Novel bioinformatics tools for epitope-based peptide vaccine design

Thomas Brian Whalley

May 2022

A thesis submitted to Cardiff University

in candidature for the degree of

Doctor of philosophy

ACKNOWLEDGEMENTS

Firstly, I would like to thank my co-supervisors Professor Andrew Sewell and Dr. Barbara Szomolay for welcoming me onto such diverse and interesting project. Thank you both for your continued patience, advice, and guidance throughout my PhD. Thank you also to Cardiff University Systems Immunity URI for providing the funding for this project.

I would also like to extend my thanks to my examiners Dr. Ivo Tews and Dr. Cedric Berger for their constructive and useful comments.

This work would not have been possible without the broader support of the T-cell modulation group. Thank you for so many interesting lab meetings, journal clubs and general conversations. Special thanks go to the various residents of 2F04 through the years. Thank you all for putting up with me moaning in front of a computer screen for so long.

Although none of the work has made it into this thesis, my thanks go out to Dr. Sascha Ott and Dr. Paul Brown of the Zeeman Insititue, Warwick University for their help and contributions to many projects and webtools.

In particular I would also like to thank Dr. Bruce MacLachlan for his help and guidance in chapter 4. Your contributions, both in ideas, motivation and code really pushed the STACEI project along. These are littered throughout STACEI but I am grateful in particular for the initial crossing angle, contact and visualisation scripts. My thanks must also go to Dr. Alex Greenshields-Watson who in particular was very willing to be a guinea pig in testing STACEI and trawling through the output files.

The HPC work in chapters 6, 7 and 8 were especially new to me at the start of the project. I would like to thank Tom Green and the staff at ARCCA for their advice on both writing CUDA code and on benchmarking it.

For chapters 9,10 and 11 I would like to extend my thanks to Professor Tom Connor. The guidance on pipelining and containerization saved me a lot of time and stress, as did all the helpful points from the microbiologist's perspective. My appreciation also goes out to the MRC CLIMB consortium for allowing me to access their HPC resources for this side of my PhD.

My final thanks are to my friends and family for rounding off the whole experience and making it so enjoyable through your constant support.

ABSTRACT

BACKGROUND

T-cells are essential in the mediation of immune responses, helping clear bacteria, viruses and cancerous cells. T-cells recognise anomalies in the cellular proteome associated with infection and neoplasms through the T-cell receptor (TCR). The most common TCRs in humans, αβ TCRs, engage processed peptide epitopes presented on the major histocompatibility complex (pMHC). TCR-pMHC interaction is critical to vaccination. In this thesis I will discuss three pieces of software and outcomes derived from them that contribute to epitope-based vaccine design.

RESULTS

Three pieces of software were developed to help scientists study and understand T-cell responses. The first, STACEI allows users to interrogate the TCR-pMHC crystal structures. The time consuming, error-prone analysis that previously would have to be ran manually, is replaced by a single, flexible package. The second development is the introduction of general-purpose computing on the GPU (GP-GPU) in aiding the prediction of T-cell epitopes by scanning protein datasets using data derived from combinatorial peptide libraries (CPLs). Finally, I introduce RECIPIENT, a reverse vaccinology tool (RV) that combines pangenomic and population genetics methods to predict good vaccine targets across multiple pathogen samples.

CONCLUSION

Across this thesis, I introduce three different methods that aid the study of T-cells that will hopefully improve future vaccine design. These methods range across data types and methodologies, with methods focusing on mechanistic understanding of the TCR-pMHC binding event; the application of GP-GPU to CPLs and using microbial genomics to aid the study and understanding of antigen-specific T-cell responses. These three methods have a significant potential for further integration, especially the structural methods.

LIST OF ABBREVIATIONS

A/E	Attaching and Effacing
ACE2	Angiotensin-converting enzyme 2
aEPEC	atypical enteropathogenic *Escherichia coli*
ALS	agonist likelihood score
AMP	antimicrobial peptide
APBS	adaptive Poisson-Boltzmann solver
APC	antigen presenting cell
API	application programming interface
ARCCA	advanced research computing @ Cardiff University
ASA	available surface area
BCR	B-cell receptor
BFP	bundle forming pilus
BSA	buried surface area
BTN3	butyrophilin 3
BWT	Burrows-Wheeler transform
C	C programming language
C++	C++ programming language
cccDNA	covalently closed circular DNA
CCR	C-C chemokine
CD4	cluster of differentiation 4
CD8	cluster of differentiation 8
CDR	complementarity determining region
CDS	coding sequence
Cif	cell-cycle inhibiting factor
CLIP	class II-associated invariant chain peptide
CNNs	convolutional neural network
COVID-19	Coronavirus disease 2019
CPL	combinatorial peptide library

CPU	central processing unit
CUDA	Compute Unified Device Architecture
DAG	directed acyclic graph
DC	dendritic cell
ds-RNA	double stranded RNA
EPEC	enteropathogenic *Escherichia coli*
ER	endoplasmic reticulum
ERGIC	ER-Golgi intermediate compartment
Esp	*Escherichia coli* secretory protein
Fab	fragment of antibody binding
FFT	fast Fourier transform
GAPDH	Glyceraldehyde 3-phosphate dehydrogenase
gDNA	genomic DNA
GFF	general feature format file
GISAID	global initiative on sharing avian influenza data
GO	gene ontology
GP-GPU	general purpose programming on the graphical processing unit
GP	Gaussian process
GPU	general processing unit
GUI	graphical user interface
HAX-1	HCLS1-associated protein X-1
HB	hydrogen bond
HbcAg	Hepatitis B c antigen
HbeAg	Hepatitis B e antigen
HBV	Hepatitis B virus
HCMV	human cytomegalovirus
HLA	human leukocyte antigen
HMM	hidden Markov model
HPC	high performance computing
IEDB	immune epitope database

IFN-γ	interferon gamma
Ig	immunoglobulin
LA	localised adherence
LEE	locus of enterocyte effacement
LPS	lipopolysaccharide
LT	heat labile
MAIT	mucosal associated invariant T-cell
MAPK	mitogen-activated protein kinase
MHC	major histocompatibility complex
MIP-1β	Macrophage inflammatory protein-1β
MLN	mesenteric lymph nodes
MLST	multilocus sequence typing
MPI	message passing interface
MR1	major histocompatibility complex, class I-related
MSA	multiple sequence alignment
NGS	next generation sequencing
NK	natural killer cell
Nle	non-LEE encoded effector
NOD	nucleotide-binding oligomerization domain
NSP1-16	non-structural protein 1-16
ORF	open reading frame
OVA	Ovalbumin
PAMP	pathogen-associated molecular patterns
PFRs	peptide flanking region
PICPL	primary identification of epitopes by combinatorial peptide library
pMHC	peptide:major histocompatibility complex
PP	Peyer's Patches
QC	quality control
RAG1/2	recombination activating gene 1 and 2
RBD	receptor binding domain

RCSB	Research Collaboratory for Structural Bioinformatics
RPS3	Ribosomal Protein S3
RSS	recognition signal sequences
RV	reverse vaccinology
SARS-CoV-2	severe acute respiratory syndrome coronavirus 2
SB	salt bridge
SC	shape/surface complementarity
SCS	single-cell sequencing
SPI	*Salmonella* pathogenicity island
ST	heat stabile
STACEI	Structural Tool for the Analysis of TCR pEptide MHC Interactions
T1F	type 1 fimbriae
T3SS	type 3 secretion system
TAA	tumour associated antigen
TAP	transporter associated with antigen processing
TCR	T-cell receptor
tEPEC	typical enteropathogenic *Escherichia coli*
Th	T helper cell
TNF-α	tumour necrosis factor alpha
Treg	regulatory T-cell
TSV	tab separated value
UMI	unique molecular identifiers
VCF	variant call file
vdW	van der Waals
Vi-TT	Vi antigen conjugate typhoid toxin
β2M	beta 2 microglobulin

LIST OF FIGURES

LIST OF TABLES

LIST OF SOFTWARE USED

ANARCI http://opig.stats.ox.ac.uk/webapps/newsabdab/sabpred/anarci/

Alakazam https://cran.rstudio.com/web/packages/alakazam/index.html

Bepipred-2 http://www.cbs.dtu.dk/services/BepiPred/

Biopython https://biopython.org

Biostructmat https://github.com/andrewguy/biostructmap

Circlize https://github.com/jokergoo/circlize

DIAMOND https://github.com/bbuchfink/diamond

dplyr https://dplyr.tidyverse.org/

FFT http://www.ccp4.ac.uk/download

ggplot2 https://ggplot2.tidyverse.org/

GNU C++ compiler https://gcc.gnu.org/

Kraken2 https://ccb.jhu.edu/software/kraken2/

Loctree3 https://rostlab.org/services/loctree3/

MASH https://github.com/marbl/Mash

NCONT http://www.ccp4.ac.uk/download

NetChop http://www.cbs.dtu.dk/services/NetChop/

netMHCpan http://www.cbs.dtu.dk/services/NetMHCpan/

netMHCIIpan http://www.cbs.dtu.dk/services/NetMHCIIpan/

NetworkD3 https://christophergandrud.github.io/networkD3/

NetworkX https://networkx.github.io/

Numpy https://numpy.org/

NVIDIA CUDA compiler https://docs.nvidia.com/cuda/cuda-compiler-driver-nvcc/index.html

Peptides https://cran.r-project.org/web/packages/Peptides/index.html

PISA http://www.ccp4.ac.uk/download

Prokka https://github.com/tseemann/prokka

PyMOL https://pymol.org/2/

Python 3.7 https://www.python.org/

R 3.5 https://www.r-project.org/

Roary https://github.com/sanger-pathogens/Roary

SC http://www.ccp4.ac.uk/download

Singularity https://singularity.lbl.gov/

Scipy https://www.scipy.org/

Swalign https://github.com/mbreese/swalign

TABLE OF CONTENTS

INTRODUCTION

1.1 THE ROLE AND ORIGINS OF THE IMMUNE SYSTEM

The immune system detects and eliminates threats posed to a host organism. This can include the presence of foreign bodies; for example, bacteria, viruses, fungi and parasites as well as aberrant self-cells which are cancerous. The immune system in higher order vertebrates can be roughly categorized into two divisions: the innate and adaptive immune systems.

1.2 INNATE IMMUNITY

Innate immunity consists of an evolutionarily old defence, designed to provide a wholesale "one size fits all" protection against pathogens which do not differ in response to changing threats. These protections include physical and chemical barriers such as the skin and antimicrobial peptides which limit pathogens' access to the body, as well as fast acting mechanisms of clearance, including molecular pathways such as complement and cellular component with cytotoxic, cytolytic and phagocytotic behaviours[1]. The innate immune system has a broad church of targets and can interact and modulate bacteria, viruses, fungi and parasites.

Naturally, these four pathogens vary vastly in terms of size and mode of interaction with the host. Viruses for example, can be range from 5-750nm in diameter. Viruses are obligate intracellular pathogens, meaning that they are unable to reproduce on their own, instead requiring the molecular machinery of their host to reproduce. Viruses can cause direct cell death by causing lysis. Bacteria are larger than viruses, have their own reproductive machinery and can engage with and destroy cells directly, or indirectly through the release of toxins. Similarly, many single celled parasites, for example members of the genus *Plasmodium* can kill cells. The largest of these parasites, helminths (parasitic worms) are too large to infect a host cell, but their presence in the body can lead to tissue damage through the formation of cysts[2].

The presence of the above pathogens is contrasted with the presence of commensal micro-organisms. Commensal micro-organisms include archaea, bacteria and fungi and cause no harm to the host and often contribute to a symbiotic relationship with the host.

A host has three main routes of dealing with pathogens: avoidance, resistance, and tolerance[3]. Avoidance refers to methods in which the host can avoid exposure to pathogens in the first place, usually through anatomical and physical barriers and behaviour. If this first barrier is succumbed, then resistance is the next step. Resistance refers to more direct methods of eliminating and reducing pathogens. To defend against these pathogens, resistance engages different effector mechanisms. Finally, tolerance involves adaptations that increases a tissue's capacity to resist damage caused by the pathogens.

1.2.1 Anatomical and chemical barriers

The foremost defence against pathogens are chemical and physical barriers. The most obvious examples of physical barriers are the skin and mucosal surfaces. At many anatomical barriers there are further resistance mechanisms that help solidify this defence such as antimicrobial peptides (AMPs) found on mucosal surfaces. AMPs have naturally occurring, broad-spectrum antimicrobial effects.

1.2.2 Inflammatory response

Once a pathogen has breached the host's first-line defence of anatomical and chemical barriers it will be met with cellular immune responses from the innate immune system. This engagement occurs when cells known as sensor cells interact with inflammatory inducers. Generally speaking, inflammatory inducers are molecules found uniquely on pathogens such as bacterial lipopolysaccharides (LPS) or molecules which are not normally found in the extracellular space, for example ATP.

The innate immune system detects pathogens by a variety of targets including lipopolysaccharides (LPS) and double stranded RNA (ds-RNA), collectively known as pathogen associated molecular patterns (PAMPs). PAMPs activate pathogen recognition receptors such as toll-like (TLR) and nucleotide binding oligomerisation domain-like receptors (NOD-like receptors). NOD-like receptors initiate inflammation which leads to a cascade of immune cell and coagulatory agents being recruited at the site of infection.

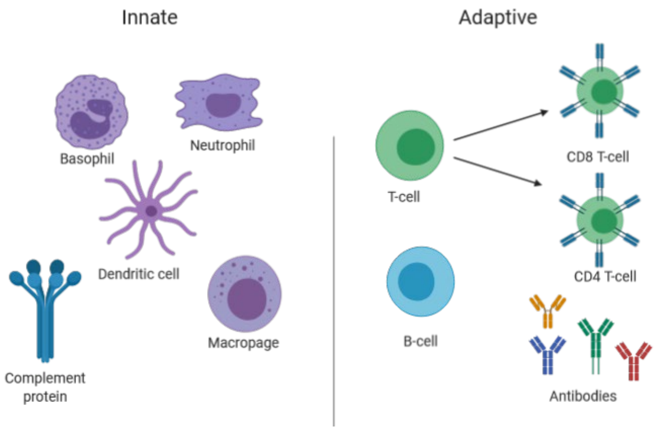


Figure 1 A simple representation of cells and proteins of the innate (left) and adaptive (right) immune systems.

The engagement of LPS and ds-RNA markers leads to further recruitment of more immune cells, including macrophages, neutrophils, dendritic cells (DCs) and basophils, which can engulf and destroy target cells and pathogens via phagocytosis. There are also more direct actions of cytotoxicity delivered by complement and natural killer (NK) cells. A number of these cells function in conjunction with the adaptive immune system by up-taking and presenting short peptide fragments of the pathogen, known as antigen on the cell surface. This is primarily performed by professional antigen presenting cells (APCs) which include DCs and macrophages.

1.3 Adaptive Immunity

The innate immune system is broadly acting and provides a good first response to any generalized target. However, this wide-ranging action is counterbalanced by its inability to launch a targeted or bespoke defence against a specific pathogen. Herein lies the role of the adaptive immune system; so called as it encompasses the ability to adapt during the life of an individual so as to remember a previously encountered pathogen.

The adaptive immune system is primarily comprised of T-cells and B-cells which detect antigens via the T-cell receptor (TCR) and B-cell receptor (BCR) respectively. The BCR also can act in its soluble form, which is known as an antibody. These receptors both carry and immense degree of sequence variation. This means that a given TCR or BCR can engage an antigen (which can consist of a protein, peptide, or small molecule) with a specificity unachievable by the innate immune system.

The TCR and BCR recognise protein epitopes (the site of the antigen responsibly for immune cell binding) in two distinct manners which give rise to different molecular behaviours.

1.4 B-CELLS AND ANTIBODIES

B-cells and antibodies interact with whole intact antigens. Antigens can be comprised of whole proteins (or lipoproteins), peptides and polysaccharides. Antigens are found either in free solution or at the cell surface. and are bound either linearly: where the epitope is one uninterrupted sequence of amino acids, or where the epitope forms a uniform sterically available site. Usually these epitopes are between 15 and 20 amino acids in length[4]. This is described in Figure 2. Human antibodies are heterodimers, consisting of a heavy and a light chain.

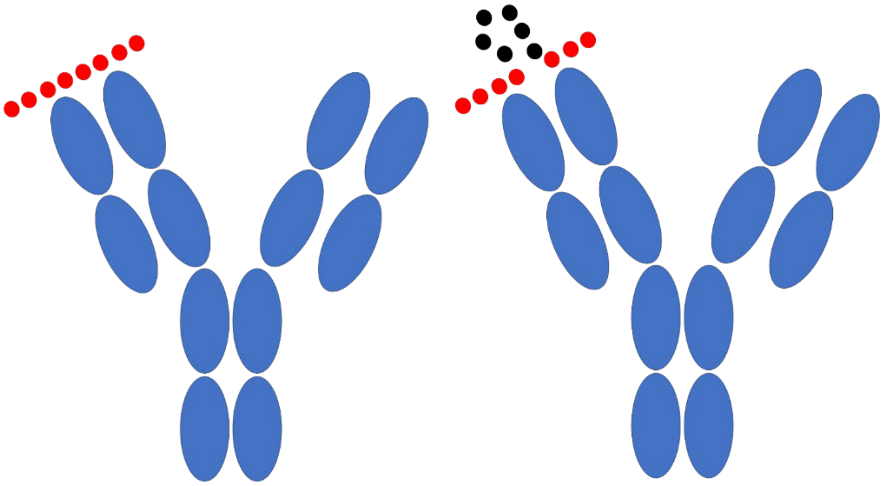


Figure 2 Example of a linear epitope (left) and a non-linear epitope (right). The red dots represent the epitope, whilst black dots are non-binding sites in the antigen.

The antigen binding region of antibodies varies vastly between different antibodies. This antigen binding region is known as the variable (V) region. This variability is what gives rise to the antibody's ability to bind with and engage a specific antigen. This V region is contrasted to by the constant (C) region which does not vary in the same way. There are five main forms of C region, known as isotypes which are used to engage different effector mechanisms. As it is membrane bound via the C region, the BCR does not have these effector functions. BCRs engage antigen via the V region, leading to B-cell activation which in turn leads to clonal expansion and antibody production[5].

Antibodies are unable to cross cell membranes so can only bind to extracellular of cell surface antigens; antibodies cannot bind to intracellular antigens. Intracellular protein antigens can be detected in processed form by the other antigen receptor – the T-cell receptor.

1.4.1 Antibody structure

Due to the fact that antibodies are soluble and are secreted in large quantities, they are easily obtained and studied. Therefore, most information on structure of BCRs and antibodies comes from the study of antibodies.

Antibodies are broadly “Y” shaped (Figure 3). The shape of the antibody allows to perform two distinct tasks, binding and engaging with antigens while also binding to effector molecules and cells that are recruited to destroy the antigen.

The ends of the two arms of the “Y”, the V regions are involved in antigen binding. As the V regions name suggests, its structure is varied. The stem or join of the “Y” confers the C region and as such is much more conserved. It is the C region that interacts with effector molecules and cells. Each “arm” of the antibody is formed from a heavy chain and a light chain, identical to those found on the other “arm”. In mammals, there are two categories of light chain, designated lambda (λ) and kappa (κ). Λ and κ chains give rise to physiochemical and structural differences in antibodies possessing the other chain[6].

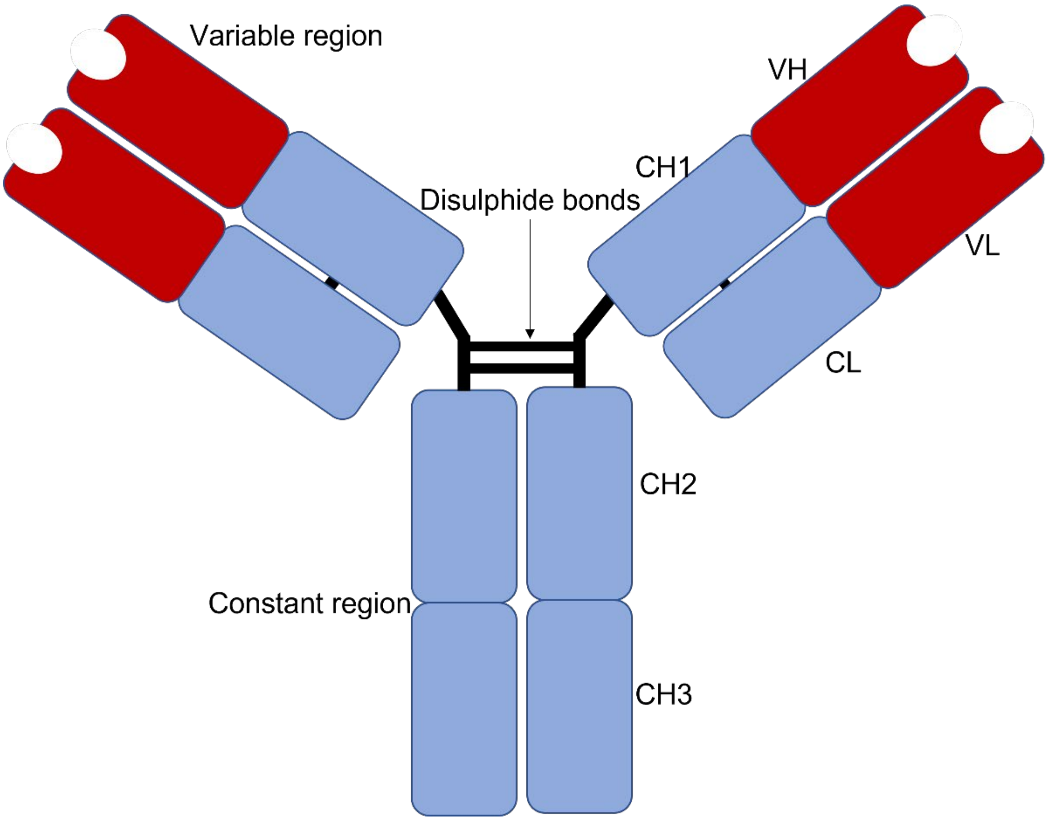


Figure 3 Cartoon representation of an antibody. The two antigen binding sites, the variable regions are found at the tips of the arms, joined to the others at the hinge region.

