with proximal methods," in *Machine Learning and Knowledge Discovery in Databases* (J. L. Balcázar, F. Bonchi, A. Gionis, and M. Sebag, eds.), pp. 418–433, Springer Berlin Heidelberg, 2010.

[90] K. P. Murphy, *Machine Learning: a Probabilistic Perspective*. MIT Press, 2012.

[91] P. Tseng, "Coordinate ascent for maximizing nondifferentiable concave functions," Tech. Rep. LIDS-P; 1840, Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, 1988.

[92] P. Tseng, "Convergence of a block coordinate descent method for nondifferentiable minimization," *Journal of Optimization Theory and Applications*, vol. 109, no. 3, pp. 475–494, 2001.

[93] Z. Q. Luo and P. Tseng, "On the convergence of the coordinate descent method for convex differentiable minimization," *Journal of Optimization Theory and Applications*, vol. 72, no. 1, pp. 7–35, 1992.

[94] A. Genkin, D. D. Lewis, and D. Madigan, "Large-scale Bayesian logistic regression for text categorization," *Technometrics*, vol. 49, no. 3, pp. 291–304, 2007.

[95] H.-F. Yu, F.-L. Huang, and C.-J. Lin, "Dual coordinate descent methods for logistic regression and maximum entropy models," *Machine Learning*, vol. 85, no. 1-2, pp. 41–75, 2011.

[96] I. Trofimov and A. Genkin, "Distributed coordinate descent for ℓ_1-regularized logistic regression," in *Analysis of Images, Social Networks and Texts* (M. Y. Khachay, N. Konstantinova, A. Panchenko, D. Ignatov, and V. G. Labunets, eds.), pp. 243–254, 2015.

[97] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[98] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan, "A dual coordinate descent method for large-scale linear SVM," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 408–415, 2008.

[99] R. Tibshirani, M. Saunders, S. Rosset, J. Zhu, and K. Knight, "Sparsity and smoothness via the fused Lasso," *Journal of the Royal Statistical Society. Series B (statistical Methodology)*, vol. 67, no. 1, pp. 91–108, 2005.

[100] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proceedings Of The 5Th Annual Workshop On Computational Learning Theory*, pp. 144–152, 1992.

[101] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[102] J. Zhu, S. Rosset, T. Hastie, and R. Tibshirani, "1-norm support vector machines," in *Advances in Neural Information Processing Systems 16* (S. Thrun, L. K. Saul, and B. Schölkopf, eds.), pp. 49–56, MIT Press, 2004.

[103] G. M. Fung and O. Mangasarian, "A feature selection newton method for support vector machine classification," *Computational Optimization and Applications*, vol. 28, no. 2, pp. 185–202, 2004.

[104] A. Demiriz, K. P. Bennett, and J. Shawe-Taylor, "Linear programming boosting via column generation," *Machine Learning*, vol. 46, no. 1–3, pp. 225–254, 2002.

[105] H. Zou and T. Hastie, "Regularization and variable selection via the elastic net," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.

[106] A. J. van der Kooij, *Prediction Accuracy and Stability of Regrsssion with Optimal Scaling Transformations*. PhD thesis, Leiden University, 2007.

[107] M. Yuan and Y. Lin, "Model selection and estimation in regression with grouped variables," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 68, pp. 49–67, 2006.

[108] Y. Nesterov, "Gradient methods for minimizing composite objective function," Tech. Rep. 2007076, Université catholique de Louvain, Center for Operations Research and Econometrics (CORE), 2007.

[109] J. Liu, S. Ji, and J. Ye, "Multi-task feature learning via efficient $\ell_{2,1}$-norm minimization," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pp. 339–348, 2009.

[110] Z. Qin, K. Scheinberg, and D. Goldfarb, "Efficient block-coordinate descent algorithms for the group Lasso," *Mathematical Programming Computation*, vol. 5, no. 2, pp. 143–169, 2013.

[111] J. Friedman, T. Hastie, and R. Tibshirani, "A note on the group lasso and a sparse group lasso," *arXiv:1001.0736 [math.ST]*, 2010.

[112] N. Simon, J. Friedman, T. Hastie, and R. Tibshirani, "A sparse-group Lasso," *Journal of Computational and Graphical Statistics*, vol. 22, no. 2, pp. 231–245, 2013.

[113] D. Kong, R. Fujimaki, J. Liu, F. Nie, and C. Ding, "Exclusive feature learning on arbitrary structures via $\ell_{1,2}$-norm," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 1655–1663, Curran Associates, Inc., 2014.

[114] H. Zou, "The adaptive Lasso and its oracle properties," *Journal of the American Statistical Association*, vol. 101, no. 476, pp. 1418–1429, 2006.

[115] X. Gu, G. Yin, and J. J. Lee, "Bayesian two-step Lasso strategy for biomarker selection in personalized medicine development for time-to-event endpoints," *Contemporary Clinical Trials*, vol. 36, no. 2, 2013.

[116] N. Meinshausen, "Relaxed Lasso," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 374–393, 2007.

[117] M. Wainwright, "Sharp thresholds for high-dimensional and noisy sparsity recovery using ℓ_1-constrained quadratic programming (lasso)," *IEEE Transactions on Information Theory*, vol. 55, no. 5, pp. 2183–2202, 2009.

[118] P. Zhao and B. Yu, "On model selection consistency of Lasso," *Journal of Machine Learning Research*, vol. 7, pp. 2541–2563, 2006.

[119] P. Bühlmann and S. van de Geer, *Statistics for High-Dimensional Data*. Springer, 2011.

[120] G. Raskutti, M. Wainwright, and B. Yu, "Minimax rates of estimation for high-dimensional linear regression over ℓ_q balls," *IEEE Transactions on Information Theory*, vol. 57, no. 10, pp. 6976–6994, 2011.

[121] J. Jia and B. Yu, "On model selection consistency of the elastic net when $p \gg n$," *Statistica Sinica*, vol. 20, no. 2, pp. 595–611, 2010.