There are five different classes of immunoglobulins known as Immunoglobulins A, D, E, G and M (IgA, IgD, IgE, IgG and IgM). They are distinguished by having different C regions with their own inherent structure and properties. These classes, or isotypes, are defined by their heavy chain which determine their effector function. The function of these heavy chains is governed by the C-terminal region of the heavy chain; that is, the region of the heavy chain not associating with the light chain. Although different, the structural properties of the isotypes are similar in terms of antigen binding.

1.5 The T-cell receptor

The obvious failing of antibody immunity is that it cannot engage with intracellular antigens or antigens whose tertiary protein structure shifts dynamically. The T-cell is key in addressing this shortcoming. Like the antibody, it is also a heterodimer, consisting of two chains.

T-cells can be grouped into two main classes based on the genes used to make their TCR: the α and β TCR or γ and δ TCR. γδ T-cells possess a constituent γ chain and δ chain and represent a minority of the TCRs found in human blood, but account for up to half of the T-cells in the gut and skin. Although known antigens of γδ TCRs are scarce, they have been shown to interact with phosphorylated isoprenoid antigens in the context of molecules called butyrophilins[7], CD1 restricted lipids[8] and general stress ligands, for example those produced in human cytomegalovirus (HCMV) infection[9].

However, by far the most common TCR in human is the αβ TCR denoted by its α and β chains. Conventionally, the αβ TCR responds to linear sequences of amino acids (called peptides) presented at the cell surface in molecular cradles called major histocompatibility complex (MHC). MHC molecules can be divided into two different classes: MHC class I which generally presents peptides of 8-11 amino acids in length derived from intracellular proteins and MHC class II which can present longer peptides of 20+ amino acids derived from extracellular proteins. In humans the MHC is called human leukocyte antigen (HLA)[10]. The TCR recognises the combination of MHC and peptide (pMHC) by making specific molecular contacts with each species[11].

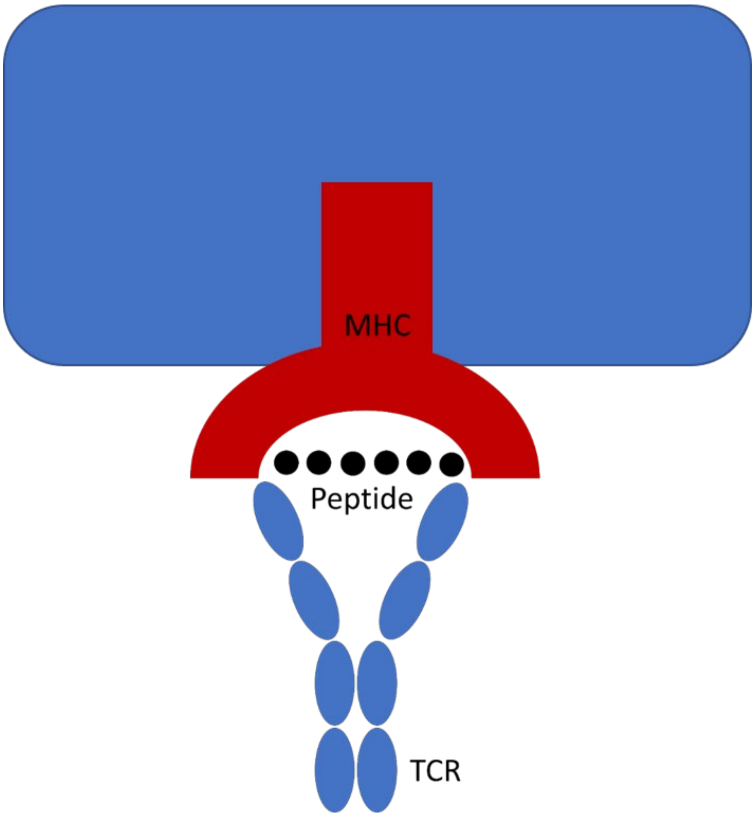


Figure 4 Cartoon schematic of the interaction of the TCR and the peptide MHC. Unlike antibodies which interact with a free antigen, in the classical case the TCR interacts with a short peptide bound to the MHC.

Between the TCR, the peptide and the MHC there is a diversity unsurpassed in molecular biology. It is estimated that the human body maintains approximately 10^{12} T-cells, of which 9% possess a distinct TCR[12]. This is much below the theoretical limit of possible human TCRs which is 10^{18} if comparable to predictions made in mice[13] as humans have considerably more TRBV genes.

Huge TCR diversity is complimented by massive germline variation in the HLA, with there being 18,691 HLA class I alleles and 7065 HLA class II at the time of writing (https://www.ebi.ac.uk/ipd/imgt/hla/stats.html)[14]. In evolutionary terms, many HLA loci are among the fastest evolving coding regions in the human genome[15].

1.5.1 V(D)J Recombination

The TCR can thank six hypervariable hairpin loops, known as complementarity determining regions (CDRs) for its incredible diversity[16]. Each TCR chain is coded for by a variable and constant domain, followed by a membrane-spanning region and cytosolic tail. It is the sequence of this variable region that give rise to the CDR loops.

The TCR variable and joining regions are encoded for by the V and J gene segments in the case of the TCRα and the V, D and J gene segments in the case of the TCRβ. During T-cell development the V, (D) and J gene segments are physically joined at recognition signal sequences (RSS) flanking the V(D) and J gene boundaries. This joining is performed by recombination activating gene 1 and 2 (RAG1/2).

Each RSS is composed of a conserved 9-mer and 7-mer nucleotide sequence separated by either 12 or 23 nucleotides, which roughly correspond two one or two turns of a DNA helix.

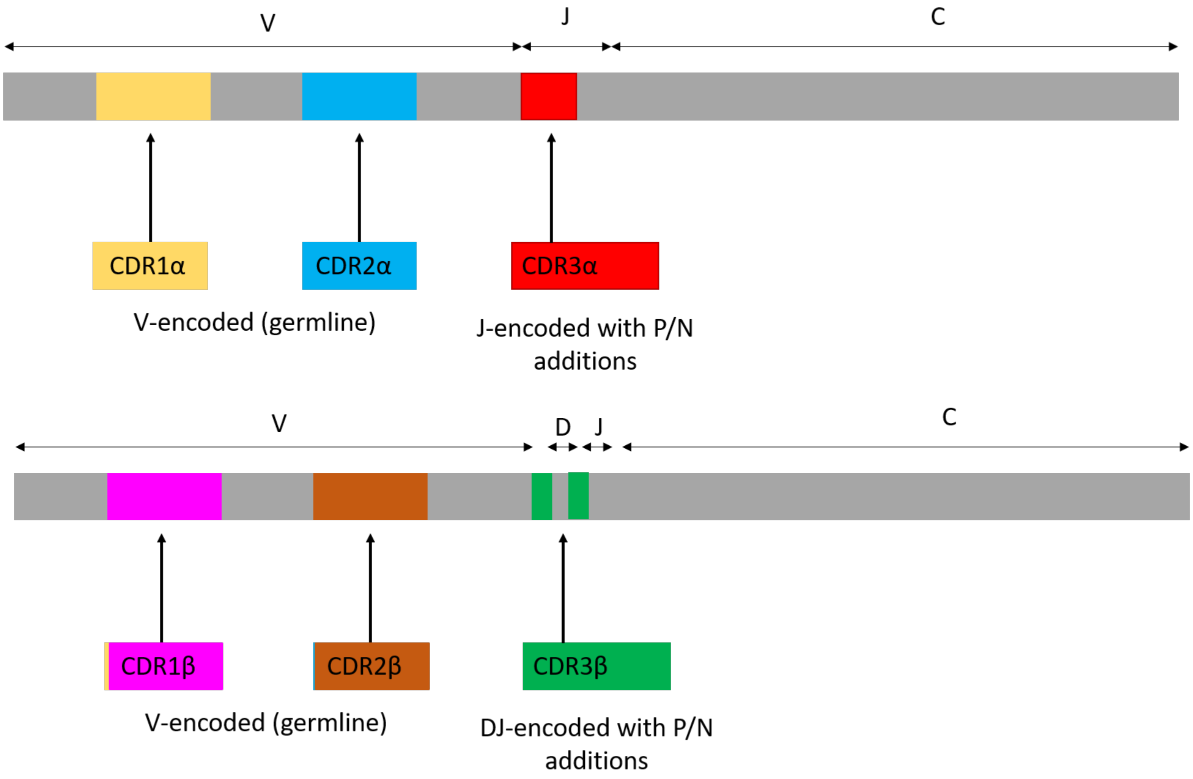


Figure 5 Cartoon of V(D)J recombination, for the α chain on the top and β chain on the bottom. The CDR1 and 2 are encoded by the germline V segments, whereas the CDR3 is encoded by the V(D) and J segments. Random additions and deletions of nucleotides gives further variation to the CDR3.

This gap, known as a spacer, is highly conserved in terms of length but not sequence. This spacer is essential for proper recombination spacing, known as the 12/23 rule.

The process is such that the CDR1 and CDR2 segments of both the α and β chains are coded entirely by the germline, but the CDR3 loops have palindromic (P) and non-template encoded

(N) nucleotides inserted at the end of the gene segments prior to the V(D) and J gene segments being paired and ligated[17].

1.5.2 The TCR-pMHC binding event

In spite of being an incredibly diverse molecule, many aspects of the TCR and indeed its mechanisms of recognition of the pMHC remain conserved. In the majority of cases, the TCR docks above the long axis of the pMHC binding cleft in a roughly diagonal binding orientation (Figure 6). Due to the orientation in the binding face of the TCR[11] the CDR1 and CDR2 loops are primarily engaged in binding the MHC.

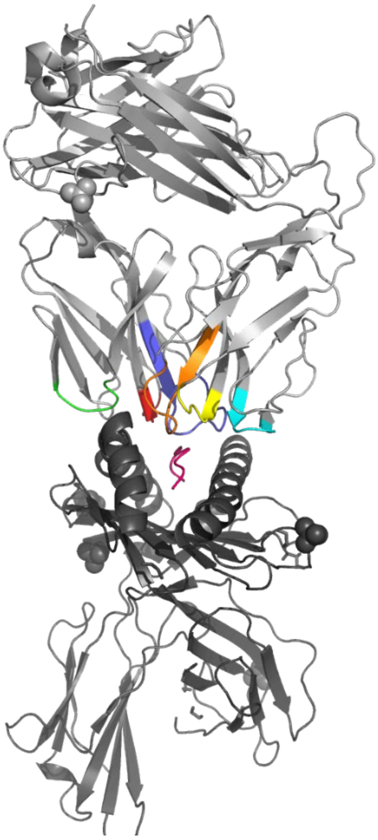


Figure 6 Cartoon representation of a TCR-pMHC structure. The TCRα (top left) and TCRβ (top right) engage the pMHC from above, along the long axis of the pMHC binding domain. The CDR loops are shown colourised as: CDR1α- red; CDR2α- green; CDR3α- blue; CDR1β- yellow; CDR2β- cyan and CDR3β- orange. The peptide is shown in magenta.

The CDR3, which sits almost directly above the peptide is the prime communicator with the peptide. This pattern of binding has been shown to be much less promiscuous than equivalent antibody antigen binding fragments (Fab) to pMHC structures. The reduced diversity of TCR binding mode has been suggested that this is because unlike antibodies, TCRs are expected to signal and to engage with the peptide in a specific manner[18].

Early structural studies showed that TCR CDR3 loops underwent conformational change upon interaction with the pMHC, suggesting that a TCR could flex and accommodate a variety of pMHCs. On a similar note, the surface complementarity (SC), a measure for “goodness of fit” between two complexes, in this case the TCR and pMHC was observed to be relatively low (0.41 to 0.64 in the first 5 structures published) suggesting that a TCR did not require particularly high binding affinity in order to activate[19].

A number of other studies have also shown that the pMHC can change its conformation in order to engage different TCRs[20,21]. Also, it has been observed that the same mutation to different TCR-pMHC does not necessarily have the same impact on peptide recognition[22].

1.5.3 TCR cross-reactivity

Unlike an antibody, the TCR never undergoes affinity maturation. This means that TCRs expressed by naïve T-cells must be required to engage peptides that it will never have encountered before, many of which are evolving dynamically at rates unachievable for the TCR itself. If TCRs could not bind to all possible pMHC, the existence of so-called T-cell “blind spots” would present a gap for rapidly evolving pathogens to exploit.

The need for individual TCRs to bind large numbers of different peptides becomes obvious when discussing the theoretical number of peptides there are for a given length. For example, the 20 proteogenic amino acids have the theoretical potential to combine to 20^{10} different 10-mer peptides. If only the top 1% are able to bind MHC that still leaves 12 x 10^{11} distinct peptides of this single length, an order of magnitude more than the total number of T-cells in the body[23].

It is unsurprising therefore that there is a growing literature evidencing TCR cross-reactivity, ranging from structural examples where the same TCR is bound to a different pMHC[24]; sequencing experiments combined with yeast expression libraries directly implicating >100 different epitopes to the same TCR[25] and mathematical modelling in conjunction with combinatorial peptide libraries (CPLs) predicting a single TCR can have on the order of 10^6 different peptide ligands[26]

1.5.4 Types of αβ T-cells

αβ T-cells make up the majority of human T-cells[27] and hence there is a bias in both the literature and methodologies surrounding this T-cell subtype. Usually αβ T-cells are categorised by the expression of a co-receptor molecule, namely cluster of differentiation 4 and 8 (CD4 and CD8, respectively).

CD8+ T-cells, also known as cytotoxic T-cells, are directly involved in the lysis of virally infected cells and cancer cells. CD8+ T-cells are also responsible for the elimination of intracellular bacteria and protozoa through the recognition of “foreign” peptides presented by MHC class I molecules at the cell surface. The majority of MHC class I-presented peptides are of 8-11 amino acids in length, but epitopes of up to 15 amino acids in length have been observed[28]. Most commonly MHC class I-restricted peptides are of length 9 or 10 amino acids[11].

CD4+ T-cells, or helper T-cells are involved in helping and regulating immune function. They recognise peptides presented by MHC class II molecules. Unlike MHC class I, these peptides are of much longer length, ranging between 12-20 amino acids as they can extend beyond the open-ended MHC peptide binding groove[29].

The TCR binding and signalling event (described in 1.5.2) is enhanced by the MHC molecule being engaged by the CD4 or CD8 co-receptor. The binding of CD4/8 occurs on the invariant regions of the MHC. CD4 Is a single chain protein composed of four Ig-like domains named D1-4 (Figure 7), where the first two domains are packed tightly together, followed by a hinge which joins to the next two domains. The MHC binding region of CD4 is found on D1. CD4 binds to a hydrophobic nook located between the α2 β2 domains of the MHC II molecule. This site is far from the peptide:MHC interface, allowing for the simultaneous engagement of the TCR. CD4 enhances sensitivity to the antigen up to 100x (meaning that 100-fold less antigen is required to activate the T-cell)[30].

CD8 is structured differently from CD4 (Figure 7) as it is present on the T-cell surface as a dimer of two α chains or as an αβ heterodimer. Naïve T-cells exclusively express CD8αβ[31] whereas CDαα homodimers also exist in activated effector and memory T-cells[32]. CD8αα

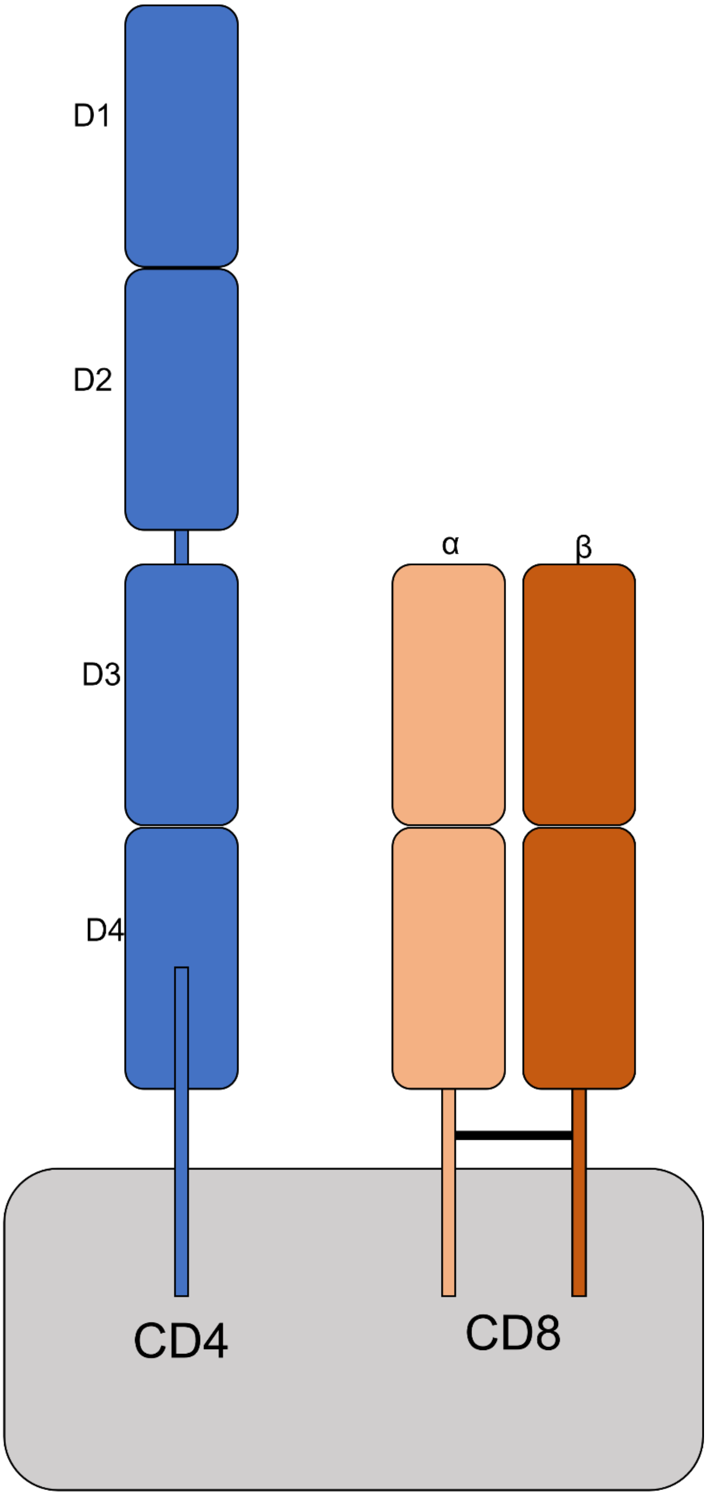


Figure 7 The CD4 and CD8 co-receptors. CD4 is a single linear molecular comprised of two hinged Ig-like units, whereas CD8 is comprised of two membrane bound Ig-like units.

homodimers also exist on unconventional T-cells such as mucosal invariant T-cells (MAITs). CDαβ binds to a conserved site on the α3 domain of the MHC class I molecule. The strength of this interaction is modulated by the glycosylation of the CD8 molecule. Like CD4 and MHC class II, the interaction between CD8 and MHC class I can happen simultaneously to the interaction between MHC and TCR.

1.6 The role of the MHC

1.6.1 MHC structure

MHC class I molecules consist of two chains, a larger variable membrane spanning heavy chain associated with a smaller conserved β2 microglobulin (β2M) domain. The heavy chain has three sub-domains called the α1, α2 and α3 domains. Similar to β2M with which it interacts, the α3 is conserved. The α1 and α2 subunits are polymorphic and make up the antigen binding groove. The binding of longer peptides is limited as the binding groove is closed at each end, meaning that longer peptides are physically unable to settle into the MHC binding groove without distorting. Due to the limited space in this binding groove, peptides of longer length tend to "bulge" out the centre of the groove. This closed groove and bulge can be seen in Figure 8.

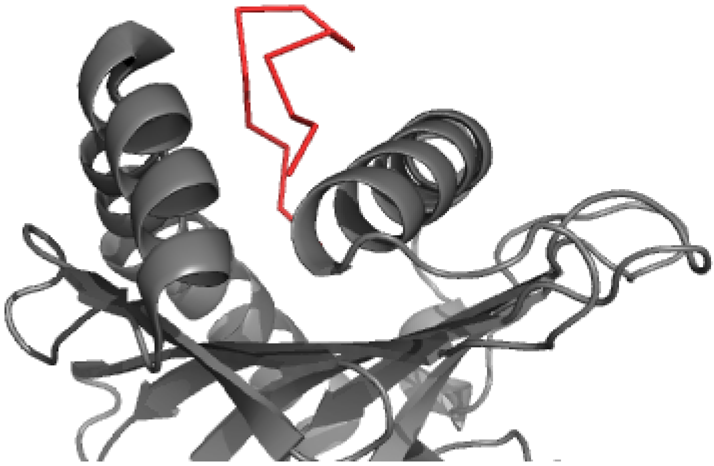


Figure 8 The MHC class I molecule's binding groove (PDB code 1XH3) presenting a 14-mer peptide, the longer length of the peptide means the central residues protrude out of the binding groove and into the CDR loops of the TCR.

MHC class II molecules are formed by an α and β chain, both of which are fixed into the plasma membrane. Both the α and β chains have 2 subunits, called α1 and α2 in the MHCα and β1 and β2 in MHCβ. Unlike the MHC class I molecule, the binding groove is made by both the α1 and β1 subunits. Both chains express polymorphism. The difference between MHC I and II is shown in Figure 9.

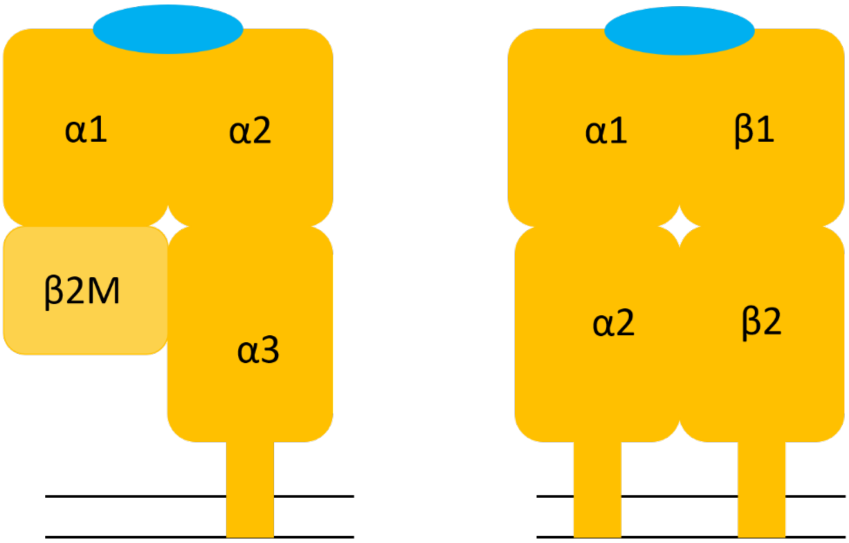


Figure 9 cartoon representation of MHC class I (left) and MHC class II (right). MHC class I is composed of one MHCα subunit and one β2M subunit. The MHCα is responsible for all binding of peptide, unlike in MHC class II, where the MHCα and MHCβ constitute two halves of the MHC binding groove.

The relative openness of the binding groove compared to MHC class I means the N and C terminal residues of the peptide can overhang the groove. The 'extra' amino acids form peptide flanking residues (PFRs) that are often described as the "ends of a hotdog" hanging outside of the MHC class II "bun". Conversely, the middle 9 amino acids form a binding core which constitutes TCR recognition[33] as depicted in Figure 8.

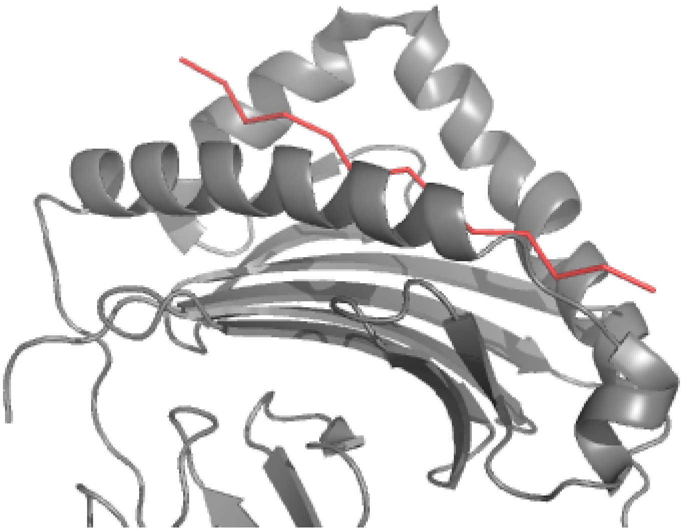


Figure 10 The MHC class II molecule's binding groove (PDB code 1FYT) presenting a 13-mer peptide. Compared to Figure 8, the peptide is much flatter against the binding groove. It also manages to extend beyond the binding groove as the MHC binding domain is opened at the terminal ends, unlike in MHC class I.

A key facet of the binding of peptide to MHC molecules is that the peptide stabilises the MHC (both MHC I and II), meaning that this binding has to be stable. The importance of this stability is demonstrated by the fact that peptide:MHC stability is a better predictor of immunogenicity than outright affinity of the peptide[34].

1.6.2 Antigen processing

MHC class I and II interact with peptides generated by two completely distinct methods of processing and presentation. MHC class I-restricted peptides are of intracellular origin and are produced by means of degradation of proteins by proteasomes, large complexes with proteolytic action[35].

Peptides are then transported to the lumen of the ER by transporter associated with antigen processing (TAP). Peptides produced by the proteasome are then further trimmed to an optimal length by peptidases. This process occurs in the endoplasmic reticulum (ER) by the aminopeptidase ERAP1[36]. Once in the ER, the MHC and peptide form a complex with the aid of chaperone proteins. This complex then leaves the ER by secretion where it heads to the cell surface for presentation to T-cells[37].

MHC class II bind to peptides derived from extracellular proteins generated through endocytic and phagocytic pathways[38]. During development MHC class II proteins are inhibited from associating with antigens by binding to an invariant chain. The MHC class II complex travels through the Golgi apparatus to the MIIC and CIIV compartment[39]. Here, the invariant chain is degraded to Class II associated invariant chain protein (CLIP) where it is then swapped for antigenic peptides with higher affinity for the specific MHC class II molecule.

It should be noted however, that while the mechanism of endogenous peptides being presented on MHC class I and exogenous peptide on MHC class II is the most common, other routes of presentation exist. Some APCs, such as DCs can present extracellularly derived peptides on their MHC class I molecules. This is known as cross presentation[40]. Cross presentation is important as it allows DCs to collect antigens from other tissues, such as those infected with virus[41] or cancerous tissues[42].

1.7 T-CELL MEDIATED IMMUNITY

Upon completing development in the thymus T-cells enter the blood stream and migrate through lymphoid tissues where they circulate between the two. Mature T-cells who have not encountered their specific antigens are called naïve T-cells. To mount an immune response a naïve T-cell must meet and engage its cognate antigen via the TCR. TCR engagement induces T-cell proliferation and differentiation. The progeny of this differentiation is known as effector T-cells.

Upon encountering antigen, naïve T-cells differentiate into several classes of effector T-cells, each having a specialised effector function. CD8 T-cells recognise MHC I presented peptides and differentiate into cytotoxic effectors that destroy infected cells whilst CD4 T-cells have a broader set of effector functions. After recognising MHC II presented peptides, naïve CD4 cells can specialise into a number of different subsets. The main effector subsets are T_H1, T_H2, T_H17, and T_{FH} which activate their target cells; there are also regulatory (T_{reg}) cells which inhibit and modulate the potency of the immune response.

T_H1 subsets, primarily defined as CD4+ and TBX21+ cells are central to responses against virus and cancer. They are show to increase antigen presentation when engaging with DCs and also allow APCs to prime CD8 T-cells[43]. T_H1 cells have also been shown to stimulate macrophages and lead to the destruction of pathogens and stimulate further presentation[44]. In the context of viral immunity T_H1 cells secrete pro-inflammatory cytokines interferon γ

(IFN-γ) and tumour necrosis factor (TNF) which can drive CD8 effectors to the site of infection as well as the involvement of innate cells[45].

T_H2 cells are CD4+ and GATA3+ positive. T_H2 cells are triggered by IL-4 and IL-2. It is suggested they have evolved in response to helminth infection. However, in the absence of helminths they are involved in mediating bacterial and fungal infections[46].

T_H17 cells are CD4+ and retinoic acid-receptor-related orphan receptor (RORC)+ and function via the release of IL-17 and IL-22. They enable the recruitment of neutrophils to sites of inflammation. It is suggested they evolved to protect the host from microbes that T_H1 and T_H2 immunities were not well-adapted for, such as extracellular bacteria and fungi. T_H17 cells produce IL-17 which is a strong mediator of stromal cells which produces inflammatory cytokines and recruits neutrophils, bridging the gap between innate and adaptive immunity[47].

T_{FH} play an important role in the formation of germinal centres that are structures that form in secondary lymphoid organs during a persistent immune response, as well aiding affinity maturation and development of high affinity antibodies and B-cells[48].

1.8 Methods and technology for understanding TCR-antigen recognition

Given the amazing diversity of the TCR-pMHC, it is unsurprising that understanding of TCR-antigen recognition is wrought with difficulties. Most mechanistic knowledge has been gathered from a very small pool of TCR-pMHC protein structures. At the time of writing there are 185 crystal structures publicly available in the PDB (as counted by the STCRDab: http://opig.stats.ox.ac.uk/webapps/stcrdab/)[49]. This means any assumption made about TCR-pMHC may only represent the "tip of the iceberg" in terms of overall variation. Similarly, next-generation sequencing (NGS) of TCRs remains relatively naïve compared to other NGS technologies. The complete list of TCRs with a known antigen specificity[50] represents only a tiny fraction of known TCR sequences. There are 81,762 TCRα and TCRβs in the literature, with only 25,881 full αβ TCR complexes (taken from vdjdb.cdr3.net [50] on 8th November 2021).

This scarcity in data, along with the inherent complexity of TCR-antigen recognition means that there are a wide number of different technologies which must be utilised, often in conjunction with one-another to explore the mechanism of TCR pMHC recognition. These technologies are underpinned by several computational methods. Both the technologies and the computational methods they rely on will be discussed below.

1.8.1 High throughput TCR sequencing

Most often, TCR sequencing is performed on genomic DNA (gDNA) and performed “in bulk” across a population of cells (e.g., whole blood or tumour). Bulk sequencing informs on the overall abundance of a given TCR chain, but does not provide pairing information[51].

TCR sequencing protocol varies depending on the exact technology used. However, the basic principle is that either a chain in its entirety is sequenced, or just the CDR3. The TCRβ chain includes a D segment and has a higher diversity than TCRα[52] and is therefore generally the chain of choice for single chain sequencing. Most commonly, only the CDR3 is sequenced as such protocols lend themselves well to the short-read lengths and high throughput achievable with the Illumina platform. As only the CDR3 sequence is non-germline the entire TCR chain can be assembled if the V gene can be assigned using software such as MiXCR (https://mixcr.readthedocs.io/en/master/).

1.8.2 RNA sequencing

The above methods give a detailed view of TCR diversity. As they rely on targeted sequencing, they do not capture any of the other transcriptional diversity happening outside of the area chosen to be sequenced.

CDR3 sequences can be extracted from bulk RNA-seq experiments. This is computationally difficult as in tissues T cells are not abundant and make up approximately 1 in every 2000 transcripts[53]. One would expect much less efficiency in sequencing this way but if wider transcriptional profiling is required then RNA-seq can provide a potentially interesting method for TCR profiling. A comparison of this method and gDNA sequencing is shown in Figure 11.

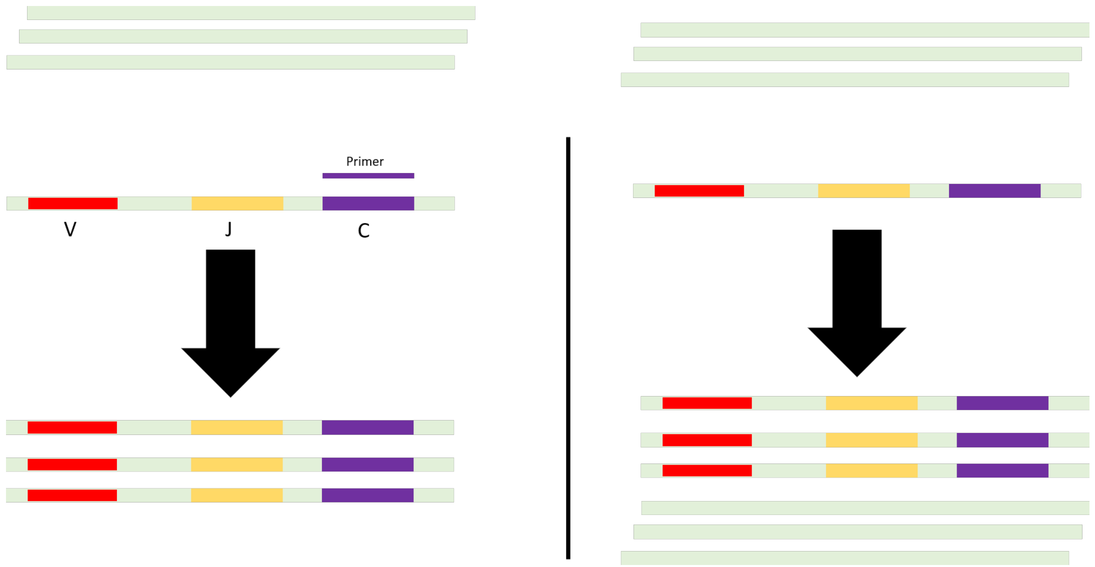


Figure 11 A comparison between gDNA sequencing of TCRs (left) and RNA-seq of TCRs (right). Non-TCR fragments are shown light green, while TCR fragments are annotated with a red, yellow, and purple band, representing the V, J and C segments respectively. In gDNA sequencing, specific primers, in this case for the C locus of the TCR mean that only TCRs fragments are amplified and sequenced, ignoring non TCR fragments. RNA-seq makes use of shotgun sequencing, so all transcripts in the sample are fragmented and have adapters annealed to them, therefore the whole array of fragments are sequenced, including those with no TCR on them.

1.8.3 Single cell sequencing (SCS)

Single cell RNA-sequencing (scRNA-seq) is a much newer technology than bulk RNA-seq. scRNA-seq works by rapid isolation of individual cells, often via fluidics. Sequencing is then performed using reverse transcription, an amplification step and library preparation. Amplification can be done using a nucleic acid sequence tag (commonly referred to as a barcode) or by using full-length cDNA sequencing. The obvious advantage of SCS of T-cells is that TCRα and TCRβ can be paired to produce the entire *functional* TCRαβ heterodimer. This technique can also provide information on the rest of the transcriptome to allow T-cell phenotyping (CD8+, CD4+, Treg etc).

1.8.4 TCR repertoire alignment and assembly

The exact protocol of the above three technologies vary, but they all result in a TCR CDR3 that is comprised of gene fragment and insertions/deletions which must be aligned, assembled, and quantified. The first step entails aligning the TCR fragments to a known reference and determining the gene usage. While this is a well-established step in classical genome assembly, it is more difficult with TCRs as they are made up of V, (D) and J segments which are very similar and must be untangled, usually this involves alignment directly to a reference set of V, (D) and J genes supplied by the IMGT database[54]. The exact step of alignment varies from implementation to implementation, with some methods using the BLAST[55] algorithm[56] or BLAST with a secondary alignment step[57]. Similarly, other tools may use alternative alignment softwares such as Bowtie2[58] for gene detection[59]. Other tools may opt for a "tag" based approach where their algorithm searches for exact matches of small substrings[60]. Finally, others may use a hybrid of the two approaches[61,62].

Once alignment has been performed, the abundance of a sequence is estimated. This is done by counting the same CDR3 sequence if the CDR3 sequence is viable; the DNA sequence is in frame and there are no premature stop codons. This step can be corrected for if the sequencing technology included used unique molecular identifiers (UMIs).

Finally, there is an error correction step where low quality (based on the PHRED score in a FASTQ file) and low abundance sequences are removed. Other methods include clustering all sequences by their Hamming distance (the number of amino acids different between two strings) and removing outliers[62].

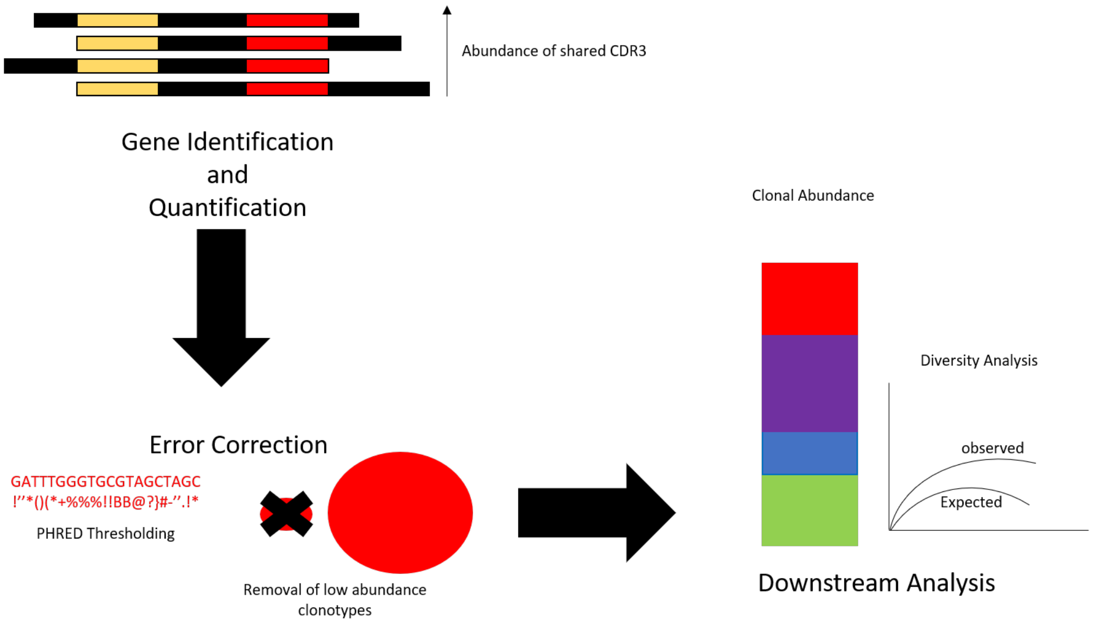


Figure 12 An overview of the steps of TCR repertoire analysis, from gene identification(top) and quantification of a clonotype, to error correction (bottom left) and finally downstream analysis (bottom right)

The final step is downstream analysis of the TCR sequence data, which can be incredibly varied and dependent on biological context and the aims of the experiment. Downstream analysis can include clonotype abundance and ecological diversity measures such as Shannon or Simpson diversity[63]; measure of abundance of a given clonotype against a theoretical distribution[64] and motif enrichment analysis[65].

1.8.5 Structural modelling

As previously discussed, 3D protein structures of TCRs and their cognate pMHCs are essential for disseminating the mechanism of interaction between TCR and antigen. Examples of the role of 3D structures are numerous, including being used to explain epitope escape having undergone mutation[66]; the knowledge of the “induced fit” of different peptide anchor residues [67] and the occurrence of cross-reactivity[68].

TCR structures are essential for “looking back” and explaining already understood phenomena and to ascribe a mechanistic explanation to it. Increasingly, the reverse is true where structures are being used proactively to predict of unknown 3D structures from sequence and in the rational design of therapeutic TCRs.

Design of TCRs has been used to model point mutations and observe their impact on structure. This is has to be done dynamically using tools such as Rosetta[69]. Such methods have been used to demonstrate that hydrophobic substitutions are often conducive to improved binding affinity as there is an improvement in interface complementarity between the TCR-pMHC[70]. Indeed, this is logical as studies have shown tryptophan, methionine and phenylalanine are the most highly conserved residues found in binding sites of proteins[71]. Other methods focus not only on binding affinity but specificity also[72].

Modelling plays an important role in TCR p-MHC structure prediction. Given the difficulty of resolving structures, sometimes prediction of the 3D structure is the only alternative to genuine structural analyses. While the general prediction of 3D structures is a well-developed field, modelling TCRs remains difficult due to the dynamism of the TCR being able to adopt multiple conformations to allow for cross-reactivity. Fundamentally, these algorithms work by aligning the TCR amino acid sequence to a reference TCR with a resolved structure, then determining the conformation of the CDR loops using anchors at the start and end of the loop as points to which the predicted sequence is grafted onto. Then the backbone atoms of each amino acid are iteratively grafted onto the template sequence[73]. Following that, side chains are refined to reduce steric clashes[74,75]. TCR structural behaviour has also been used to inform sequence based models[76] as discussed below.

1.8.6 Sequence based modelling.

The advent of NGS protocols for TCR sequencing has generated enough data that TCR epitope prediction from sequence is possible. Epitope prediction is framed to take the form of a classification problem where the user supplies a sequence of a TCR (typically the sequence of the CDR3β alone) and a potential epitope and the user receives a Boolean prediction whether the TCR will bind, with no answer to the degree at which this interaction occurs.

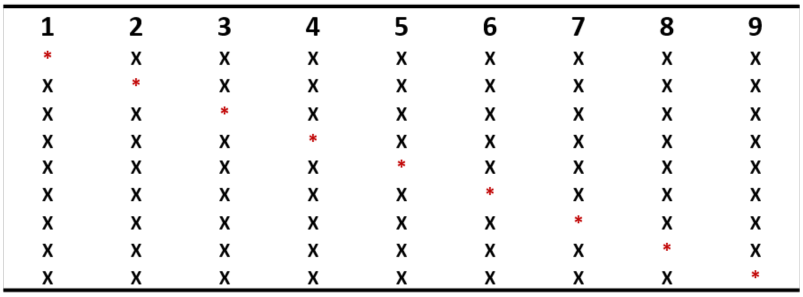
While the exact methods differ, the core methodology is well-conserved and entails encoding the CDR3 and epitope sequence into a 2D matrix of physiochemical features, along with data pertaining to the V/J gene usage. The exact method of prediction varies, with several tools opting to use convolutional neural networks (CNNs)[77,78], Gaussian processes (GPs)[79] or random forests[80]. All these methods share a commonality in that they allow for spatial features of the TCR-epitope to be retained.

Another facet of sequence-based modelling is the clustering of TCRs into groups with similar or identical epitopes, based on their properties. These methods make use of detecting motifs in both TCRs and epitopes. String based methods can then be used to create a pairwise distance metric between TCRs as shown by *Dash et al.*[81] or to calculate global and local convergence as in *Glanville et al.*[76]*.* Other methods employ a graph based method to find over-represented TCRs in NGS data[82] by comparing it to a theoretical distribution generated by a Bayesian model[64].