[122] N. Meinshausen and P. Bühlmann, "Stability selection," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 72, no. 4, pp. 417–473, 2010.

[123] D. Freedman, "A remark on the difference between sampling with and without replacement," *Journal of the American Statistical Association*, vol. 72, pp. 681–681, 1977.

[124] P. Bühlmann and B. Yu, "Analyzing bagging," *Annals of Statistics*, vol. 30, pp. 9279–61, 2002.

[125] A. L.Blum and P. Langley, "Selection of relevant features and examples in machine learning," *Artificial Intelligence*, vol. 97, no. 2, pp. 245–271, 1997.

[126] T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, "Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, no. 5439, pp. 531–537, 1999.

[127] P. Pavlidis, J. Weston, J. Cai, and W. N. Grundy, "Gene functional classification from heterogeneous data," in *Proceedings of the Fifth Annual Interna-*

tional Conference on Computational Biology, RECOMB'01, pp. 249–255, Association for Computing Machinery, 2001.

[128] J. Devore and R. Peck, *Statistics: The Exploration and Analysis of Data*. Duxbury Press, 3 ed., 1997.

[129] J. G. Thomas, J. M. Olson, S. J. Tapscott, and L. P. Zhao, "An efficient and robust statistical modeling approach to discover differentially expressed genes using genomic expression profiles," *Genome research*, vol. 11, no. 7, pp. 1227–1236, 2001.

[130] O. G. Troyanskaya, M. E. Garber, P. O. Brown, D. Botstein, and R. B. Altman, "Nonparametric methods for identifying differentially expressed genes in microarray data," *Bioinformatics*, vol. 18, no. 11, pp. 1454–1461, 2002.

[131] D. Ghosh and A. M. Chinnaiyan, "Classification and selection of biomarkers in genomic data using LASSO," *Journal of Biomedicine and Biotechnology*, vol. 2005, no. 2, pp. 147–154, 2005.

[132] H. H. Zhang, J. Ahn, X. Lin, and C. Park, "Gene selection using support vector machines with non-convex penalty," *Bioinformatics*, vol. 22, no. 1, pp. 88–95, 2006.

[133] S. Zheng and W. Liu, "An experimental comparison of gene selection by Lasso and Dantzig selector for cancer classification," *Computers in Biology and Medicine*, vol. 41, no. 11, pp. 1033–1040, 2011.

[134] H. R. Frost and C. I. Amos, "Gene set selection via LASSO penalized regression," *Nucleic Acids Research*, vol. 45, no. 12, p. e114, 2017.

[135] J. C. Rieckmann, R. Geiger, D. Hornburg, T. Wolf, K. Kveler, D. Jarrossay, F. Sallusto, S. S. Shen-Orr, A. Lanzavecchia, M. Mann, and F. Meissner, "Social network architecture of human immune cells unveiled by quantitative proteomics," *Nature Immunology*, vol. 18, pp. 583–593, 2017.

[136] S. Avey, S. Mohanty, J. Wilson, H. Zapata, B. S. Samit R Joshi, S. Tsang, A. C. Shaw, and S. H. Kleinstein, "Multiple network-constrained regressions expand insights into influenza vaccination responses," *Bioinformatics*, vol. 33, no. 14, pp. 208–216, 2017.

[137] X.-S. He, T. H. Holmes, S. Sasaki, M. C. Jaimes, G. W. Kemble, C. L. Dekker, A. M. Arvin, and H. B. Greenberg, "Baseline levels of influenza-specific CD4 memory T-cells affect T-cell responses to influenza vaccines," *PLoS One*, vol. 3, no. 7, p. e2574, 2008.

[138] Z. Hu, D. A. Follmann, and K. Miura, "Vaccine design via nonnegative lasso-based variable selection," *Statistics in Medicine*, vol. 34, no. 10, pp. 1791–1798, 2015.

[139] L. Jacob, G. Obozinski, and J.-P. Vert, "Group Lasso with overlap and graph Lasso," in *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 433–440, 2009.

[140] G. Obozinski, L. Jacob, and J.-P. Vert, "Group Lasso with overlaps: the latent group Lasso approach," 2011.

[141] H. Park, A. Niida, S. Miyano, and S. Imoto, "Sparse overlapping group Lasso for integrative multi-omics analysis," *Journal of Computational Biology*, vol. 22, no. 2, pp. 73–84, 2015.

[142] L. Jacob, G. Obozinski, and J.-P. Vert, "Group Lasso with overlap and graph Lasso," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML'09, pp. 433–440, Association for Computing Machinery, 2009.

[143] Y. Wang, X. Li, and R. Ruiz, "Weighted general group Lasso for gene selection in cancer classification," *IEEE Transactions on Cybernetics*, vol. 49, no. 8, pp. 2860–2873, 2019.

[144] B. Zhang and S. Horvath, "A general framework for weighted gene co-expression network analysis," *Statistical Applications in Genetics and Molecular Biology*, vol. 4, no. 1, pp. 1–45, 2005.

[145] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic determination of genetic network architecture," *Nature Genetics*, vol. 22, no. 3, pp. 281–285, 1999.

[146] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo, "Validating clustering for gene expression data ," *Bioinformatics*, vol. 17, no. 4, pp. 309–318, 2001.

[147] S. Ma, X. Song, and J. Huang, "Supervised group Lasso with applications to microarray data analysis," *BMC Bioinformatics*, vol. 8, no. 1, p. 60, 2007.

[148] S. Kim and E. P. Xing, "Tree-guided group lasso for multi-response regression with structured sparsity, with an application to eQTL mapping," *The Annals of Applied Statistics*, vol. 6, no. 3, pp. 1095–1117, 2012.

[149] T. Chandrasekhar, K. Thangavel, and E. Elayaraja, "Effective clustering algorithms for gene expression data," *International Journal of Computer Applications*, vol. 32, no. 4, pp. 25–29, 2011.

[150] D. Furman and M. M. Davis, "New approaches to understanding the immune response to vaccination and infection," *Vaccine*, vol. 33, no. 40, pp. 5271–5281, 2015.

[151] D. Furman, V. Jojic, B. Kidd, S. Shen-Orr, J. Price, J. Jarrell, T. Tse, H. Huang, P. Lund, H. T. Maecker, P. J. Utz, C. L. Dekker, D. Koller, and M. M. Davis, "Apoptosis and other immune biomarkers predict influenza vaccine responsiveness," *Molecular Systems Biology*, vol. 9, no. 1, p. 659, 2013.

[152] C. C. Whiting, J. Siebert, A. M. Newman, H. wu Du, A. A. Alizadeh, J. Goronzy, C. M. Weyand, E. Krishnan, C. G. Fathman, and H. T. Maecker,

"Large-scale and comprehensive immune profiling and functional analysis of normal human aging," *PLoS One*, vol. 10, no. 7, 2015.

[153] P. Durek, K. Nordstrom, G. Gasparoni, A. Salhab, K. Kressler, M. de Almeida, K. Bassler, T. Ulas, F. Schmidt, J. Xiong, and J. K. Polansky, "Epigenomic profiling of human CD4+ T cells supports a linear differentiation model and highlights molecular regulators of memory development," *Immunity*, vol. 45, no. 5, pp. 1148–1161, 2016.

[154] A. M. Pesenacker, A. Y. Wang, A. Singh, J. Gillies, Y. Kim, C. A. Piccirillo, D. Nguyen, W. N. Haining, S. J. Tebbutt, C. Panagiotopoulos, and M. K. Levings, "A regulatory T-cell gene signature is a specific and sensitive biomarker to identify children with new-onset type 1 diabetes," *Immunology and Transplantation*, vol. 65, no. 4, pp. 1031–1039, 2016.

[155] D. Biswas, N. J. Birkbak, R. Rosenthal, C. T. Hiley, E. L. Lim, K. Papp, S. Boeing, M. Krzystanek, D. Djureinovic, L. La Fleur, M. Greco, B. Döme, J. Fillinger, H. Brunnström, Y. Wu, D. A. Moore, M. Skrzypski, C. Abbosh, K. Litchfield, M. Al Bakir, T. B. K. Watkins, S. Veeriah, G. A. Wilson, M. Jamal-Hanjani, J. Moldvay, J. Botling, A. M. Chinnaiyan, P. Micke, A. Hackshaw, J. Bartek, I. Csabai, Z. Szallasi, J. Herrero, N. McGranahan, C. Swanton, and T. Consortium, "A clonal expression biomarker associates with lung cancer mortality," *Nature Medicine*, vol. 25, no. 10, pp. 1540–1548, 2019.

[156] Y. Zhou, R. Jin, and S. C. Hoi, "Exclusive Lasso for multi-task feature selection," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (Y. W. Teh and M. Titterington, eds.), vol. 9 of *Proceedings of Machine Learning Research*, pp. 988–995, 2010.

[157] D. Kong, J. Liu, B. Liu, and X. Bao, "Uncorrelated group LASSO," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, vol. 30, 2016.

[158] F. Campbell and G. I. Allen, "Within group variable selection through the exclusive Lasso," *Electronic Journal of Statistics*, vol. 11, no. 2, pp. 4220–4257, 2017.

[159] Y. Huang and J. Liu, "Exclusive sparsity norm minimization with random groups via cone projection," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 12, pp. 6145–6153, 2018.

[160] D. Ming, C. Ding, and F. Nie, "A probabilistic derivation of LASSO and l12-norm feature selections," in *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, vol. 33, 2019.

[161] Y. Shi, M. Lei, R. Ma, and L. Niu, "Learning robust auto-encoders with regularizer for linearity and sparsity," *IEEE Access*, vol. 9, pp. 17195–17206, 2019.

[162] J. Peng, X. Zhu, Y. Wang, L. An, and D. Shen, "Structured sparsity regularized multiple kernel learning for Alzheimer's disease diagnosis," *Pattern Recognition*, vol. 88, pp. 370–382, 2019.

[163] S. Dey, P. Zhang, D. Sow, and K. Ng, "PerDREP: Personalized drug effectiveness prediction from longitudinal observational data," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1258–1268, 2019.

[164] H. Zhao, S. Ding, X. Li, and L. Zhao, "ℓ_p norm independently interpretable regularization based sparse coding for highly correlated data," *IEEE Access*, vol. 7, pp. 53542–53554, 2019.

[165] M. Huai, C. Miao, Q. Suo, Y. Li, J. Gao, and A. Zhang, "Uncorrelated patient similarity learning," in *Proceedings of the 2018 SIAM International Conference on Data Mining*, pp. 270–278, 2018.

[166] M. Takada, T. Suzuki, and H. Fujisawa, "Independently interpretable lasso: A new regularizer for sparse regression with uncorrelated variables," in *Pro-*

ceedings of the 21th International Conference on Artificial Intelligence and Statistics (A. Storkey and F. Perez-Cruz, eds.), vol. 84 of *Proceedings of Machine Learning Research*, pp. 454–463, PMLR, 2018.

[167] M. Moon and K. Nakai, "Stable feature selection based on the ensemble ℓ_1-norm support vector machine for biomarker discovery," *BMC Genomics*, vol. 17, p. 1026, 2016.

[168] J. Richiardi, A. Altmann, A.-C. Milazzo, C. Chang, M. M. Chakravarty, T. Banaschewski, G. J. Barker, A. L. Bokde, U. Bromberg, C. Büchel, P. Conrod, M. Fauth-Bühler, H. Flor, V. Frouin, J. Gallinat, H. Garavan, P. Gowland, A. Heinz, H. Lemaître, K. F. Mann, J.-L. Martinot, F. Nees, T. Paus, Z. Pausova, M. Rietschel, T. W. Robbins, M. N. Smolka, R. Spanagel, A. Ströhle, G. Schumann, M. Hawrylycz, J.-B. Poline, M. D. Greicius, and I. consortium, "Correlated gene expression supports synchronous activity in brain networks," *Science*, vol. 348, no. 6240, pp. 1241–1244, 2015.