1.8.7 Combinatorial peptide libraries

Combinatorial peptide libraries (CPLs) provide a way screen for recognition of a vast number of peptides by a given T-cell clone. For a 9-mer peptide, a CPL is divided into 180 different peptide mixtures arranged alphabetically in positional scanning format using single letter amino acid code. In each peptide mixture, a single residue remains fixed, while the remaining 8 residues can be made up for any one of 19 naturally occurring amino acids (cysteine is excluded as it causes peptides to aggregate via disulphide bond formation). Recognition of CPL by T-cells informs on the importance of each amino acid at each fixed position as shown in Figure 13. These mixtures are then exposed to the T-cell of interest and activity of the T-cell is measured by an ELISA of macrophage inflammatory protein-1β (MIP-1β); the most sensitive readout for CD8+ T-cells[83,84]

This approach has been used to predict potential existing epitopes[85], as well as estimate the breadth of cross-reactivity of a given T-cell clone[26].



1	2	3	4	5	6	7	8	9
*	X	X	X	X	X	X	X	X
X	*	X	X	X	X	X	X	X
X	X	*	X	X	X	X	X	X
X	X	X	*	X	X	X	X	X
X	X	X	X	*	X	X	X	X
X	X	X	X	X	*	X	X	X
X	X	X	X	X	X	*	X	X
X	X	X	X	X	X	X	*	X
X	X	X	X	X	X	X	X	*

Figure 13 An example of a 9-mer amino acid CPL library. The "X" indicates a fixed amino acid, and the "O" denotes a degenerate mix of amino acids. There are 180 of these per library scan, meaning each combination of fixed amino acids is covered.

1.8.8 Knowledge driven models.

A number of tools and predictive methods incorporate non-immunological parameters as a "context" for enhanced predictive power. This is particularly prevalent in neoantigen, and tumour associated antigen (TAA) prediction. For example, weighting a prediction of a MHC class II antigen in the context of neoantigens has been shown to increase predictive performance[86]. This same study showed that there was a bias in peptides that were secreted or expressed on the cell membrane.

The importance of subcellular location of antigen is not just limited to self/cancer immune recognition. The Ovalbumin (OVA) antigen has been used in both *Escherichia coli*[87] and *Toxoplasma gondii* [88] to demonstrate that subcellular localisation has an impact on CD4+ and CD8+ T-cell activation and expansion, respectively.

2 AIMS OF PROJECT

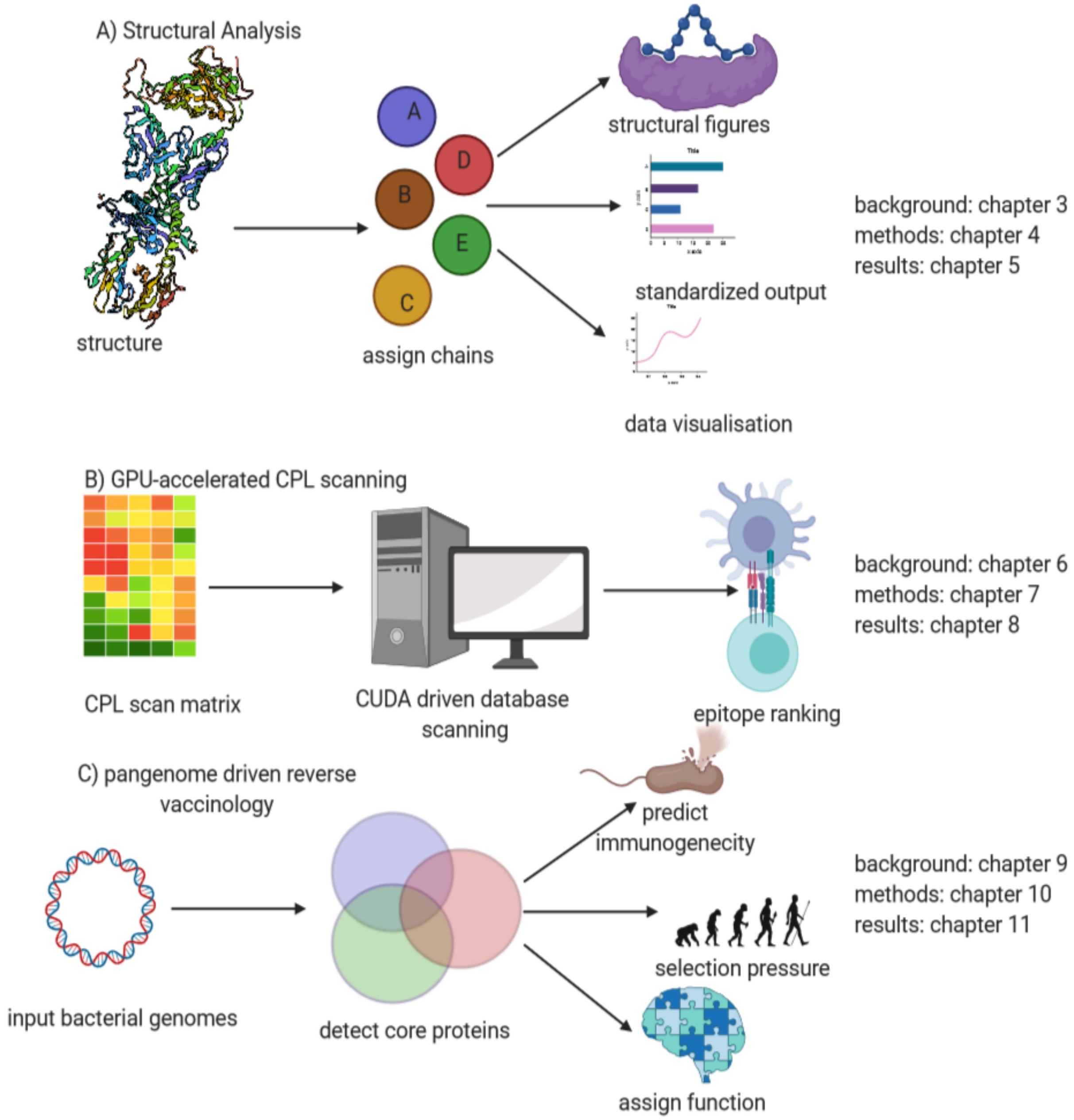


Figure 14 the function of each of the three pillars of the thesis. A) Discusses STACEI, a tool for structural analysis of TCR-pMHC structures. B) discusses GPU-accelerated epitope prediction through CPL scans and finally C) discusses RECIPIENT, a tool for pangenome reverse vaccinology.

As discussed above, interaction between TCRs and pMHC complex is incredibly complex and a broad understanding of these interactions requires a range of technologies. Most of these technologies require computation and bioinformatics. In this thesis, I aim to present three methods of exploring the TCR-pMHC interactions as follows:

1. STACEI (structural tool for analysis of TCR-pMHC interactions): A tool for the structural analysis of TCR-peptide-MHC Interactions
2. Application of parallel computing to CPL driven epitope prediction
3. RECIPIENT (reverse vaccinology for protein vaccine candidates): a pangenome reverse vaccinology tool

The general workflow of each of these chapters are shown above in Figure 14.

These three methodologies aim to understand the TCR-pMHC complex across different scales of biological interpretation. As they require different bioinformatics backgrounds, I will discuss the preliminaries of these in the following chapter.

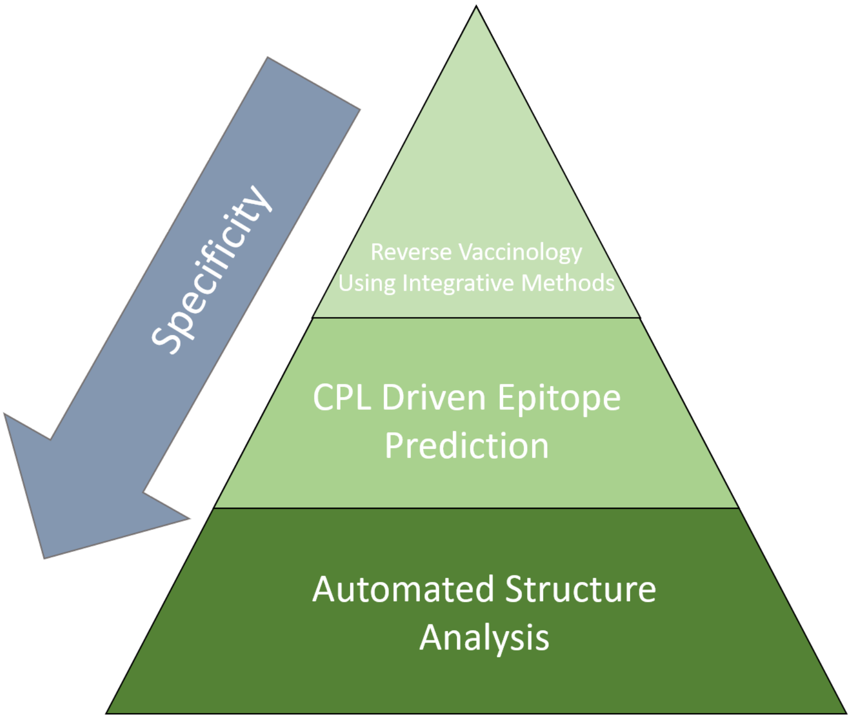


Figure 15 The 3 computational methods described in this thesis. Each theme moves to a more TCR-specific method for understanding the TCR-antigen interaction and its application towards vaccine design.

Although these three chapters differ in their methodologies, the biological goal that underpins them is the same: to enhance understanding of the interaction between TCR and antigen across different scales of biological interpretation, going from deriving mechanistic insights of TCR-pMHC recognition at a structural level; identifying epitopes for T-cells using

CPL scans and parallel computing and broad context-dependent discovery of vaccine targets using reverse vaccinology (RV).

2.1 AIMS OF STACEI: A TOOL FOR THE STRUCTURAL ANALYSIS OF TCR-PMHC INTERACTIONS

Section 1.8.5 discusses the importance of structural biology in deeply understanding the precise molecular mechanisms governing TCR-pMHC recognition. This will be expanded further on in Chapter 3. While numerous structural biologists study TCR complexes, there is very little in the way of software and tooling for the bespoke analysis of TCR-pMHCs specifically. This analysis is performed often on an *ad hoc* basis, meaning that the process is time-consuming and wrought with potential avenues for human error. Different groups have defined molecular interactions in different ways over time so there is an important need for standardised methodology for comparisons. I will discuss the creation of STACEI (structural tool for analysis of TCR-pMHC interactions) and its role in automating, standardising and ultimately expediating this analysis. STACEI will be discussed in chapters 4 and 5.

2.2 AIMS OF THE APPLICATION OF PARALLEL COMPUTING TO CPL DRIVEN EPITOPE PREDICTION.

As discussed briefly in 1.8.7, CPL scans provide a powerful avenue for assessing both TCR cross-reactivity as well as the outright prediction of T-cell epitopes. This will be introduced, along with the concepts of parallel programming in chapter 6. In chapters 7 and 8 I will discuss the role of using parallel computing, namely Compute Unified Device Architecture (CUDA) to speed up and enhance these predictions previously carried out by Peptide Identification by CPL (PICPL).

2.3 AIMS OF RECIPIENT: A PANGENOME REVERSE VACCINOLOGY TOOL

Chapters 8, 9 and 10 will discuss RECIPIENT (reverse vaccinology for protein vaccine candidates), a tool for the identification of potential peptide or full-protein vaccinations using RV. RV is the concept of using pre-existing knowledge, usually at an 'omics level to try and predict vaccine targets. In this section of the thesis, I will discuss the implications on integrating T-cell epitope prediction, MHC binding predictions, subcellular location, and population genetics methods to pangenomes in order to predict T-cell epitopes.

3 BACKGROUND: STACEI- A TOOL FOR THE STRUCTURAL ANALYSIS OF TCR-PMHC INTERACTIONS

Structural data of TCR-pMHCs, derived almost exclusively from x-ray crystallography, is key to understanding the function of these recognition events at the molecular level. This information has been helpful in revealing the mechanisms of recognition of tumour, bacteria, and viruses; along with the role of TCRs in recognition of self-antigens in the context of transplant rejection and autoimmunity.

3.1.1 Generalised tools for analysing TCR-pMHC structures

Most structural biology software for the analysis of these TCR-pMHC structures is not bespoke for analysing these complexes, rather general-purpose tools for analysing any macromolecular protein complexes are used. The lack of bespoke, customised software means that several tools are used on an *ad hoc* basis. In addition, most of general molecular recognition software has a specific goal in mind. For example, there are a number of tools specifically designed for visualising PDB structures, including RASMOL[89], CCP4MG[90] and PyMOL[91]. While these tools offer some flexibility in analysing structures, they are by no means field leaders when it comes to gaining mechanistic insight into a structure (other than that gained from visualising the 3D structure).

A number of tools exist for the mechanistic insight of structures and determination of contacts between residues. A contact is usually defined as two atoms being within 4Å, however the exact distance used has varied over time and across different research groups. Contacts can be calculated a number of ways including using an application programming interface (API) or library to read the file programmatically, such as BioPython[92] and Atomium[93] in Python; BioJulia (https://biojulia.net/) in Julia and Rpdb (https://cran.r-project.org/web/packages/Rpdb/index.html) and bio3d[94] in the R language. The existing general tools all require the user to be able to access the packages mother programming language, install that package and use the API to run through the steps to get the data they require. An alternative is using a software with a graphical user interface (GUI). A tool with a GUI tends to be easier to install and is more user-friendly to a structural biologist who may not have a programming background. The main example of a GUI tool that calculates contacts is NCONT and its predecessor CONTACT[7], part of the CCP4 suite[95]. The CCP4 suite has executable binaries of all of its softwares as well as a GUI. PyMOL also offers a contact function, shown in Figure 16.

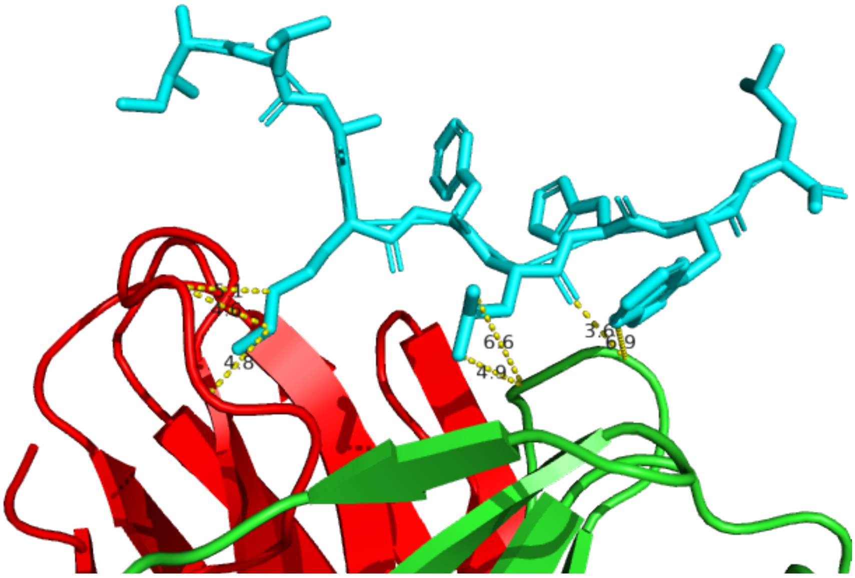


Figure 16 An example of measured contact distances in PyMOL. The distance is measured between two atoms in Euclidian space. It can be used to show the engagement of the TCR (TCRα in red, TCRβ in green) with the peptide (cyan). The PDB file in question is 5MEN.

Another metric commonly applied to TCR-pMHC complexes that is highly generalizable is buried surface area (BSA). BSA is a geometric quantity that measures to total surface area (in Å^2) of a complex buried within a another complex[96]. In terms of the TCR-pMHC this is useful for a number of reasons, it can be used as a proxy for contact formation in low-resolution structures, informing how much each CDR loop is contacting the pMHC and contributing to recognition[22], as well as being used as a quick comparison between numerous structures [22,97]. BSA can also be used to determine which parts of the peptide are buried/acting as anchors to the MHC and which are available for recognition by the TCR[98]. At the time of writing, there appears to be only two commonly used tools for this purpose: PISA[99] which again is part of the CCP4 suite as a standalone binary and GUI tool, as well as a webtool (https://www.ebi.ac.uk/pdbe/pisa/pistart.html) and Chimera[100].

The final general method used to inspect the TCR-pMHC complex is shape complementarity (SC). SC can be viewed as a measure of “goodness of fit” between two complexes (Figure 17).

It depends on the overall structure of the complexes, as well as how the interaction between them brings parts of each complex into contact. SC exists as a score between 0 and 1, where 0 represents no proximity between the two complexes and 1 represents a full interface[101]. This measure has a long history of being used to assess the interface of the TCR and pMHC [102,103]. Current implementations for calculating SC include CCP4[104] and Rosetta[69].

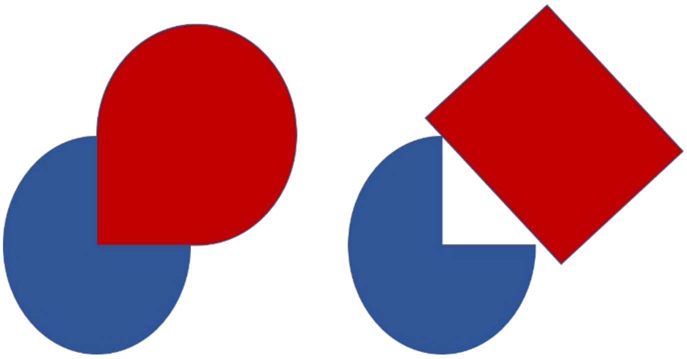


Figure 17 Cartoon representing SC. A high SC interaction will show two complexes in a large amount of contact, forming one interface (left), which a poor or low SC will be indicative of an interface with a large degree of gaps in it (right)

All the above methodologies have well-characterized tools available already. However, their use in the context of TCR-pMHC complexes still requires a degree of user experience in working these tools and applying them correctly to the TCR-pMHC structure. Normally, this would require an expert user to know the constituent chains in the PDB file for each protein in the structure (TCRα, TCRβ, peptide, MHα and β2M/MHβ) as well as the MHC class restricting the peptide as this must be known and given to the software *a priori*.

3.1.2 Bespoke analysis of TCR-pMHC complexes

Unlike the above methodology, some of the routines used to analyse these complexes are not present as a software package. For example, the contacts and BSA of a structure may be able to be computed using a software package, but there are no pre-existing tools that allow ready exploration and visualisation. This existing software requires manual detection/assignment of CDR loops and manual counting of contacts and their nature. The structural biologist must then port these data into the plotting software of their choosing to generate data or figures. The same manual input is required for visualising 3D structures in PyMOL. In addition, these tools do not allow for ready description of features used to describe the TCR-pMHC recognition landscape such as crossing angle.

3.1.3 Standards in structural biology and analysing TCRs

3.1.3.1 The PDB File

Structural biology utilises numerous well-conserved and defined standards for analysing structural complexes. Probably the most obvious, is the PDB file. The PDB format is curated by the Research Collaboratory for Structural Biology (RCSB)[105] and described at http://www.wwpdb.org/documentation/file-format-content/format33/v3.3.html. The start of each line of a PDB file is annotated with a term explaining what the line represents. These header annotations, known as records, are vast so the main ones are summarized in Table 1.

Table 1 a summary of the main header annotations found in a PDB file.

Remark	Description
HEADER	Contains metadata about file, including date of deposition and type of structure
TITLE	Title of experiment, often the same as the title of the paper associated with it
SEQRES	Record of covalently linked amino acids/nucleic acids
SSBOND	Identifies location of disulphide bonds
ANISOU	Information about anisotropic temperature factors
ATOM	Information about each atom and its location in 3D space

The most important header summarized in Table 1 is the ATOM header. ATOM describes the location of each atom in 3D space, as well as information about what amino acid it is a part of, what the residue number is and whether it is an insertion of deletion. The ATOM header is described in more detail below in Table 2. To summarize, the ATOM header is a column delimited line where each piece of information is stored in a certain set of columns which are then filled with whitespace. This consistency in the PDB header allows for easy parsing and handling computationally.

Table 2 Summary of the content of the ATOM header

Start column	End column	Name	Description
1	6	ATOM	Header
7	11	serial	Serial atom number
13	16	name	Name of atom
17	NA	altLoc	Alternate location
18	20	resName	Residue name
22	NA	chainID	Chain identifier
23	26	resSeq	Residue sequence number
27	27	iCode	Code for insertion residue
31	38	x	Coordinate in X plane in Angstroms
39	46	y	Coordinate in Y plane in Angstroms
47	54	z	Coordinate in Z plane in Angstroms
55	60	occupancy	Occupancy factor
61	66	tempFactor	Temperature factor
77	78	element	Element symbol
79	80	charge	Charge of atom

Asides from the *ATOM* header, another important header is the *HETATM* field. The *HETATM* field represents “non-standard” residues. Here, “non-standard” refers to residues that do not originate from either nucleic acids or amino acids. These can include co-factors and inhibitors, ions and solvents. Water is classically stored in the *HETATM* field. In the case of non-peptide derived antigens, such as those recognised by γδ TCRs.

3.1.3.2 IMGT standards for describing TCR sequences

The other side of the standards regarding TCR-pMHC structures has been overlooked by the structural community is the IMGT numbering scheme. The IMGT numbering scheme renumbers the TCR chains to allow for consistent comparison between CDR loops as they vary in length. Each CDR loop is constrained such that the first and last residue remain the same regardless of length. The CDR1 and CDR3 loops are determined by counting in from alternating sides of the loop, which are fixed by a conserved cysteine at the start and a conserved aromatic residue at the end, travelling from the N to C terminus. The CDR2 is determined by an anchor residue at position 54; and an anchor at 67 for the CDRα and position 69 for the CDR2β. In longer CDR3 loops, there are additional residues between position 111 and 112 represented by a decimal value (111.1, 111.2 etc.). This is shown visually in Figure 18.