[169] J. M. Rondina, J. Shawe-Taylor, and J. Mourão-Miranda, "A new feature selection method based on stability theory—exploring parameters space to evaluate classification accuracy in neuroimaging data," in *Machine Learning and Interpretation in Neuroimaging* (G. Langs, I. Rish, M. Grosse-Wentrup, and B. Murphy, eds.), pp. 51–59, Springer Berlin Heidelberg, 2012.

[170] J. M. Rondina, T. Hahn, L. de Oliveira, A. F. Marquand, T. Dresler, T. Leitner, A. J. Fallgatter, J. Shawe-Taylor, and J. Mourao-Miranda, "SCoRS—a method based on stability for feature selection and mapping in neuroimaging," *IEEE Transactions on Medical Imaging*, vol. 33, no. 1, pp. 85–98, 2014.

[171] J. Fan and J. Lv, "Sure independence screening for ultrahigh dimensional feature space," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 70, no. 5, pp. 849–911, 2008.

[172] C. Ding and H. Peng, "Minimum redundancy feature selection from microar-

ray gene expression data," *Journal of Bioinformatics and Computational Biology*, vol. 03, no. 02, pp. 185–205, 2005.

[173] Q. He and D.-Y. Lin, "A variable selection method for genome-wide association studies," *Bioinformatics*, vol. 27, no. 1, pp. 1–8, 2010.

[174] C. M. Yates, I. Filippis, L. A. Kelley, and M. J.E.Sternberg, "SuSPect: Enhanced prediction of single amino acid variant (SAV) phenotype using network features," *Journal of Molecular Biology*, vol. 426, no. 14, pp. 2692–2701, 2014.

[175] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins, "Text classification using string kernels," *Journal of Machine Learning Research*, vol. 2, pp. 419–444, 2002.

[176] C. Leslie, E. Eskin, and W. S. Noble, "The spectrum kernel: a string kernel for SVM protein classification," *Pacific Symposium on Biocomputing*, vol. 7, pp. 566–575, 2002.

[177] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004.

[178] T. S. Jaakkola and D. Haussler, "Exploiting generative models in discriminative classifiers," in *Advances in Neural Information Processing Systems 11* (M. S. Kearns, S. A. Solla, and D. A. Cohn, eds.), pp. 487–493, Bradford Books, MIT Press, 1999.

[179] C. Saunders, J. Shawe-Taylor, and A. Vinokourov, "String kernels, Fisher kernels and finite state automata," in *Advances in Neural Information Processing Systems 15* (S. Becker, S. Thrun, and K. Obermayer, eds.), pp. 649–656, MIT Press, 2003.

[180] C. Leslie, E. Eskin, J. Weston, and W. S. Noble, "Mismatch string kernels for SVM protein classification," in *Advances in Neural Information Processing Systems 15*, pp. 1441–1448, MIT Press, 2002.

[181] L. Wu, I. E.-H. Yen, S. Huo, L. Zhao, K. Xu, L. Ma, and S. J. C. Aggarwal, "Efficient global string kernel with random features: beyond counting substructures," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 520–528, 2019.

[182] R. Singh, J. Lanchantin, G. Robins, and Y. Qi, "Transfer string kernel for cross-context DNA-protein binding prediction," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2016.

[183] D. Chen and L. Jacob, "Recurrent kernel networks," *arXiv:1906.03200 [stat.ML]*, 2019.

[184] P. Yanardag and S. Vishwanathan, "Deep graph kernels," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1365–1374, 2015.

[185] C. H. Elzingaa and H. Wang, "Versatile string kernels," *Theoretical Computer Science*, vol. 495, pp. 50–65, 2013.

[186] B. Schölkopf, K. Tsuda, and J.-P. Vert, eds., *Kernel Methods in Computational Biology*. MIT Press, 2004.

[187] T. S. Jaakkola, M. Diekhans, and D. Haussler, "Using the Fisher kernel method to detect remote protein homologies," in *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology* (T. Lengauer, R. Schneider, P. Bork, D. L. Brutlag, J. I. Glasgow, H.-W. Mewes, and R. Zimmer, eds.), pp. 149–158, AAAI Press, 1999.

[188] T. Jaakkola, M. Diekhans, and D. Haussler, "A discriminative framework for detecting remote protein homologies," *Journal of Computational Biology*, vol. 7, no. 1–2, 2000.

[189] K. J. Won, C. Saunders, and A. Prügel-Bennett, "Evolving Fisher kernels for biological sequence classification," *Evolutionary Computation*, vol. 21, no. 1, pp. 83–105, 2013.

[190] G. Csurka and F. Perronnin, "Fisher vectors: Beyond bag-of-visual-words image representations," in *Computer Vision, Imaging and Computer Graphics. Theory and Applications* (P. Richard and J. Braz, eds.), vol. 229, pp. 28–42, Springer, 2010.

[191] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional SIFT descriptor and its application to action recognition," in *Proceedings of the 15th ACM International Conference on Multimedia*, pp. 357–360, ACM, 2007.

[192] C. Paganelli, M. Peroni, F. Pennati, G. Baroni, P. Summers, M. Bellomi, and M. Riboldi, "Scale invariant feature transform as feature tracking method in 4D imaging: a feasibility study," in *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 6543–6546, 2012.

[193] J. E.W.Koh, E. Y.K.Ng, S. V.Bhandary, Y. Hagiwara, A. Laude, and U. R. Acharya, "Automated retinal health diagnosis using pyramid histogram of visual words and Fisher vector techniques," *Computers in Biology and Medicine*, vol. 92, pp. 204–209, 2018.

[194] K. Tsuda, T. Kin, and K. Asai, "Marginalized kernels for biological sequences," *Bioinformatics*, vol. 18, pp. 268–275, 2002.

[195] N. Thomas, J. Heather, W. Ndifon, J. Shawe-Taylor, and B. Chain, "Decombinator: a tool for fast, efficient gene assignment in T-cell receptor sequences using a finite state machine," *Bioinformatics*, vol. 29, no. 5, pp. 542–50, 2013.

[196] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.