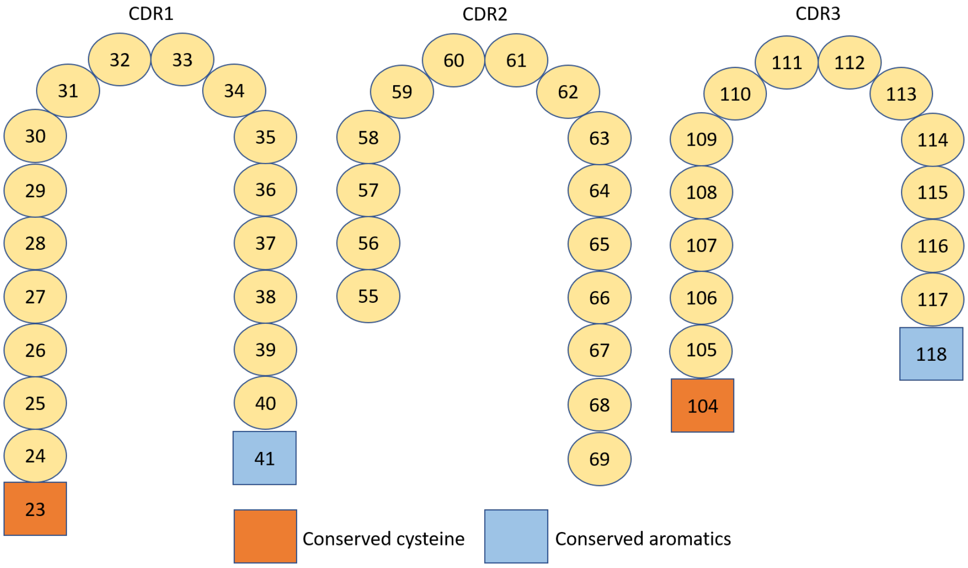


Figure 18 IMGT numbering scheme for TCRs, shown for the CDR1 (left), CDR2 (middle) and CDR3 (right). The conserved anchors for CDR1 and 3 are shown as they are conserved. The anchor residues for CDR2 vary between CDR2α and CDR2β.

While there are tools for the annotation of TCR-pMHC structures[49], this is usually done separately to analysis and appears not to have been taken up in the literature. This is most likely because those who are performing the analysis are not comparing across a large number of samples. Reliable comparison between structures requires that the IMGT residue numbering system be adhered to. This serious issue is only relevant to the structure of antigen receptors due to the unique quasirandom nature of the V(D)J process that generates them as described in section 1.5.1. There is one pre-existing tool that will annotates a TCR sequence to the IMGT standard, ANARCI[106], ANARCI leverages a hidden Markov model (HMM) implemented in the HMMER package[107] to first detect TCRα and TCRβ chains before detecting domains of the TCR and numbering the amino acids accordingly. ANARCI does not work directly on PDB files, hence a PDB file must be converted into a FASTA file prior to numbering.

4 METHODS: STACEI- A TOOL FOR THE EXPLORATION OF TCR-pMHC COMPLEXES

4.1 AIMS

The aims of STACEI was to provide a robust platform for the analysis of TCR-pMHC structural complexes. As discussed in the introduction, there are a numerous databases available for collecting basic information about TCR-pMHC complexes and a number of general tools for analysing protein structures. However, very little tooling exists in the niche between the two, that allows for the bespoke, in-depth interrogation of TCR-pMHC structures.

4.2 DATA COLLECTION

To provide testing material for the development of STACEI, as well as reviewing the current scope of TCR-pMHC structures, a dataset of all TCR-pMHC structures found on the PDB was generated. At the time of writing there are 135 MHC class I restricted structures and 51 MHC class II.

The annotation of TCR-pMHC structures is often varied in style and some structures do not wholly adhere to the PDB specification. For example, the header file does not strictly specify the peptide/antigen and instead refer to it as the amino acid sequence it has. Likewise, there is no guarantee the MHC or TCR are named accordingly. This exists both in the header file of the PDB itself and the annotation in the PDB website (if the structure is publicly available).

The PDB API names and describes the contents of PDB files by the header and deposition information. This is not done in a consistent style, so a web scraper written in Python was written to manually collect structures from the database. The script loops through all structures (169,436 at the time of writing) and stores their ID in a *set()* object in Python. If the structure contains 5 chains, then it is then downloaded to be assessed. The PDB file is then converted into a FASTA file using the *Biopython* module. The FASTA file is then passed into ANARCI[106] to detect both α and β subunits of the TCR. If this criterion is not met, then the structure is skipped. If not, the remaining sequences not detected as the TCR are aligned against a database of MHC alleles[14] (downloaded from https://github.com/ANHIG/IMGTHLA/tree/Latest/fasta) using the Smith-Waterman algorithm using the *swalign* Python package (https://pypi.org/project/swalign/). If both the MHCα and β2M/MHCβ are detected, based upon a similarity score >300 then the file is kept. The *set()* object is then stored so that the next time the script is ran these structures are skipped meaning only new structures are analysed.

4.3 Workflow

STACEI is written primarily in Python, that integrate numerous pre-existing analyses from other structural biology software with novel analyses. The general workflow is described in Figure 19. The workflow first collects and input PDB file, detects which chains are the TCRα, TCRβ, peptide, MHCα and β2M/MHCβ; performs a number of housekeeping and normalisation steps. The MHC class is also determined. Once these steps have been performed, several downstream structural analyses are performed. These include the analysis of contacts, shape complementarity, BSA and crossing angle. The files generated by those analyses are then used to produce a collection of static and interactive visualisations in the form of tables, plots, graphs and 3D structures generated in PMOL. These data are then summarized in a HTML document.

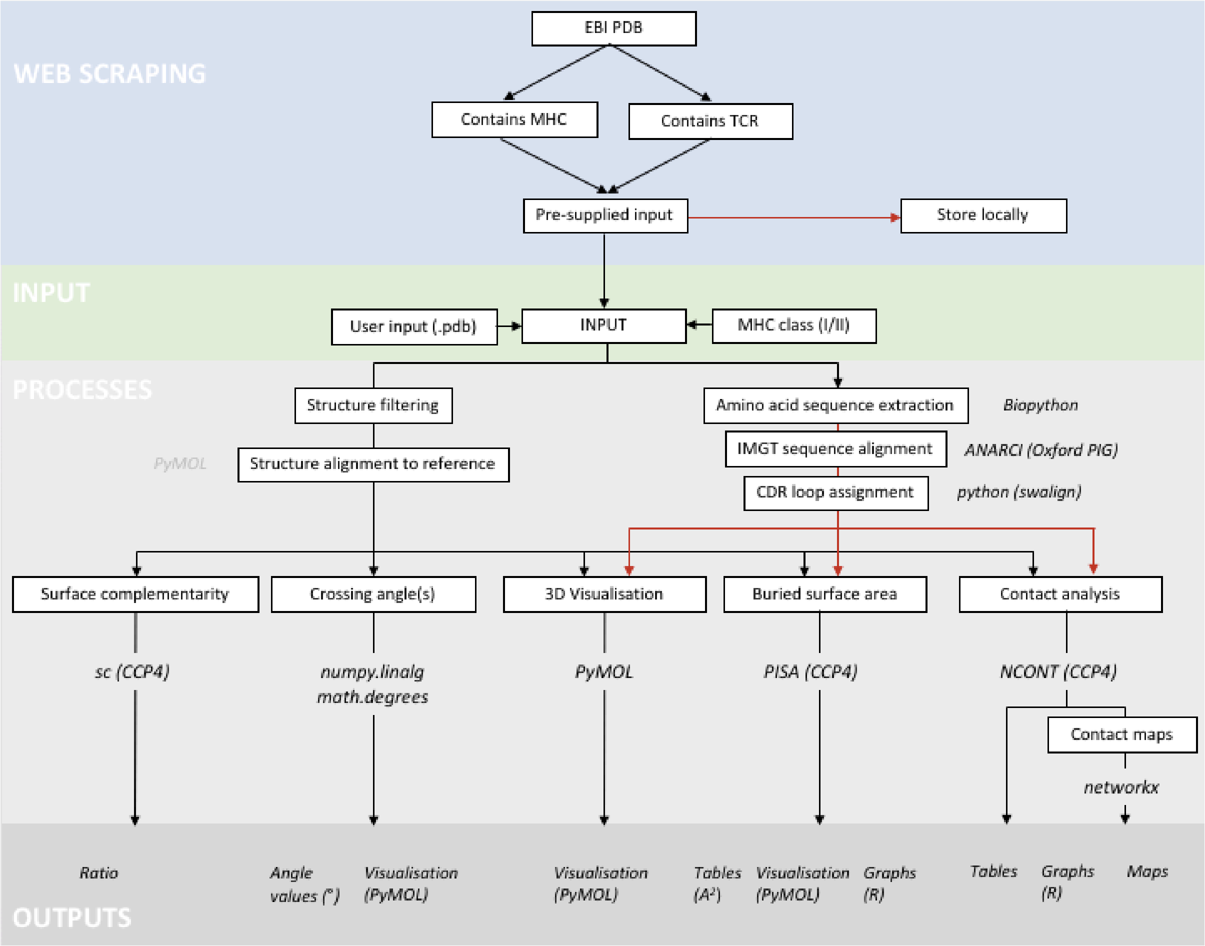


Figure 19 The STACEI workflow. First the structure is annotated and its MHC class is determined, then the annotated structure is passed down a number of "arms" of the analysis. These include surface/shape complementarity, crossing angle, BSA and contacts. Workflow diagram was created internally with Dr. Bruce MacLachlan for use in presentations.

4.3.1 PDB file cleaning

Most the PDB files fed into STACEI require cleaning in order to prevent errors in downstream analyses. In particular, the CCP4 programs and any modules using Biopython can often throw errors when interacting with insertions and deletions in the sequences; isomeric conformations; gapped numbering in the α of the PDB file and the presence of small molecules.

In order to deal with faulty or poorly annotated PDB files, STACEI first performs a sanitation step. This module collects all the ATOM denominated parts of the PDB file. These atom chains are then checked for the presence of alternative locations by looking for the presence of a letter in index 17 of the PDB file (as discussed in the introduction, PDB files are column delimited). Alternative locations indicate the presence of atoms indicated in more than one location, either because the atom is mobile or because the structure could not be resolved. If an alternate location is discovered, the first location is taken. In analysing all the TCR-pMHC structures in the PDB, none of these alternative locations were found to be near the interface of the TCR-pMHC complex, but if the atom is within the interface the user is encouraged to replace these lines themselves, as there is no way to tell which is correct without *a priori* knowledge of the structure.

A non-standard annotation in PDB files is the definition of insertions and deletions at column 21 of the ATOM header. In the same iteration as above, any residues with letter in column 21 are deleted and the chain is renumbered (e.g. 112A becomes 113) as well as all the chains downstream of this.

As all the non-ATOM headed lines are removed. STACEI is not suitable for structures with non-amino acid ligands (e.g. γδ TCR complexes). This decision, however, was a conscious decision as non-amino-acid residues have caused errors in CCP4 programmes and in Biopython.

4.3.2 Chain and class determination

The user can provide chain annotation themselves via the command line arguments of STACEI, however this can be a hindrance when analysing a large quantity of files. Therefore, to avoid needing user input STACEI identifies the chains and MHC class of a TCR-pMHC complex. This determination is the same as described as above, TCR chains are detected by ANARCI and MHC by the *swalign* package.

Once the TCR-pMHC chains have between identified it is necessary to find which TCRs bind to which MHCs in the case of PDB files with multiple units. For TCRs this is performed by calculating the distance between the conserved cysteine residue at position 104 in both the α and β chains and pairing each based on closest proximity to one another in Euclidian space. For MHC this is designated by calculating the greatest number of contacts between MHCα and β chains at residues 20-80. TCR-MHC complexes are then determined by finding the pairing of TCR and MHC with the largest number of contacts made by residues 80-120 of the TCRα and TCRβ chains to the MHC. Peptides are then selected as chains that are not determined to be a TCR or MHC chain by ANARCI or alignment and are of a relevant number of residues (<30 amino acids). The correct peptide held within the MHC is then determined as the peptide making the most contacts to a TCR-MHC complex. Users can also supply chain information in the program, should they wish to analyse a specific monomer of the supplied PDB file. All of the contact calculations are performed by making calls to CCP4's NCONT via a self-contained Python module, the output is written to a text file before being parsed and counted again in Python.

4.3.3 Gene usage and CDR loop annotation

STACEI makes use of the IMGT numbering scheme to assure that analyses are comparable between different TCR-pMHC complexes. For example, with structures from the PDB, there is no guarantee that the conserved cysteine at position 104 is actually at residue 104 in the PDB file due to inconsistent numbering. The IMGT numbering scheme also accounts for variable length CDR loops, meaning that it is easier to compare between CDR loops.

The IMGT numbered file is created by a Python snippet reading the ANARCI output file containing the variable region of the TCR and the corresponding numberings. The amino acids in the ANARCI file are concatenated to make a single sequence which is then aligned against the original file using the swalign package's Smith-Waterman alignment. This allows STACEI to find the start of the variable region numbered by ANARCI and join it to the constant region in the original PDB file. The constant region is then numbered to match where the variable region starts, as defined by ANARCI.

The only assignment that is non-canonical is the CDR2 loops. It was observed that some CDR2 loops, while annotated correctly in terms of gene usage, did not fully encompass all contacts made by that loop of the TCR. This behaviour is shown in Table 3 Example of the extended CDR2 definition scheme using the TCR 5MEN. Using the canonical IMGT annotation the CDR2α would range from residues 56-64, wherein the range of CDR2 contacts were expanded beyond that of the gene definition. This range was determined to be from residues 56 to 64. CDR1 and CDRfw loops are determined by taking residues from within the range adopted by the IMGT. CDR1, CDR3 and the framework are left to be determined by ANARCI.

Table 3 Example of the extended CDR2 definition scheme using the TCR 5MEN. Using the canonical IMGT annotation the CDR2α would range from residues 56-64, meaning that some residues that make contacts would not be annotated.

Donor Chain	Donor Annotation	Canonical	Donor Residue	Acceptor Chain	Acceptor Annotation
TCRA	CDR2a	No	55	MHCA	MHCa2
TCRA	CDR2a	No	55	MHCA	MHCa2
TCRA	CDR2a	Yes	57	MHCA	MHCa2
TCRA	CDR2a	Yes	58	MHCA	MHCa2
TCRA	CDR2a	Yes	58	MHCA	MHCa2
TCRA	CDR2a	No	66	MHCA	MHCa2

4.3.4 Buried surface area

Buried surface area (BSA) and available surface area (ASA) are measures of the total area that makes contact between two interfaces. They are commonly used to describe TCR-pMHC engagement[108]. BSA and ASA are both calculated using CCP4's PISA[109]. Both are calculated for the pMHC alone and the whole TCR-pMHC complex. Assuming the that BSA > 0, a value for availability (e.g. how much of a given residue is buried relative to how much is free) is calculated:

$$availability = 100 - \left(\frac{ASA - BSA}{BSA} \times 100\right)$$

Once BSA has been calculated in PISA, the residue levels of BSA and ASA are parsed using a Python script. These numbers are then used to calculate availability as above. The values are then integrated with the residue number, amino acid letter, chain and (if necessary) CDR loop or MHC subunit.

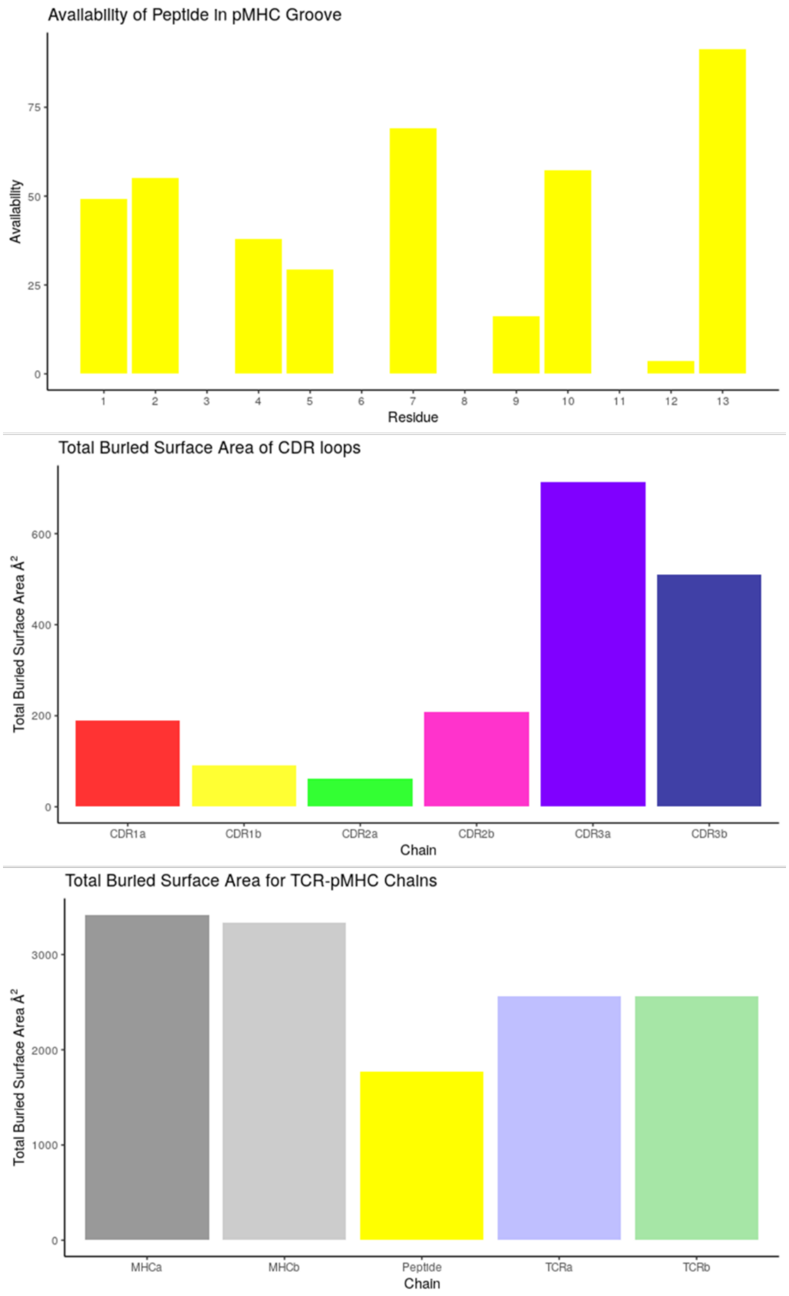


Figure 20 Three examples of buried surface area (BSA) at varying degrees of information. Visualization of BSA, available surface area (ASA) and availability (fraction of total surface area not buried). Top: Total availability of individual peptide residues for the TCR based on how buried the peptide is in the MHC. Middle: Overall BSA of each CDR loop, that is, how well each CDR loop is interfacing with the pMHC. Bottom: total BSA of each chain in the TCR-pMHC complex.

A plot is generated for the availability of each residue in the peptide in relation to the TCR. This gives an indication of where the anchor residues are in the peptide, as they will not be available to the TCR given that they are buried within the MHC. As well as this, the tabulated data is passed into shiny R, where the user can view the BSA, ASA or availability on a residue-by-residue basis, for each CDR loop, for each chain or for the whole complex. This multilevelled analysis is performed by reformatting the data in R using the dplyr verb system upon user input. An example of these different levels of analysis are shown above in Figure 20.

4.3.5 Shape complementarity

SC is a geometric value describing the "goodness of fit" between a concave and convex surface. High SC values indicate that there is a large interface between the TCR and pMHC, whereas a low SC indicates that some parts of the TCR and pMHC may not be accessible to one another. SC is calculated by passing a cleaned PDB file and its chain information into the SC programme[110]. The output file is then parsed leaving a single value between 0 and 1, where 0 is no fit at all and 1 is a fully complementary fit.

4.3.6 Contact analysis

The ligand binding action of the TCR and pMHC can be inferred by quantifying the number of non-covalent intermolecular contacts. A disparity in the number of contacts between regions (e.g. CDR loops) can be useful in describing the mode of binding in the TCR-pMHC complex. The contact distance between atoms is calculated using NCONT[104]. STACEI determines a contact as any two pairs of atoms being within 4Å, although this can be modified by the user. Contacts determined by STACEI are then annotated as one of three bond types: hydrogen bond (HB), salt-bridge (SB) and van der Waal (vdW) interaction.

HBs are determined by having a backbone carbon or side chain hydrogen donor within 3.4Å of a backbone amide or hydrogen acceptor side chain. The side chain donors and acceptors are described by the IMGT[111]. SBs are also annotated as contacts within 3.4Å where a glutamine or aspartic acid carboxyl carbon pairs with a lysine or arginine amino atom. The remaining contacts that do not meet the HB or SB criteria are annotated as a vDW. The exact list of donors and acceptors is described in Table 4.