[197] N. Japkowicz and M. Shah, *Evaluating Learning Algorithms: A Classification Perspective*. Cambridge University Press, 2011.

[198] C. Leng, Y. Lin, and G. Wahba, "A note on the LASSO and related procedures in model selection," *Statistica Sinica*, vol. 16, pp. 1273–1284, 2006.

[199] J. Zhang, X. J. Jeng, and H. Liu, "Some two-step procedures for variable selection in high-dimensional linear regression," *arXiv:0810.1644 [math.ST]*, 2008.

[200] Q. Li, K. Fisher, W. Meng, B. Fang, E. Welsh, E. B. Haura, J. M. Koomen, S. A. Eschrich, B. L. Fridley, and Y. A. Chen, "GMSimpute: a generalized two-step Lasso approach to impute missing values in label-free mass spectrum analysis," *Bioinformatics*, 2019.

[201] M. Cinelli, Y. Sun, K. Best, J. M. Heather, S. Reich-Zeliger, E. Shifrut, N. Friedman, J. Shawe-Taylor, and B. Chain, "Feature selection using a one dimensional naïve Bayes' classifier increases the accuracy of support vector machine classification of CDR3 repertoires," *Bioinformatics*, vol. 33, no. 7, pp. 951–955, 2017.

[202] Y. Sun, K. Best, M. Cinelli, J. M. Heather, S. Reich-Zeliger, E. Shifrut, N. Friedman, J. Shawe-Taylor, and B. Chain, "Specificity, privacy, and degeneracy in the CD4 T cell receptor repertoire following immunization," *Frontiers in Immunology*, vol. 8, 2017.

[203] N. Meinshausen and B. Yu, "Lasso-type recovery of sparse representations for high-dimensional data," *Annals of Statistics*, vol. 37, no. 1, pp. 246–270, 2009.

[204] X. Zhang, Y. Wu, L. Wang, and R. Li, "Variable selection for support vector machines in moderately high dimensions," *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, vol. 78, no. 1, pp. 53–76, 2016.

[205] V. Gómez-Verdejo, E. Parrado-Hernández, J. Tohka, and A. D. N. Initiative, "Sign-consistency based variable importance for machine learning in brain imaging," *Neuroinformatics*, vol. 17, no. 1, pp. 1–17, 2019.

[206] M. Andersen, J. Dahl, and L. Vandenberghe, "CVXOPT: Python software for convex optimization." `https://cvxopt.org/`, 2004.

[207] S. Diamond and S. Boyd, "CVXPY: a Python-embedded modeling language for convex optimization," *Journal of Machine Learning Research*, vol. 17, no. 83, pp. 1–5, 2016.

[208] A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd, "A rewriting system for convex optimization problems," *Journal of Control and Decision*, vol. 5, no. 1, pp. 42–60, 2018.

[209] CVX Research Inc., "CVX: Matlab software for disciplined convex programming, version 2.0." `http://cvxr.com/cvx`, Aug. 2012.

[210] M. Grant and S. Boyd, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control* (V. Blondel, S. Boyd, and H. Kimura, eds.), Lecture Notes in Control and Information Sciences, pp. 95–110, Springer-Verlag Limited, 2008. `http://stanford.edu/~boyd/graph_dcp.html`.

[211] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine Learning in Python ," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[212] P. J. Hanley and C. M. Bollard, "Controlling cytomegalovirus: Helping the immune system take the lead," *Viruses*, vol. 6, no. 6, pp. 2242–2258, 2014.

[213] L. J. Zhang, P. Hanff, C. Rutherford, W. H. Churchill, and C. S. Crumpacker, "Detection of human cytomegalovirus DNA, RNA, and antibody in normal donor blood," *Journal of Infectious Diseases*, vol. 171, no. 4, pp. 1002–1006, 1995.

[214] R. Emerson, W. DeWitt, M. Vignali, J. Gravley, J. Hu, E. Osborne, C. Desmarais, M. Klinger, C. Carlson, J. Hansen, M. Rieder, and H. Robins, "Immunosequencing identifies signatures of cytomegalovirus exposure history and HLA-mediated effects on the T cell repertoire," *Nature Genetics*, vol. 49, no. 5, pp. 659–665, 2017.

[215] R. A. Fisher, "On the interpretation of χ^2 from contingency tables, and the calculation of p," *Journal of the Royal Statistical Society*, vol. 85, no. 1, pp. 87–94, 1922.

[216] N. D. Neuter, E. Bartholomeus, G. Elias, N. Keersmaekers, A. Suls, H. Jansens, E. Smits, N. Hens, P. Beutels, P. V. Damme, G. Mortier, V. V. Tendeloo, K. Laukens, P. Meysman, and B. Ogunjimi, "Memory CD4+ T cell receptor repertoire data mining as a tool for identifying cytomegalovirus serostatus," *Genes & Immunity*, vol. 20, no. 3, pp. 255–260, 2019.

[217] X. Liang, L. U. Weigand, I. G. Schuster, E. Eppinger, J. C. van der Griendt, A. Schub, M. Leisegang, D. Sommermeyer, F. Anderl, Y. Han, J. Ellwart, A. Moosmann, D. H. Busch, W. Uckert, C. Peschel, and A. M. Krackhardt, "A single TCRα-chain with dominant peptide recognition in the allorestricted HER2/neu-specific T cell repertoire," *Journal of Immunology*, vol. 184, no. 3, pp. 1617–1629, 2010.

[218] K. Peggs, S. Verfuerth, A. Pizzey, J. Ainsworth, P. Moss, and S. Mackinnon, "Characterization of human cytomegalovirus peptide–specific CD8+ T-cell repertoire diversity following in vitro restimulation by antigen-pulsed dendritic cells," *Blood*, vol. 99, no. 1, pp. 213–223, 2002.

[219] N. Babel, G. Brestrich, L. P. Gondek, A. Sattler, M. W. Wlodarski, N. Poliak, N. Bethke, A. Thiel, M. H. Hammer, P. Reinke, and J. P. Maciejewski, "Clonotype analysis of cytomegalovirus-specific cytotoxic T lymphocytes," *Journal of the American Society of Nephrology*, vol. 20, no. 2, pp. 344–352, 2009.