4.3.7 Contact tables

Atom level contacts are concatenated into residue level contacts and are then further collected to chain (TCRα, TCRβ, MHCα, MHCβ/β2M & peptide) and sub-chain (CDR1α, MHCα1 etc.) components by integrating the annotations derived earlier. These groupings allow the quantification of contacts by each component and their relative contribution to ligand engagement. The tables are exported into an interactive shiny R output that allows for

Table 4 Contact annotation criteria for TCR-pMHC and p-MHC interactions: atom IDs using the IUPAC nomenclature for amino acids used by PDB [112]*. All contacts that meet the criteria for HB or SB but not the distance cut-off is annotated as vdW providing they meet the vdW cut-off criterium.*

	Donor atoms	Acceptor atoms	Distance cut-off (Å)
Hydrogen Bonds (HB)			
Backbone			**< 3.4**
All	**N**	**O**	
Side chain			
Arginine	**N**ε, **N**H1 (2), **N**H2 (2)		
Asparagine	**N**δ2 (2)	**O**δ1 (2)	
Aspartic acid		**O**δ1 (2), **O**δ2 (2)	
Glutamine	**N**ε2 (2)	**O**ε1 (2)	
Glutamic acid		Oε1 (2), Oε2 (2)	
Histidine	**N**δ1, **N**ε2	**N**δ1, **N**ε2	
Lysine	**N**ζ (3)		
Serine	**O**γ	**O**γ (2)	
Threonine	**O**γ1	**O**γ1 (2)	
Tryptophan	**N**ε1		
Tyrosine	**O**H	**O**H	
Salt Bridges (SB)			
Side chain			**< 3.4**
Aspartic acid		**O**δ1 (2), **O**δ2 (2)	
Glutamic acid		**O**ε1 (2), **O**ε2 (2)	
Lysine	**N**ζ (3)		
Arginine	**N**H1 (2), **N**H2 (2)		
van der Waals (vdW)			
Backbone			**< 4.0**
All	All	**All**	
Side chain			
All	All	**All**	

filtering, sorting, and exportation into a file format of the user’s choosing. The tables are presented interactively using the DT package which interacts with the JavaScript library DataTables.

4.3.8 Visual representations of contacts

These contact tables generated by STACEI are then used to create downstream analyses. These analyses can be on the residue, chain or CDR loop level. These plots are important for investigating the binding mode between TCR and pMHC.

4.3.8.1 Contact bar plots

A simple and frequently used representation of contact data is a bar plot of the number of contacts made by a given residue, chain or sub-chain. There are a large number of combinations which can be calculated when considering that the data can be viewed as between different residues, CDR loops or whole chains. In order to account for this the R shiny package is again used to provide an interactive platform for users to customise their data.

The user can toggle which chains are to be included in the plot (from the 5 of TCRα, TCRβ, MHCα, MHCβ/β2M and peptide), as well as being able to facet the data into separate subplots (e.g. a subplot of just TCRα contacts, followed by a second subplot for TCRβ) contacts as well as colouring the bars either by the contact force as determined previously, as well as the donor and acceptor chain.

4.3.8.2 Static plots

Sankey plots and Circos plots are both generated in R, using the NetworkD3 (https://christophergandrud.github.io/networkD3/) and Circlize (https://jokergoo.github.io/circlize_book/book/) packages, respectively. The Circos plot represents the number of contacts between complexes. There are two plots produced, one showing contacts between the 5 chains of the TCR-pMHC complex and one showing the same but sub-grouped by each CDR loop. The latter is shown in Figure 21. The Sankey plot shows a similar flow of information but includes the chains and sub-chains in one plot as a flow between them. Finally, a pie chart showing the contribution of each CDR loop to contacts with the peptide, MHC and pMHC is also generated.

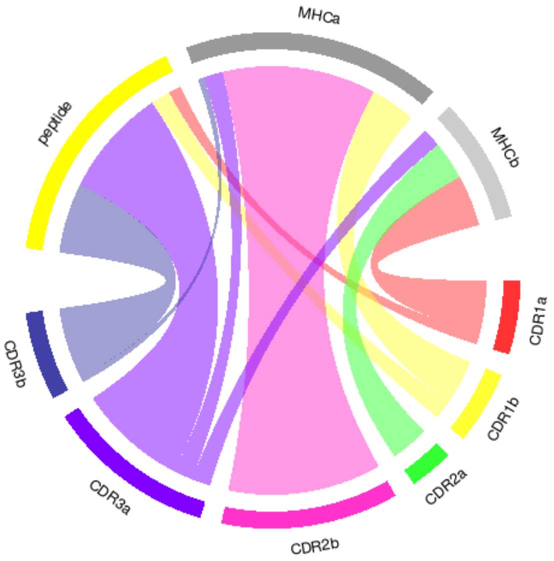


Figure 21 An example of a circos plot including the CDR loops

4.3.8.3 Contact maps

Contact maps are plots that show the peptide sequence and a TCR sequence with lines drawn between the two representing contacts. They are produced using the *networkx* (https://networkx.github.io/documentation/stable/) package. Each letter of a chain is encoded into node in a *networkx* graph, the contact data is then parsed such that any contact between two residues defines an edge in the graph. The edges are then drawn in the package, where each force is represented by a different colour. An example is shown in Figure 22. These plots are creating across all iterations of the different CDR loops and acceptor chain regions.

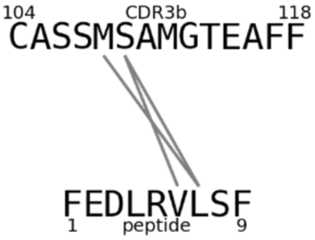


Figure 22 A contact map between the CDR3β and peptide.

4.3.9 Crossing angle calculations

In most TCR-pMHC structures, the TCR interacts with the pMHC with a relatively conserved action, wherein the TCR engages across the pMHC surface on a diagonal[113]. However, examples exist that challenge this “canonical” binding engagement[114]. It is not known whether this reverse polarity enables coreceptor binding and efficient TCR-mediated signal transduction and it could be a non-functional anomaly. STACEI calculates three angles which describe this docking orientation.

The crossing angle, described first by Rudoph[11] is the angle in 3 dimensional Euclidian space between the TCR axis and the MHC binding groove. The TCR axis consists of a vector between the averaged coordinates of the conserved cysteine residues at positions 23 and 104 of the TCRα and the corresponding coordinates for the same residues on the TCRβ. The MHC axis is then determined as the line of best fit through the Cα atoms of the binding groove. In MHC class I these are residues 50 to 86 and 140 to 176 in the α subunit and in MHC class II it is residues 46 to 78 of the α subunit and residues 54 to 64 and 67 to 91 of the β subunit. The *numpy* library least squares function is used to generate lines of best fit for the two axes. The crossing angle is then formally described:

$$\theta_{crossing} = arccosine\left(\frac{M\acute{H}C \cdot TC\acute{R}\overrightarrow{\beta\alpha}}{|M\acute{H}C| \cdot \left|TC\acute{R}\overrightarrow{\beta\alpha}\right|}\right)$$

$\theta_{crossing}$ reports the lowest angle between each of the vectors. The direction or polarity of the engagement is also reported, meaning that the user receives information of whether the TCR is binding the MHC in the the range 0-180° or 180-360°.

As well as the angle in 3D space ($\theta_{crossing}$) STACEI also reports two angles in 2D space. These are the rotation of the TCRαβ in relation to the face of the pMHC and the pitch or “tilt” of the TCRαβ away from the pMHC. These two angles (herein referred to as rotation and tilt, respectively) both contribute to the overall 3D angle.

This methodology first generates a plane of best fit through the MHC binding groove as previously. The rotation angle is calculated by orthogonally projecting the TCRα, TCRβ and pMHC axes onto the pMHC plane. The angle of the TCR projections (TCRα$_{proj}$ and TCRβ$_{proj}$) can then be calculated and from this the rotation angle is calculated:

$$\theta_{rotation} = arccosine\left(\frac{\overrightarrow{MH\acute{C}_{proj}} \cdot TCR\overrightarrow{\acute{\beta}_{proj}\alpha_{proj}}}{\left|\overrightarrow{MH\acute{C}_{proj}}\right| \cdot \left|TCR\overrightarrow{\acute{\beta}_{proj}\alpha_{proj}}\right|}\right)$$

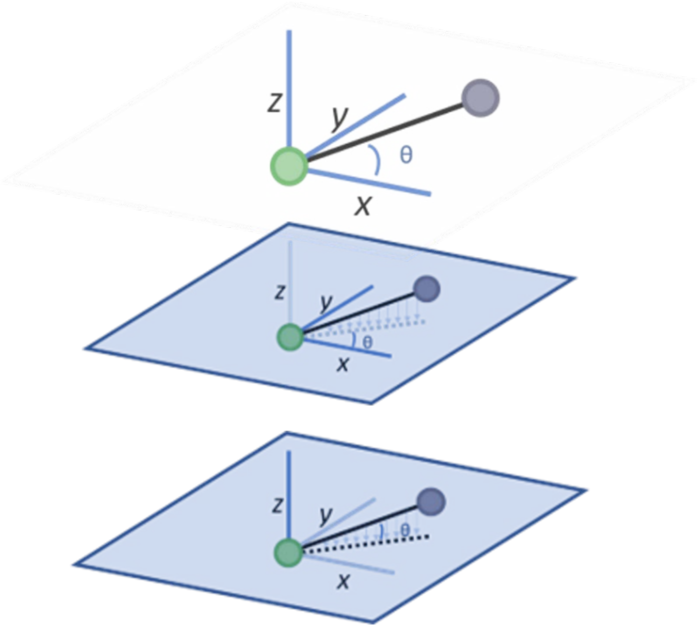


Figure 23 Overview of the 3 binding angles enforced in STACEI: crossing angle (top), rotation angle (middle) and tilt angle (bottom).

To calculate the tilt, the TCRα and β vectors are shifted across the project vector so that the coordinates of TCRβ are now in place along TCRβ*proj* while the TCRα remains in place. The tilt angle is then calculated as the angle between the TCR plane and the TCR projection:

$$\theta_{tilt} = arccosine\left(\frac{TCR\overrightarrow{B_{proj}A_{proj}} \cdot TCR\overrightarrow{BA}}{\left|TCR\overrightarrow{B_{proj}A_{proj}}\right| \cdot \left|TCR\overrightarrow{BA}\right|}\right)$$

Tilt angles are reported such that TCRs which tilt “up” away from the plane have negative tilt values.

4.3.10 3D structure visualisation

STACEI also aims to provide publication-ready visualisations of TCR-pMHC complexes. In order to do so, STACEI makes use of PyMOL to generate images of the 3D structure of the TCR. STACEI has a standardised colour system for each chain of the TCR-pMHC complex, as well as the CDR loops. This is the same as the colours used in generating the plots above. When STACEI generates these images, the TCR-pMHC being analysed is aligned against a reference structure, which means that every image is being viewed from the same position. This allows users to superimpose images of the same complex or different complexes in a way that is conducive of meaningful comparison. An example of these visualisations are found in Figure 24, which shows three visualisations of crossing angle.

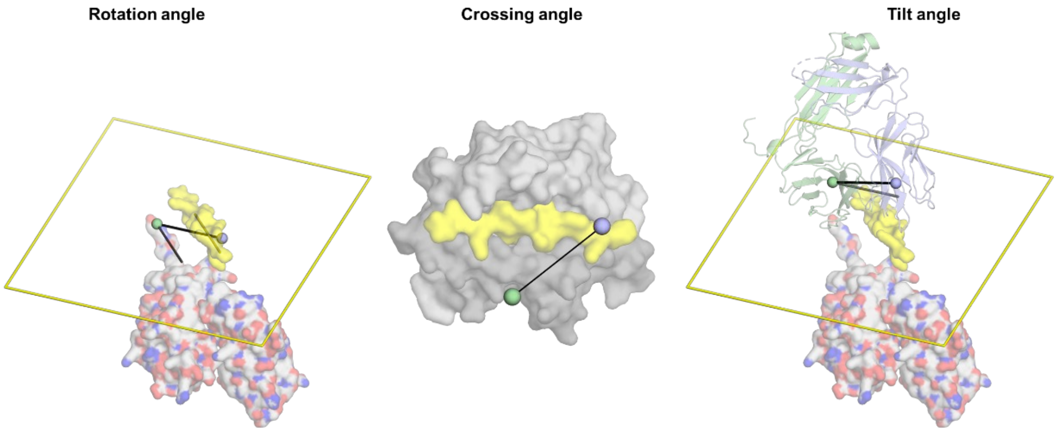


Figure 24 Samples of visualisations of crossing angles in PyMol. Left: Rotation angle across the Y and Z plane. Middle: Crossing angle in X, Y and Z plane. Right: Tilt angle in X and Y plane.

As well as publication ready images, the images are stored as PyMol session files for easy customisation.

4.3.11 pMHC-centric analyses

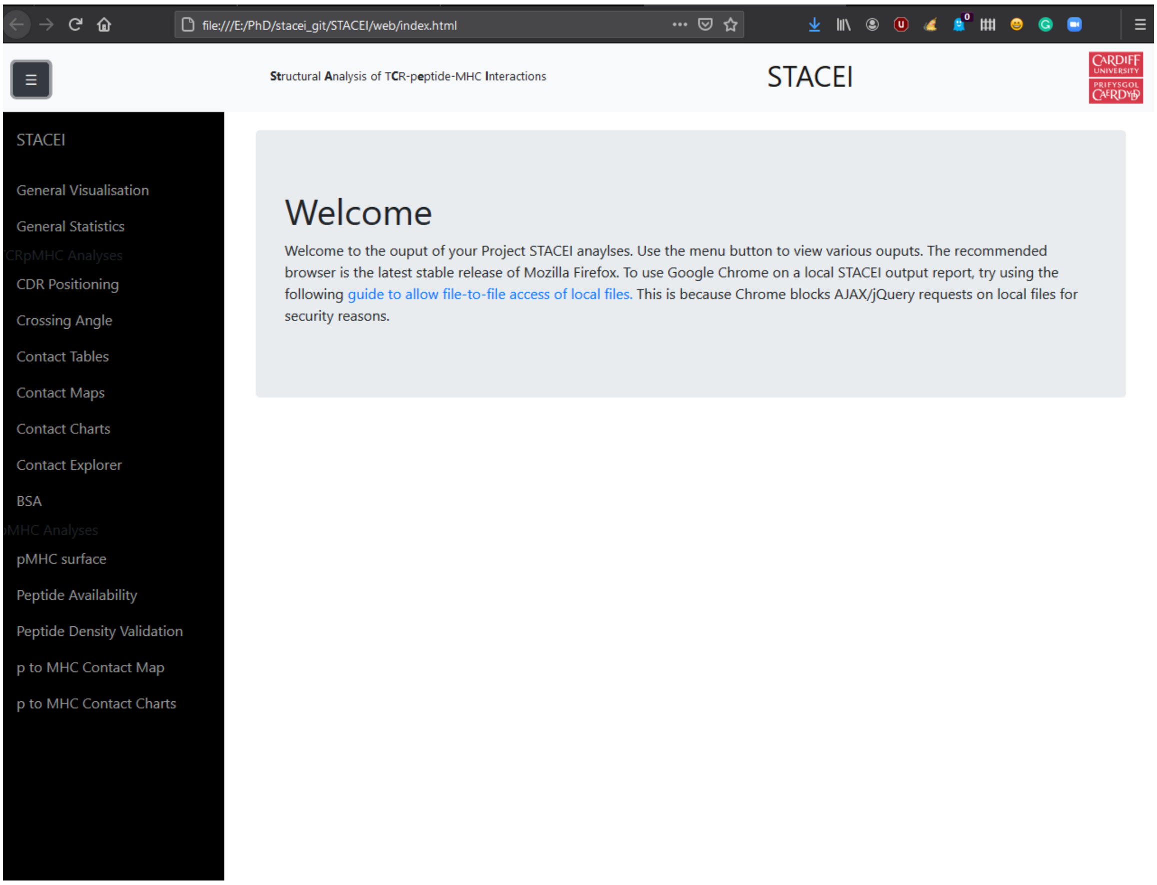
In addition to analysing the TCR to pMHC interface, STACEI also performs analyses on the pMHC component. Namely, electron density maps are calculated which allow the validation of structural data around modelled peptide residues. Electron density maps are created by using an MTZ file. The user can supply their own MTZ file, or if the name of their file matches that of a structure in the PDB then the matching MTZ file is downloaded from the PDB.

Electrostatic (*APBS*) analysis is performed to calculate the surface charge of the pMHC being presented to the TCR. This is performed by removing the peptide from the structure, running a single cycle of refinement in recmac5. This refinement is then transformed by a Fast Fourier Transform (FFT) using the fft package in CCP4 before

Finally, contacts made between peptide and MHC are calculated as are the BSA, ASA and availability values as described previously, and displayed as network maps, tables and charts giving an insight into the anchoring of peptide ligands into the MHC groove.

4.3.12 Output file

As there is a large number of files outputted from a run of STACEI, the tool outputs everything in a directory structure. A HTML document (Figure 25) is written to incorporate these outputs in a single document that can be opened in a web browser. The interactive plots from Shiny are also embedded in these documents.



Welcome

Welcome to the ouput of your Project STACEI anaylses. Use the menu button to view various ouputs. The recommended browser is the latest stable release of Mozilla Firefox. To use Google Chrome on a local STACEI output report, try using the following guide to allow file-to-file access of local files. This is because Chrome blocks AJAX/jQuery requests on local files for security reasons.

Figure 25 the STACEI landing page. For every structure analysed the output is collated into a single directory. A HTML file is then generated to point to these local files to let the user explore their results in a structured and easy to interpret manner.

4.3.13 Availability and license

STACEI is available at http://github.com/whalleyt/stacei. The python package and all its child dependencies are able to be installed using the setup tools (https://pypi.org/project/setuptools/) package. ANARCI, CCP4 and PyMol are all open sourced and free to download but must have their license agreed to separately. This means the user must download them themselves to be included in the package. Failing that, a Dockerfile is provided so that the user can that instead of installing other softwares. The Dockerfile is available from https://hub.docker.com/repository/docker/twhalley93/stacei. The tool is licensed under the GNU GPL.

4.4 REVIEWING THE PROTEIN DATABANK

Of the 135 MHC class I restricted structures and 51 MHC class II structures, there were 120 and 45 valid structures, respectively. The excluded TCR-pMHC structures contained pMHC complexes where the peptide was linked to the MHC chain artificially. As there is no way of knowing for certain where the MHC ends and where the peptide begins without a priori knowledge these are excluded from the analysis.

These 186 structures (IDs listed in supplementary table S1) were then used to conduct a review of TCR-pMHCs in the PDB. The sake of this was twofold, first to assure the validity of the outputs of STACEI. Structures of TCR-pMHCs are very heterogenous, with variation in all chains in the complex, along with different extra pieces of information to deal with, such as the insertions, deletions, mis-numberings, multiple asymmetric units in the PDB file and different metadata. Unlike RNA and DNA sequencing data, where there exists several tools to artificially generate data for testing (for example seqgendiff[115] and Polyester[116] for RNA-seq and BEAR[117] and grinder[118] for genome sequencing) there is no such tool for reliably generating PDB files. Similarly, while there are tools for generating artificial or in silico immune repertoire sequencing data (most recently, immuneSIM[119]) to generate TCR sequences, there is no way to meaningfully have these as protein structures. The absence of simulated TCR structures means that the only viable option for testing the implementation of STACEI was to run it on all available structures. The second rationale was to expediate and standardise the reviewing process for αβ TCR-pMHC structures. At the time of writing, prominent reviews[11,19] will have been performed by manually annotating the structures. This is both time consuming, potentially error-prone, and self-limiting. Therefore, STACEI was run on each of the 186 structures in the PDB and outputs were collected in an automated fashion used to generate a report of all known publicly available TCR-pMHC structures. The outcome was used to elucidate the molecular “rules” for TCR-pMHC binding which could then be applied in numerous fields such as vaccine design, immune response and cancer biology.

5 RESULTS: STACEI- A TOOL FOR THE STRUCTURAL ANALYSIS OF TCR-PMHC INTERACTIONS

5.1 VALIDATING THE OUTPUTS OF STACEI

Prior to reviewing the structures in the PDB, the primary goal of running STACEI on all the structures was to gain an insight into the performance of STACEI. This meant that each structure was checked first for a valid exit status. Then each image was inspected to check that it was informative and was self-explanatory. For images of the 3D structure particular attention was paid to how the structure was aligned such that the images were consistent across structures. Contacts and buried surface area were validated by manually performing runs of NCONT and PISA respectively, likewise for calculation of SC in the eponymously named SC programme.

5.2 ERROR HANDLING

In response to some errors in handling incorrect or unmanageable structures, a number of exit conditions were added to STACEI to allow for a descriptive error to be handled on certain conditions. The first and probably most common error was in the initial annotation and chain determination step. The tool exits and removes all intermediary files if:

1. there are less than five chains in the PDB file; the tool cannot find at least one TCRα and one TCRβ using ANARCI.
2. the tool cannot find one MHCα and a β2M/MHCβ. This is determined if there is a chain sequence with a score of greater than 50 using the *swalign* Smith-Waterman alignment score.
3. The tool does not identify a peptide, determined as being a non-TCR and non-MHC complex by failing the assessments points 1 and 2. A peptide is only considered if it is less than 50 amino acids long. This is because some structures contain a superantigen which may pass other criteria
4. There must be at least one pair of TCRα and TCRβ determined by the TCR's CDR loops being within 22Å.
5. There must be at least one MHCα and a β2M/MHCβ where their centre of masses are within 22Å, also.

6. There must be one peptide making contacts the MHC complex. This pMHC complex must then have contacts being made with it by one TCR complex

If none of these rules fail to be satisfied, then STACEI will exit. Other exit statuses that occur later include if the TCRs fail to have a V and J gene annotation by ANARCI. STACEI also warns users about potential incorrect annotations. In testing, ANARCI was shown to mislabel some TCRα chains as TCRδ chains. This is permissible as the numbering remains correct so instead of exiting, the tool will instead warn the user but carry on with its execution. Finally, some structures are not permissible to be ran in SC because of "imaginary contains" in the input PBD file. Imaginary contains have been associated with hydrogen atoms being in the PDB file but at the time of writing this is still considered a bug and the exact explanation as to the cause remains unknown.