[220] L. Janbazian, D. A. Price, G. Canderan, A. Filali-Mouhim, T. E. Asher, D. R. Ambrozak, P. Scheinberg, M. R. Boulassel, J.-P. Routy, R. A. Koup, D. C. Douek, R.-P. Sekaly, and L. Trautmann, "Clonotype and repertoire changes drive the functional improvement of HIV-specific CD8 T cell populations under conditions of limited antigenic stimulation," *Journal of Immunology*, vol. 188, no. 3, pp. 1156–1167, 2012.

[221] E. M. Iancu, P. Corthesy, P. Baumgaertner, E. Devevre, V. Voelter, P. Romero, D. E. Speiser, and N. Rufer, "Clonotype selection and composition of human CD8 T cells specific for persistent herpes viruses varies with differentiation but is stable over time," *Journal of Immunology*, vol. 183, no. 1, pp. 319–331, 2009.

[222] A. Schub, I. G. Schuster, W. Hammerschmidt, and A. Moosmann, "CMV-specific TCR-transgenic T cells for immunotherapy," *Journal of Immunology*, vol. 183, no. 10, pp. 6819–6830, 2009.

[223] M. Klinger, K. Kong, M. Moorhead, L. Weng, J. Zheng, and M. Faham, "Combining Next-Generation Sequencing and Immune Assays: A Novel Method for Identification of Antigen-Specific T Cells," *PLoS One*, vol. 8, sep 2013.

[224] N. Khan, N. Shariff, M. Cobbold, R. Bruton, J. A. Ainsworth, A. J. Sinclair, L. Nayak, and P. A. H. Moss, "Cytomegalovirus seropositivity drives the CD8 T cell repertoire toward greater clonality in healthy elderly individuals," *Journal of Immunology*, vol. 169, no. 4, pp. 1984–1992, 2002.

[225] P. L. Klarenbeek, E. B. M. Remmerswaal, I. J. M. ten Berge, M. E. Doorenspleet, B. D. C. van Schaik, R. E. E. Esveldt, S. D. Koch, A. ten Brinke, A. H. C. van Kampen, F. J. Bemelman, P. P. Tak, F. Baas, N. de Vries, and R. A. W. van Lier, "Deep sequencing of antiviral T-cell responses to HCMV and EBV in humans reveals a stable repertoire that is maintained for many years," *PLoS Pathogens*, vol. 8, no. 9, 2012.

[226] M. H. M. Heemskerk, R. S. Hagedoorn, M. A. W. G. van der Hoorn, L. T. van der Veken, M. Hoogeboom, M. G. D. Kester, R. Willemze, and J. H. F. Falkenburg, "Efficiency of T-cell receptor expression in dual-specific T cells is controlled by the intrinsic qualities of the TCR chains within the TCR-CD3 complex," *Blood*, vol. 109, no. 1, pp. 235–243, 2007.

[227] C. Retière, V. Prod'homme, B.-M. Imbert-Marcille, M. Bonneville, H. Vié, and M.-M. Hallet, "Generation of cytomegalovirus-specific human T-lymphocyte clones by using autologous B-lymphoblastoid cells with stable expression of pp65 or IE1 proteins: a tool to study the fine specificity of the antiviral response," *Journal of Virology*, vol. 74, no. 9, pp. 3948–3952, 2000.

[228] K. K. Wynn, Z. Fulton, L. Cooper, S. L. Silins, S. Gras, J. K. Archbold, F. E. Tynan, J. J. Miles, J. McCluskey, S. R. Burrows, J. Rossjohn, and R. Khanna, "Impact of clonal competition for peptide-MHC complexes on the CD8+ T-cell repertoire selection in a persistent viral infection," *Blood*, vol. 111, no. 8, pp. 4283–4292, 2008.

[229] I. Miconnet, A. Marrau, A. Farina, P. Taffé, S. Vigano, A. Harari, and G. Pantaleo, "Large TCR diversity of virus-specific CD8 T cells provides the mechanistic basis for massive TCR renewal after antigen exposure," *Journal of Immunology*, vol. 186, no. 12, pp. 7039–7049, 2011.

[230] M. P. Weekes, M. R. Wills, J. G. P. Sissons, and A. J. Carmichael, "Long-term stable expanded human CD4+ T cell clones specific for human cytomegalovirus are distributed in both CD45RAhigh and CD45ROhigh populations," *Journal of Immunology*, vol. 173, no. 9, pp. 5843–5851, 2004.

[231] D. J. van Bockel, D. A. Price, M. L. Munier, V. Venturi, T. E. Asher, K. Ladell, H. Y. Greenaway, J. Zaunders, D. C. Douek, D. A. Cooper, M. P. Davenport, and A. D. Kelleher, "Persistent survival of prevalent clonotypes within an immunodominant HIV gag-specific CD8 + T cell response," *Journal of Immunology*, vol. 186, pp. 359–371, jan 2011.

[232] D. Koning, A. I. Costa, R. Hasrat, B. P. X. Grady, S. Spijkers, N. Nanlohy, C. Keşmir, and D. Van Baarle, "In vitro expansion of antigen-specific CD8+ T cells distorts the T-cell repertoire," *Journal of Immunological Methods*, vol. 405, pp. 199–203, 2014.

[233] Y. Hamel, P. Rohrlich, V. Baron, D. Bonhomme, F. Rieux-Laucat, A. Necker, F. Lemonnier, L. Ferradini, A. Fischer, and M. Cavazzana-Calvo, "Characterization of antigen-specific repertoire diversity following in vitro restimulation by a recombinant adenovirus expressing human cytomegalovirus pp65," *European Journal of Immunology*, vol. 33, no. 3, pp. 760–768, 2003.

[234] S. Schwele, A. M. Fischer, G. Brestrich, M. W. Wlodarski, L. Wagner, M. Schmueck, A. Roemhild, S. Thomas, M. H. Hammer, N. Babel, A. Kurtz, J. P. Maciejewski, P. Reinke, and H.-D. Volk, "Cytomegalovirus-specific regulatory and effector T cells share TCR clonality—possible relation to repetitive CMV infections," *American Journal of Transplantation*, vol. 12, no. 3, pp. 669–681, 2012.

[235] M. Dziubianau, J. Hecht, L. Kuchenbecker, A. Sattler, U. Stervbo, C. Rodelsperger, P. Nickel, A. U. Neumann, P. N. Robinson, S. Mundlos, H.-D. Volk, A. Thiel, P. Reinke, and N. Babel, "TCR repertoire analysis by next generation sequencing allows complex differential diagnosis of T cell-related pathology," *American Journal of Transplantation*, vol. 13, no. 11, pp. 2842–2854, 2013.

[236] A. Arakaki, K. Ooya, Y. Akiyama, M. Hosokawa, M. Komiyama, A. Iizuka, K. Yamaguchi, and T. Matsunaga, "TCR-β repertoire analysis of antigen-specific single T cells using a high-density microcavity array," *Biotechnology and Bioengineering*, vol. 106, no. 2, pp. 311–318, 2010.

[237] S. Giest, A. McWhinnie, M.-P. Lefranc, A.-M. Little, S. Grace, S. Mackinnon, J. A. Madrigal, and P. J. Travers, "Cytomegalovirus-specific CD8+

T cells targeting different peptide/HLA combinations demonstrate varying T-cell receptor diversity," *Immunology*, vol. 135, no. 1, pp. 27–39, 2012.

[238] P. J. Hanley, J. J. Melenhorst, S. Nikiforow, P. Scheinberg, J. W. Blaney, G. Demmler-Harrison, C. R. Cruz, S. Lam, R. A. Krance, K. S. Leung, C. A. Martinez, H. Liu, D. C. Douek, H. E. Heslop, C. M. Rooney, E. J. Shpall, A. J. Barrett, J. R. Rodgers, and C. M. Bollard, "CMV-specific T cells generated from naïve T cells recognize atypical epitopes and may be protective in vivo," *Science translational medicine*, vol. 7, p. 285ra63, apr 2015.

[239] R. M. Brennan, J. J. Miles, S. L. Silins, M. J. Bell, J. M. Burrows, and S. R. Burrows, "Predictable $\alpha\beta$ T-cell receptor selection toward an HLA-B*3501-restricted human cytomegalovirus epitope," *Journal of Virology*, vol. 81, no. 13, pp. 7269–7273, 2007.

[240] E. K. Day, A. J. Carmichael, I. J. M. ten Berge, E. C. P. Waller, J. G. P. Sissons, and M. R. Wills, "Rapid CD8+ T cell repertoire focusing and selection of high-affinity clones into memory following primary infection with a persistent human virus: human cytomegalovirus," *Journal of Immunology*, vol. 179, no. 5, pp. 3203–3213, 2007.

[241] T. H. O. Nguyen, L. C. Rowntree, D. G. Pellicci, N. L. Bird, A. Handel, L. Kjer-Nielsen, K. Kedzierska, T. C. Kotsimbos, and N. A. Mifsud, "Recognition of distinct cross-reactive virus-specific CD8+ T cells reveals a unique TCR signature in a clinical setting," *Journal of Immunology*, vol. 192, no. 11, pp. 5039–5049, 2014.

[242] L. Trautmann, M. Rimbert, K. Echasserieau, X. Saulquin, B. Neveu, J. Dechanet, V. Cerundolo, and M. Bonneville, "Selection of T cell clones expressing high-affinity public TCRs within human cytomegalovirus-specific CD8 T cell responses," *Journal of Immunology*, vol. 175, no. 9, pp. 6123–6132, 2005.

[243] G. C. Wang, P. Dash, J. A. McCullers, P. C. Doherty, and P. G. Thomas,

"T cell receptor $\alpha\beta$ diversity inversely correlates with pathogen-specific antibody levels in human cytomegalovirus infection," *Science Translational Medicine*, vol. 4, no. 128, p. 128ra42, 2012.

[244] V. Venturi, H. Y. Chin, T. E. Asher, K. Ladell, P. Scheinberg, E. Bornstein, D. van Bockel, A. D. Kelleher, D. C. Douek, D. A. Price, and M. P. Davenport, "TCR β-chain sharing in human CD8+ T cell responses to cytomegalovirus and EBV," *Journal of Immunology*, vol. 181, no. 11, pp. 7853–7862, 2008.

[245] R. M. Brennan, J. Petersen, M. A. Neller, J. J. Miles, J. M. Burrows, C. Smith, J. McCluskey, R. Khanna, J. Rossjohn, and S. R. Burrows, "The impact of a large and frequent deletion in the human TCR β locus on antiviral immunity," *Journal of Immunology*, vol. 188, no. 6, pp. 2742–2748, 2012.

[246] P. Scheinberg, J. J. Melenhorst, J. M. Brenchley, B. J. Hill, N. F. Hensel, P. K. Chattopadhyay, M. Roederer, L. J. Picker, D. A. Price, A. J. Barrett, and D. C. Douek, "The transfer of adaptive immunity to CMV during hematopoietic stem cell transplantation is dependent on the specificity and phenotype of CMV-specific T cells in the donor," *Blood*, vol. 114, no. 24, pp. 5071–5080, 2009.

[247] H. Nakasone, Y. Tanaka, R. Yamazaki, K. Terasako, M. Sato, K. Sakamoto, R. Yamasaki, H. Wada, Y. Ishihara, K. Kawamura, T. Machishima, M. Ashizawa, S. i Kimura, M. Kikuchi, A. Tanihara, J. Kanda, S. Kako, J. Nishida, and Y. Kanda, "Single-cell T-cell receptor-β analysis of HLA-A*2402-restricted CMV- pp65-specific cytotoxic T-cells in allogeneic hematopoietic SCT," *Bone Marrow Transplantation*, vol. 49, pp. 87–94, 2014.

[248] L. Sompayrac, *How the Immune System Works*. Wiley-Blackwell, 5 ed., 2015.