5.3 REVIEWING THE PROTEIN DATABANK

5.3.1 Rationale

The first rationale in reviewing the PDB was to first provide enough data to successfully validate the tool as in section 5.1. The second was to demonstrate the utility of STACEI in being able to analyse a wide number of PDB files quickly and effectively. Other reviewing efforts were either automated but surface level[120,121], aimed to be a reference of all structures rather than an analysis; or in depth but time consuming, with most analysis being manually performed[11,19].

5.3.2 Data collection

A total of 120 MHC class I restricted and 45 MHC class II restricted TCR-pMHC structures were collected using the web scraper described in 4.2. This was then ran iteratively in STACEI using the following command:

```
STACEI -F <pdb_file> -O outputs/<pdb_file> -R
```

This input means that chains were determined automatically by STACEI and required no prior knowledge from the user. Hydrogen bonds van der Waals forces and salt bridges were defined as the default values as discussed in 4.3.6.

5.3.3 Overview of data

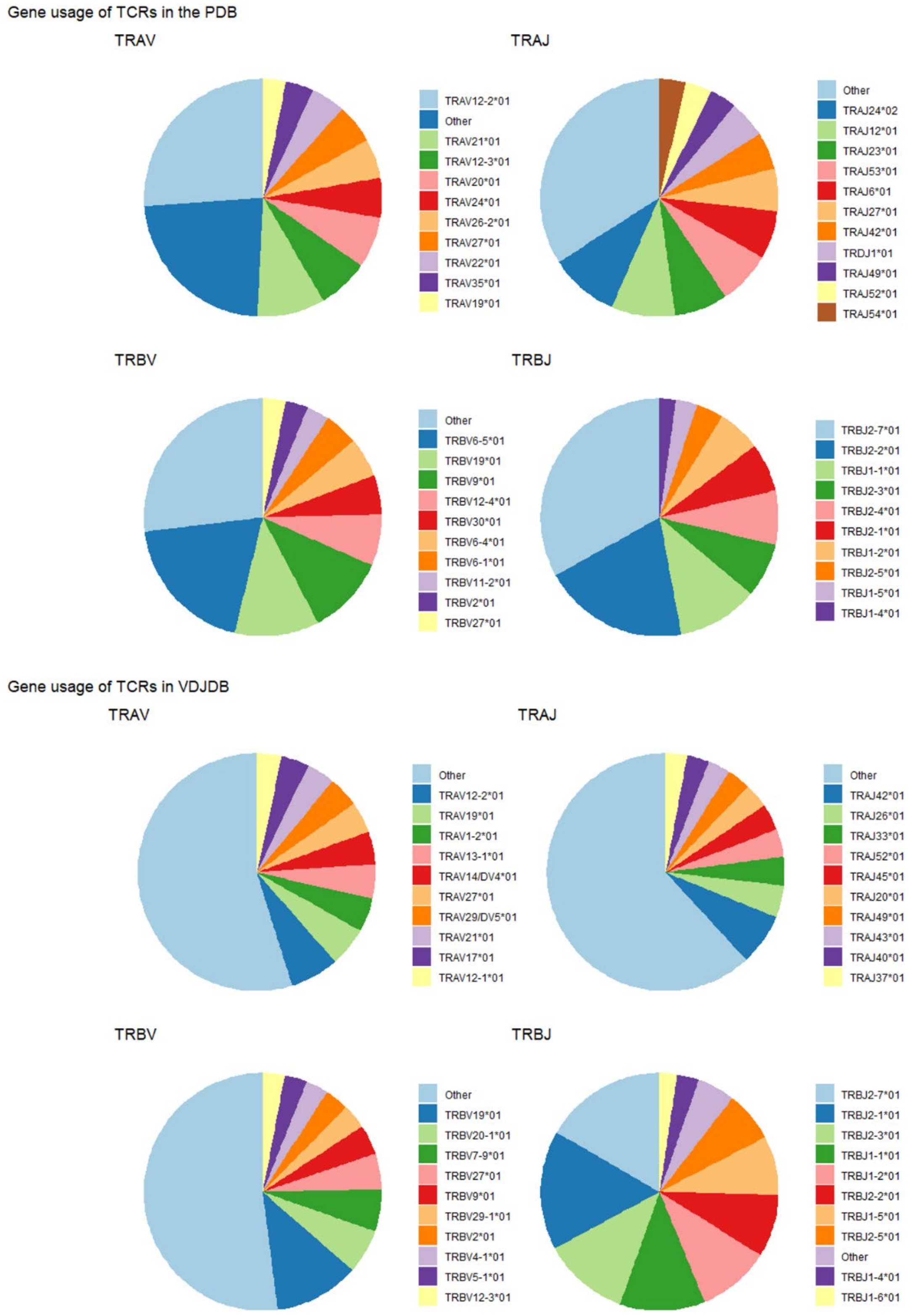


Figure 26 The V and J gene usage of TCRs in the PDB analysed by STACEI and sequences extracted from VDJDB.

Before any full exploratory analysis the V and J gene usage of all chains of the TCRs were checked and compared against the sequence in the VDJDB[50]. The rationale here was that the number of protein structures of TCRs means that there is more likely to be a bias in V and J gene usage, both because some TCRs may be easier to crystallise and because there is inherently a smaller pool of TCRs to select from. For a simple comparison, the top 10 genes were counted for TRAV, TRAJ, TRBV and TRBJ in both the PBD files and the sequencing data VDJDB as shown in Figure 26.

Of the top 10 genes in the PDB and VDJB, 4 of the same gene was shared between the two TRAV; 3 of the same gene was shared between the two TRAJ genes; 4 were shared in the TRBV gene and 9 were shared between the two TRBJ genes suggesting that there is indeed a bias in the PDB assuming that there is less bias in the VDJDB.

5.3.4 Contacts

Contacts were calculated as any atoms between the donor and acceptor chains within 4Å.

5.3.4.1 Global TCR and pMHC contacts

The first assessment made was the total number of contacts made by the TCR to the pMHC and then between the peptide and MHC. These are displayed as histograms below in Figure 27. It shows that there both MHC class I and II complexes make a wide range of contacts. In peptide to MHC contacts MHC class II complexes make on average more contacts with the peptide than MHC class I (median 57 contacts vs 41 contacts). This is perhaps unsurprising given the longer length of MHC class II restricted peptides.

The behaviour of the TCR contacting the pMHC is much closer, with both median number of contacts being made (42 contacts in MHC class I and 40.5 in MHC class II).

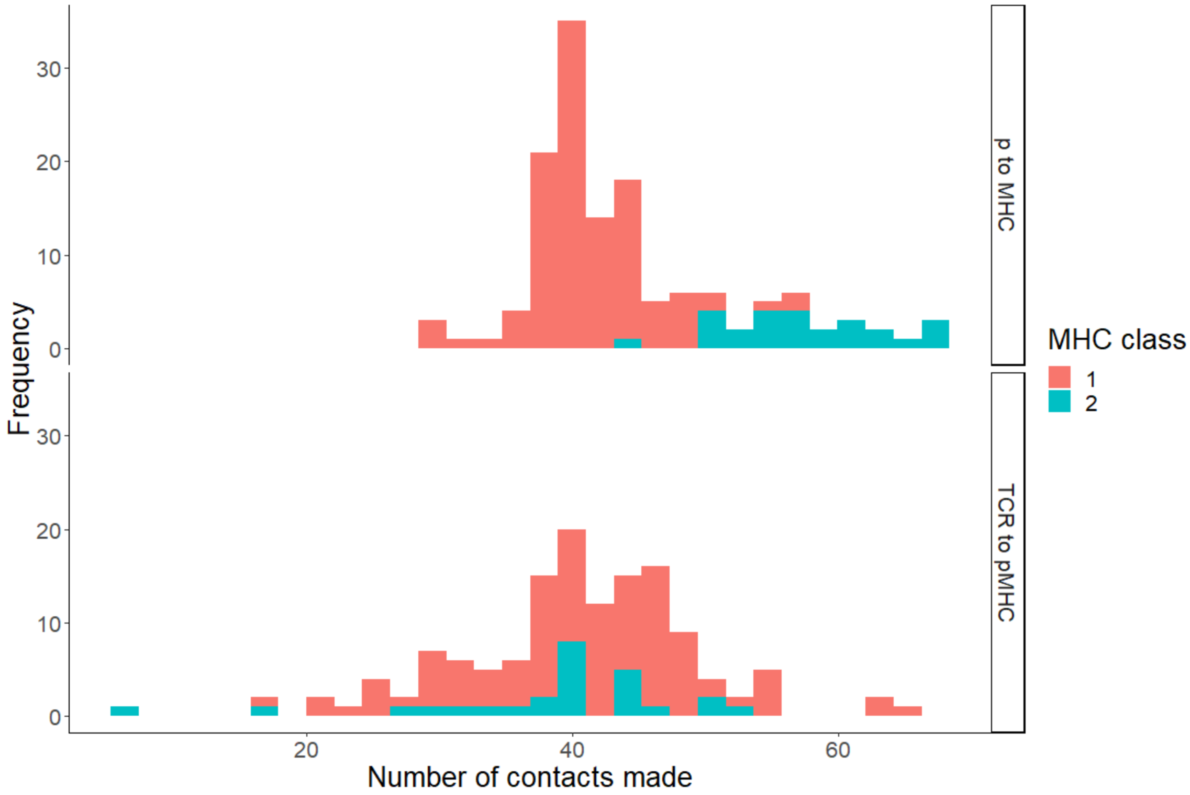


Figure 27 histogram of the number of contacts made by the interface between the TCR and pMHC (bottom) and the peptide and MHC (top).

5.3.4.2 Contacts made by CDR loops

In order to decipher whether there was a difference between the overall behaviour of TCR-pMHC contacts between MHC class I and II, the contacts were further broken down by individual CDR The difference in contacts were then compared using an unpaired t-test with p values ≤ 0.05 being considered significant (Figure 28). Using this criterion, the CDR2α, CDR2β, CDR3α and FWα region all made significantly different amounts of contacts between MHC class I and class II structures. Again, like the number of contacts between peptide and MHC, this might be seen as unsurprising as the CDR1 and 2 preferentially engage the largely conserved MHC, then the contacts may be different between the two major classes of MHC, with little variance within these classes as engagement of CDR2 has been shown to be energetically inessential to the formation of the TCR-pMHC complex[20,122]. More surprising is that the CDR3α makes a significantly different number of contacts to the pMHC while CDR3β does not.

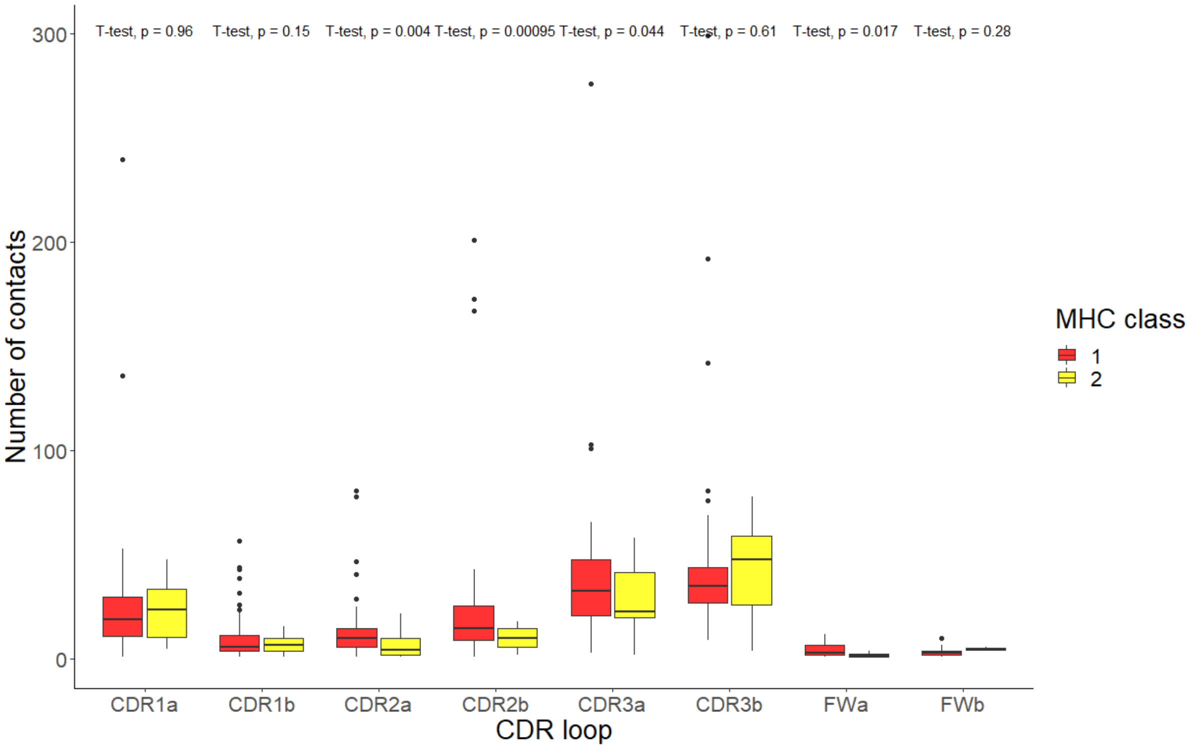


Figure 28 number of contacts made by each CDR loop to the pMHC for both MHC class I and class II restricted complexes. MHC class I and class II CDR loops were compared using an unpaired t-test.

5.3.4.3 Describing the difference between MHC I and II complexes

Figure 28 shows that there are differences in the binding patterns between the TCR-pMHC of MHC class I and II restricted structures, but not in such a way that it that can easily be explained. Numerous machine learning approaches can be used to predict categorical variables (in this case the class of MHC) Many of these methods for predicting categorical data are so-called “black boxes” and do not help understand the underlying methods. However, decision trees have proven to be helpful in both prediction and mechanistic understanding.

Decision trees are a non-parametric statistical method that selects the important features used to discriminate between two or more categorical variables. The algorithm creates a "tree" of decisions where a top down search of "questions" that help distinguish between the two datasets are generated. These decisions are measured by their Gini index, a measure of the degree of a variable being wrongly classified when randomly selected, where 0 denotes that all predictions are to a certain class and 1 is where the elements are randomly distributed across the classes.

The number of contacts between each CDR loop and each domain of the pMHC were collated from STACEI’s output and read into columnar data. The MHCα2 and MHCβ1 domains were masked to be the same functional domain (otherwise the algorithm could predict the difference based on the presence of absence of this label). The domains used were MHCα1, MHCα2/ MHCβ1 and peptide. As there are 8 CDR loops (including framework regions) this led to there being 24 different variables used in the decision tree. A decision tree was built using the rpart package in R (https://cran.r-project.org/web/packages/rpart/).

The data was split into a training set of 80% of the data and a test set of 20% of the data to avoid overfitting. The decision tree, shown in Figure 29 demonstrates that the discriminating factors

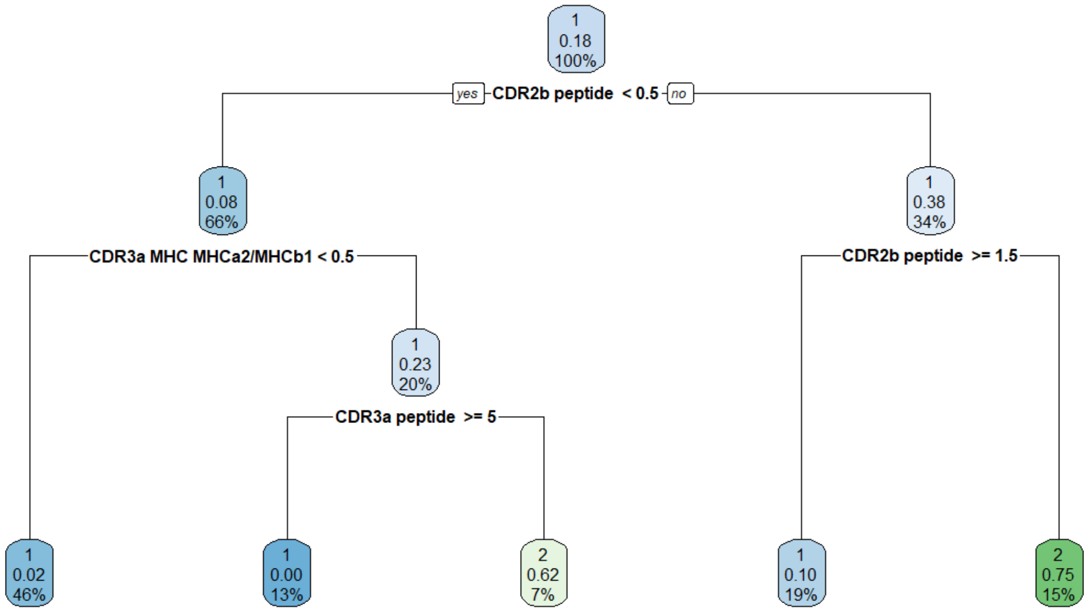


Figure 29 decision tree to predict MHC class. The roots of each tree show the "question" asked of each structure and the branches show the decision made.

between MHC class I and II are the number of contacts made by the CDR2β to the peptide, with MHC class II making more contacts on average; and the CDR3a making more contacts to the peptide and less to the MHC in MHC class I than MHC class II. This model had an accuracy of 85.7%.

5.3.5 Shape complementarity

As discussed in the methods, the SC is a measure of the "goodness" of fit between two complexes, originally discussed in the field of antibodies. Shown below is the distribution of

The relationship between SC and number of contacts in Figure 30 made is weak (Spearman's correlation 0.56; Pearson's correlation 0.59) suggesting that the number of contacts made is not necessarily necessitated by the "goodness of fit" of the overall interface of the entire pair of complexes, suggesting that there are certain "hotspots" that contribute to TCR-pMHC SC values, ranging from 0.44 to 0.81.

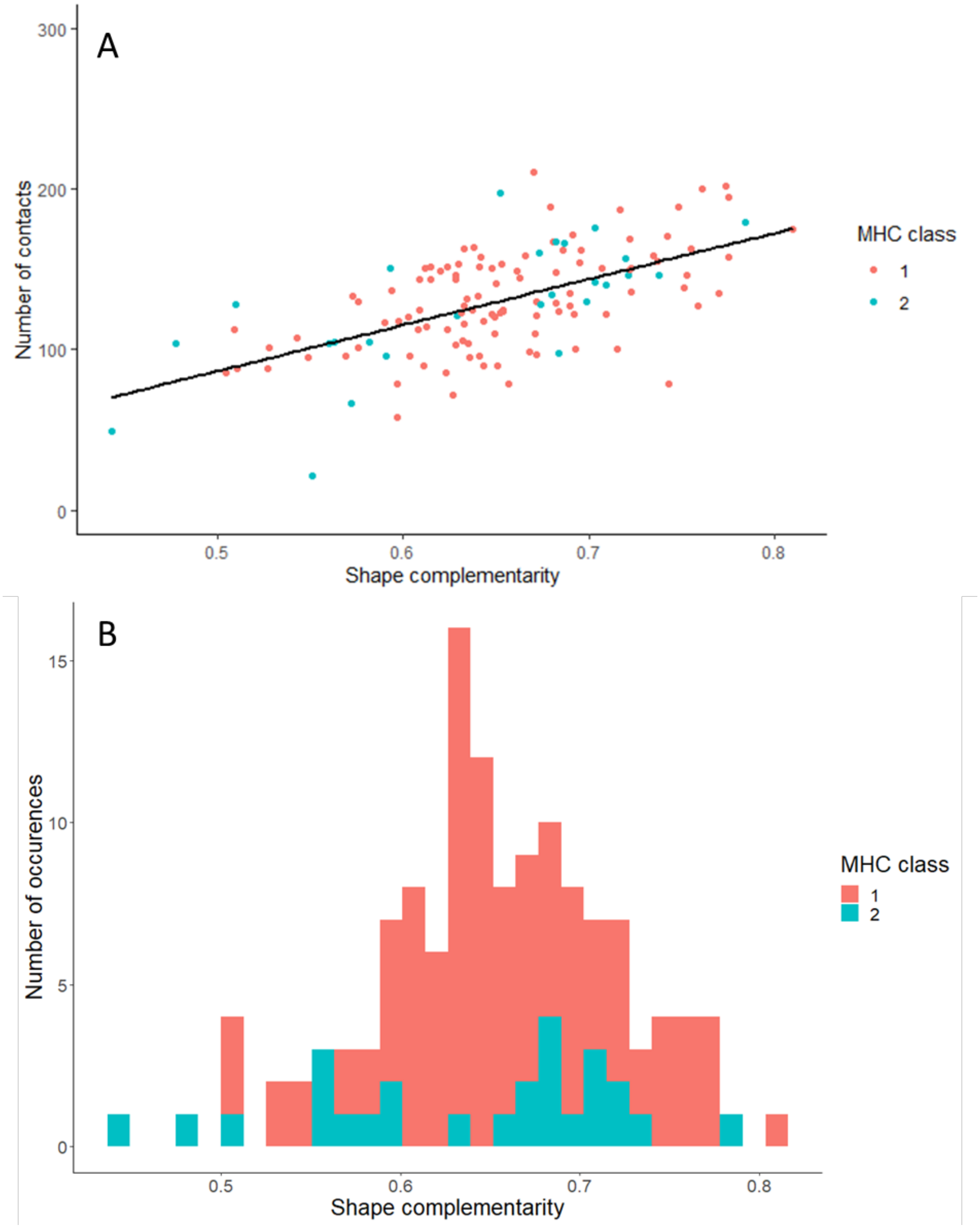


Figure 30 A) The relationship between shape complementarity and number of contacts. While there is a relatively strong correlation (Spearman's correlation = 0.56; Pearson's correlation = 0.59) the number of contacts do not fully explain the SC. B) Histogram of the SC values for MHC class I and II TCR-pMHC complexes.

interaction regardless of how the overall conformation of the structure is. This could also be the contribution of non-CDR loop residues which are not important in pMHC binding bringing the average SC value down.