[249] Y. Delneste, C. Beauvillain, and P. Jeannin, "Immunité naturelle: Structure

et fonction des toll-like receptors (Innate immunity: structure and function of TLRs)," *Medecine Sciences (Paris)*, vol. 23, no. 1, pp. 67–74, 2007.

[250] B. Beutler, Z. Jiang, P. Georgel, K. Crozat, B. Croker, S. Rutschmann, X. Du, and K. Hoebe, "Genetic analysis of host resistance: Toll-like receptor signaling and immunity at large," *Annual Review of Immunology*, vol. 24, pp. 353–389, 2006.

[251] T. Doan, R. Melvold, S. Viselli, and C. Waltenbaugh, *Lippincott's Illustrated Reviews: Immunology*. Lippincott Williams & Wilkins, 1 ed., 2007.

[252] J. Punt, S. Stranford, P. Jones, and J. Owen, *Kuby Immunology*. W. H. Freeman, 8 ed., 2018.

[253] W. Commons, "File:hematopoiesis (human) diagram.png — wikimedia commons, the free media repository," 2017. [Online; accessed 5-August-2019].

[254] J. P. Robinson and G. F. Babcock, eds., *Phagocyte Function: A Guide for Research and Clinical Evaluation*. Wiley-Blackwell, 1998.

[255] Y. Chen and X. Zhang, "Pivotal regulators of tissue homeostasis and cancer: macrophages," *Experimental Hematology & Oncology*, vol. 6, no. 23, 2017.

[256] M. A. Velasco-Velázquez, D. Barrera, A. González-Arenas, C. Rosales, and J. Agramonte-Hevia, "Macrophage–Mycobacterium tuberculosis interactions: role of complement receptor 3," *Microbial Pathogenesis*, vol. 35, no. 3, pp. 125–131, 2003.

[257] V. Matzaraki, V. Kumar, C. Wijmenga, and A. Zhernakova, "The MHC locus and genetic susceptibility to autoimmune and infectious diseases," *Genome Biology*, vol. 18, no. 76, 2017.

[258] J. Bain, N. A. R. Gow, and L.-P. Erwig, "Novel insights into host-fungal pathogen interactions derived from live-cell imaging," *Seminars in Immunopathology*, vol. 37, no. 2, pp. 131–139, 2015.

[259] W. Commons, "File:phagocytosis2.png — wikimedia commons, the free media repository," 2019. [Online; accessed 05-August-2019].

[260] C. Alexander and E. T. Rietschel, "Bacterial lipopolysaccharides and innate immunity," *Journal of Endotoxin Research*, vol. 7, no. 3, pp. 167–202, 2001.

[261] V. Apostolopoulos and I. F. Mckenzie, "Role of the mannose receptor in the immune response," *Current Molecular Medicine*, vol. 1, no. 4, pp. 469–474, 2001.

[262] P. J. Delves, S. J. Martin, and D. R. B. and, *Roitt's Essential Immunology*. Wiley-Blackwell, 13 ed., 2017.

[263] M. Haniffa, M. Gunawan, and L. Jardinea, "Human skin dendritic cells in health and disease," *Journal of Dermatological Science*, vol. 77, no. 2, pp. 85–92, 2015.

[264] A. J. Stagg, A. L. Hart, K. S. C, and M. A. Kamm, "The dendritic cell: its role in intestinal inflammation and relationship with gut bacteria," *Gut*, vol. 52, pp. 1522–1529, 2003.

[265] P. N. Fries and P. J. Griebel, "Mucosal dendritic cell diversity in the gastrointestinal tract," *Cell Tissue Research*, vol. 343, no. 1, pp. 33–41, 2011.

[266] C. Pilona, B. Levast, F. Meurens, Y. L. Vernd, D. Kerboeuf, H. Salmonc, F. Velge-Roussel, Y. Lebranchu, and C. Baron, "CD40 engagement strongly induces CD25 expression on porcine dendritic cells and polarizes the T cell immune response toward Th1," *Molecular Immunology*, vol. 46, no. 3, pp. 437–447, 2009.

[267] C. Tecchio, A. Micheletti, and M. A. Cassatella, "Neutrophil-derived cytokines: facts beyond expression," *Frontiers in Immunology*, vol. 5, no. 508, 2014.

[268] A. W. Segal, "How neutrophils kill microbes," *Annual Review of Immunology*, vol. 23, pp. 197–223, 2005.

[269] A. Stevens, J. S. Lowe, and B. Young, *Wheater's Basic Histopathology : A Color Atlas and Text*. Churchill Livingstone, 4 ed., 2002.

[270] T. G. Uhm, B. S. Kim, and I. Y. Chung, "Eosinophil development, regulation of eosinophil-specific genes, and role of eosinophils in the pathogenesis of asthma," *Allergy, Asthma & Immunology Research*, vol. 4, no. 2, pp. 68–79, 2012.

[271] F. H. Falcone, H. Haas, and B. F. Gibbs, "The human basophil: a new appreciation of its role in immune responses," *Blood*, vol. 96, pp. 4028–4038, 2000.

[272] J. V. Sarma and P. A. Ward, "The complement system," *Cell Tissue Research*, vol. 343, no. 1, pp. 227–235, 2011.

[273] M. Noris and G. Remuzzi, "Overview of complement activation and regulation," *Seminars in Nephrology*, vol. 33, no. 6, pp. 479–492, 2013.

[274] F. V.Suurs, M. N. de Hooge, E. G. de Vries, and D. J. A. de Groot, "A review of bispecific antibodies and antibody constructs in oncology and clinical challenges," *Pharmacology & Therapeutics*, vol. 201, pp. 103–119, 2019.

[275] R. Rouet, D. Lowe, and D. Christ, "Stability engineering of the human antibody repertoire," *FEBS Letters*, vol. 588, no. 2, pp. 269–277, 2015.

[276] G. Vidarsson, G. Dekkers, and T. Rispens, "IgG subclasses and allotypes: From structure to effector functions," *Frontiers in Immunology*, vol. 5, no. 520, 2014.