5.3.6 Buried surface area

The BSA is often measuring the same parameter as contacts, meaning the analysis is somewhat duplicated. One interesting measure however is the “availability” of each residue of the peptide to the TCR when buried in the MHC.

As an example, in MHC class I restricted complexes (Figure 31, top) peptides of length 8 shows that the anchor residues are at positions 2/3 on the C terminus as they are buried within the MHC and at position 8 at the N terminus. When comparing peptides of length 8 to length the average availability of the central residues is much higher. For example, residues 7,8,9 of the 13-mer peptides all have availabilities >90% compared to none being that high in an 8-mer peptide. This demonstrates the “bulging” effect of peptide length in MHC class I restricted peptides. The picture is not as clear in MHC class II. Although one can still identify the anchor residues (e.g. 5 and 10 for 11-mers) there is a lot more overlap in the availability of some residues, possibly due to the lack of samples.

5.3.7 Physiochemical parameters

The R package Alakazam was used to calculate physiochemical properties of the CDR loops of the TCR. The physiochemical parameters were then correlated against the number of contacts made by each CDR loop, however no significant results were returned.

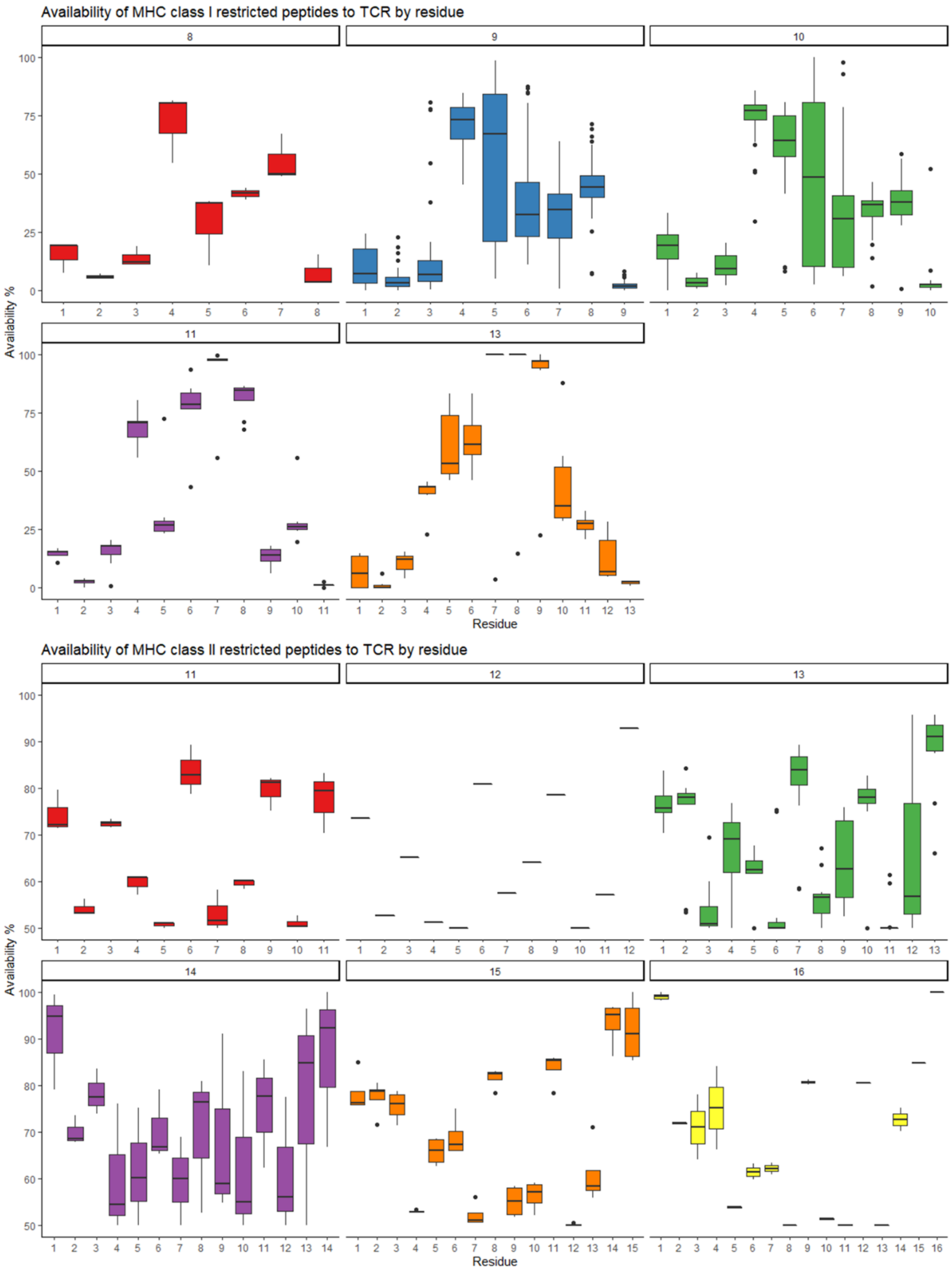


Figure 31 The availability of each residue of the peptide from each TCR-pMHC complex for MHC class I and II

5.3.8 Crossing angle

As shown in Figure 32 most crossing angles fall below 90 degrees, apart from 4 MHC class II structures (4gg6, 4z7u, 5ks9 and 5ksa) and 1 MHC class I structure (4qrp). Whilst this is not enough to say categorically, it may suggest that the more extreme crossing angles tend to be made by MHC class II restricted TCRs.

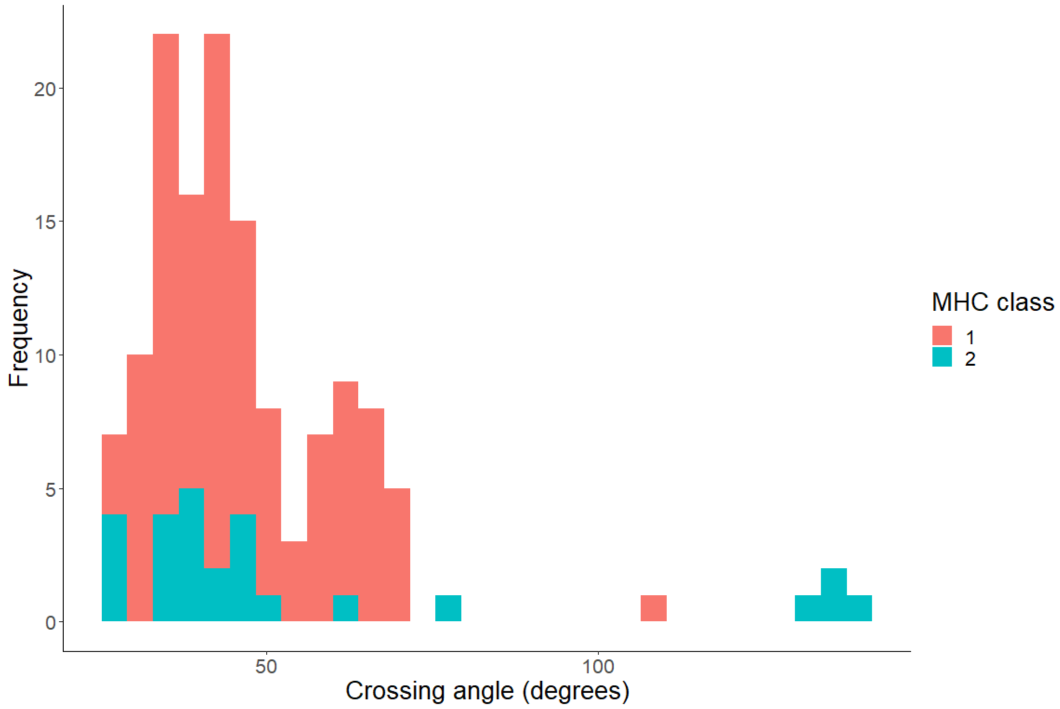


Figure 32 distribution of 3D crossing angle of MHC class I and II structures

There also appeared to be no relationship between rotation and tilt. Like the 3D crossing angle that the rotation and tilt combine to make up, most structures fall close to one another, with 4 MHC class II structures having extremely high rotation values (the same as the high crossing angles) and also 1 MHC class I structure (also the same as in the 3D crossing angle), suggesting that while these "extremes" are possible, they are rare.

5.4 PERFORMANCE AND RUNTIME

The average runtime for a ray-traced structure was 12 minutes and 30 seconds, this was taken from all the structures used in the PDB review. Very little optimisation was performed as the rate limiting step was rendering and ray-tracing images in PyMOL which was unavoidable.

6 BACKGROUND: APPLICATION OF PARALLEL COMPUTING TO CPL DRIVEN EPITOPE PREDICTION

6.1.1 CPL driven database scanning

As briefly introduced in 1.8.7 CPL scans are generated by fixing a single amino acid in one position and leaving the rest of the residues in the peptide a degenerate mixture of equal amounts of 19 proteogenic amino acids (cysteine is excluded from degenerate positions)[123]. In the context of TCR epitope discovery this is then repeated for each of the 20 amino acids in each of the residues of a peptide.

CPL scanning has a number of advantages, namely it allows for a large amount of peptides to be synthesised and tested; experiments can use non naturally occurring amino acids or D-amino acids[124] and experiments can be optimised following the initial experiments to narrow down candidates[123]. One of the major potential downsides to CPL screening is the assumption that each residue position contributes equally; the methodology also ignores the differences in solubility of peptides of different sequence in aqueous solution which presumably favour hydrophilic sequences at the expense of peptides with multiple hydrophobic residues.

Our group has previously released software for T-cell epitope discovery using the CPL approach[85]. An agonist likelihood (ALS or Λ) was derived:

$$\Lambda(\alpha_1, \alpha_2, \ldots, \alpha_n; i) = \sum_{p=1}^{n} \ln \frac{Y_p^{a_p}(i)}{\sum a' \in Y_p^{a'}(i)}$$

The ALS is the sum of each the natural logarithm of T-cell effector function value (MIP1β expression measured by Enzyme-Linked Immunosorbent Assay; ELISA) for a given amino acid at a given residue, normalised with respect to each fixed amino acid's MIP1β expression at that position. Rather than calculating this score for every possible peptide (for a 9-mer peptide this would require 20^9 calculations) a FASTA file of protein sequences is scanned, and each peptide is scored. This is quicker, if $<20^9$ peptides are scanned and gives biological context to the origin of the peptide.

6.1.2 Parallel programming

Biologists often need to make use of high-performance computing (HPC) environments for their analyses as many of them require long processing times and high memory or storage usage. The most common way of parallelising and speeding up software is central processing unit (CPU) based parallelisation through either shared memory multiprocessing like OpenMP[125] or distributed memory multiprocessing such as message passing interface (MPI)[126].

Parallel programming is the notion of a program running on more than one processor simultaneously. The algorithm run by the program is broken down into sub-components which can be run independently. These parallel threads can communicate through fixed means, such as mutexes and locks which are discussed below. These allow threads to pass information between them safely.

The exact means of parallelisation is a consequence of the type of algorithm being implemented, as well as the data type and hardware architecture being used. There are several paradigms through which this can be viewed. Perhaps the most common is the notion of task parallelism and data parallelism. Task parallelism breaks the algorithm into separate sub-algorithms and runs them in parallel before communicating the results between the different sub-jobs. Data parallelism is where the algorithm is kept whole, and the data is broken up before being operated on in parallel.

In most parallel programs, at some point there will be a time where the independent parallel processes need to communicate between each other. A simple example of this would be multiple threads running a computation before summing the results at the end. Without a safeguard, there is no guarantee that these threads will complete at the same time, meaning access to the results is variable and unstable. The most common way to combat temporal differences is to add a synchronization step where the program waits for all threads to finish before proceeding, shown in Figure 33. These conditions are sometimes called race conditions.

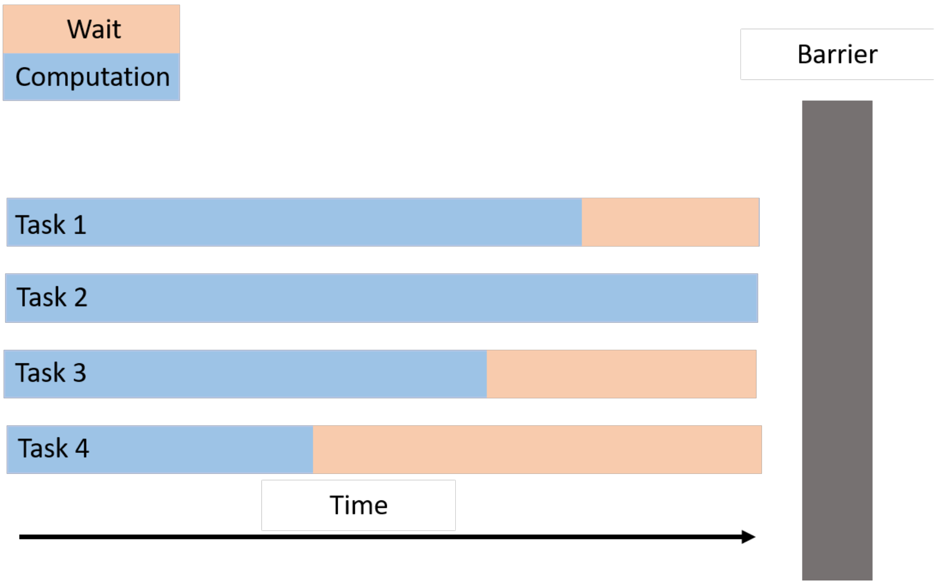


Figure 33 An example of a race condition. A barrier or synchronization step forces the quicker running tasks (tasks 1,2 and 4 in this case) to wait for the slowest thread (thread 4).

Another important concept in parallel computing is the concept of a mutual exclusion (mutex), sometimes referred to as a lock. Locks are synchronization methods that limit access of certain variables or memory locations. An example of when a lock is needed is where two parallel threads are trying to write to the same variable, e.g., summing several values into one variable. The lock ensures that only one thread can access this variable at a time to avoid variable clashes, this is visualised in Figure 34.

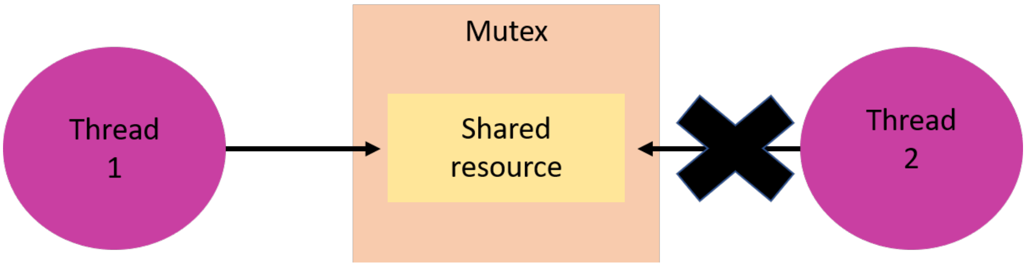


Figure 34 An example of the application of a mutex. Thread 1 and 2 both require access to the shared resource. The mutex allows only one thread access at a time, meaning thread 1 can interact with the resource but thread 2's access is limited.

6.1.3 CUDA

In recent years general-purpose computing on graphics processing units (GP-GPU) has come to the fore. Instead of using the CPU, like most parallel programming APIs, GP-GPU uses the graphics card or graphics processing unit (GPU) to perform computations.

The GPU's architecture in comparison to CPU is advantageous in certain processing contexts[127]. The primary design purpose of the GPU is to perform large quantities of very quick and simple operations across thousands of short-lived threads. Compared to the CPU, which has much lower multithreading capability this means the GPU excels at running simple computations. The architectural difference between GPUs and CPUs is shown in Figure 35. Typically, on a per thread basis, the time per operation (latency) is slower, but this is counter-acted by a greater throughput.

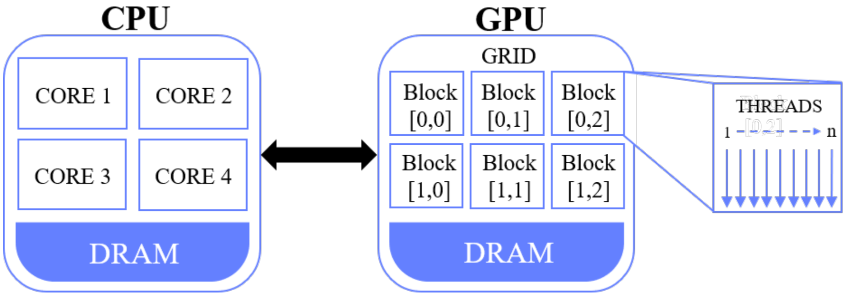


Figure 35 Architecture differences between GPUs and CPUs. CPUs (left) typically have a small number of cores with limited multithreading capabilities. In comparison GPUs (right) have a larger number of cores. These are arranged into blocks which represent a group of threads, which operate in 3D dimensions.

CUDA is an API developed by the GPU manufacturer NVIDIA. It is designed to work in a number of lower-level programming languages, mainly C, C++ and Fortran; with APIs that can be called from higher level languages such as MATLAB and Python. CUDA represents a set of libraries and functions paired with compiler directives which are then compiled with nvcc, NVIDIA’s C/C++ compiler (https://developer.nvidia.com/cuda-llvm-compiler). While other APIs such as OpenCL[128] exist that work on most GPU architectures, this comes at a cost of being lower level and harder to generalise. NVIDIA in contrast is only designed to work on NVIDIA GPUs. NVIDIA is arguably the most beneficial choice, as CUDA shows a marginal increase in performance [129] with better documentation.

As there is a consistency in the architecture of GPUs CUDA is designed to work with, CUDA programmes can be developed with a certain structure in mind. Each individual thread is grouped together to form a block, blocks are then combined to make grids. The exact arrangement of these grids and blocks varies from GPU to GPU. A CUDA function, or kernel, executes with two assumptions: that every thread will run with the exact same function and each thread has a unique ID that can be used to access the memory at that location.

By nature, blocks are required to execute independently, meaning that code can call any thread in any order. In contrast, threads in a block can cooperate by sharing data. Threads can also be synchronised to regulate memory access. This is particularly useful when two threads attempt to access the same memory location. Threads can exist in 3 dimensions within a single block, likewise blocks within grids also exist in 3 dimensions [130]. This means to get an individual thread ID the user must access the thread with respect to its index in 3D space like so:

$$int\ x = blockIdx.x * blockDim.x + threadIdx.x$$

CUDA programs also require memory to be allocated on the GPU prior to calling the kernel. Once memory has been allocated, the user must also copy variables or pointers to variables across onto the GPU. Once a kernel has completed, any variables on the device that need to be used most be copied back to the host. Memory allocation and the physical copying of data from the host to the device is an important caveat to the CUDA paradigm. The data transfer must be quick enough relative to the computation time to make the overhead worthwhile.

6.1.4 Applications of CUDA in bioinformatics

CUDA is most widely adopted in molecular dynamics, protein modelling and numerical optimisation [131] but in recent years there have been a great number of methods focussing on DNA, RNA and Protein sequence data. For example, there are a multitude of GP-GPU optimised methods for alignment. These can include novel algorithms[132] or reimplementation of existing algorithms like the Burrows-Wheeler transform (BWT)[133] and BLAST[134]. Outside of alignment it has been used to quickly scan protein databases in the context of HMMER [135,136].

7 METHODS: APPLICATION OF PARALLEL COMPUTING TO CPL DRIVEN EPITOPE PREDICTION

7.1 AIMS

The aim of this chapter was to increase the performance and expand the use cases of the existing PICPL tool using NVIDIA's CUDA API, a tool suite for GPU parallelising C or C++ code. The motivation for expanding the codebase to include parallel programming was because many of the potential uses of PICPL were not computationally feasible, e.g. scanning a very large database (e.g. all publicly deposited protein sequences totals ~350 million protein sequences). This would also mean that CPL scanning could be expanded into other applications, for example comparing peptides to all theoretical combinations.

7.2 IMPLEMENTATION

7.2.1 PICPL

7.2.1.1 Description of previous implementations

The pre-existing implementation of PICPL was a webserver (https://picpl-dev.arcca.cf.ac.uk) back-ended by a parallel MATLAB script. The script (via the web portal) took 3 user inputs, a valid CPL scan file consisted of ELISA measures of MIP1β release of a T-cell clone against each peptide mixture in a CPL with the top results returned as an input FASTA file. The script then iterates through each protein in the FASTA file, breaks it into peptides of the same length as that described by the CPL scan file and scores each peptide. Scoring is performed by adding to the user is then returned a tab-delimited file of the top n peptides, what protein they correspond to and their ALS score. To reiterate the scoring step as described in the introduction, the ALS is defined as:

$$\Lambda(\alpha_1, \alpha_2, \dots, \alpha_n; i) = \sum_{p=1}^{n} \ln \frac{Y_p^{a_p}(i)}{\sum a' \in Y_p^{a'}(i)}$$

The ALS is the sum of the MIP1β expression value for a given amino acid at a given residue, normalised with respect to each residue of the CPL scan and natural log transformed.

7.2.1.2 Conversion to serial C++

The initial step in writing a CUDA implementation of PICPL was to convert it into C++ code. While CUDA implementations exist embedded in higher level languages such as MATLAB and Python exist, they are limited in their scope and often to get optimal performance the code must be written from scratch in a lower level language like C or C++. Many of the simpler operations have like for like functions in both MATLAB and C++, so the basic structure of the code remained the same. However, many of the in-built operations of MATLAB centred around matrices do not have an equivalent. This means the scoring process was rewritten to access the CPL scores by matching to a map dictionary based on the amino and its position in the peptide. Also, the normalisation of the CPL table was re-written.

7.2.1.3 Implementation of OpenMP parallel CPL scanning

The code was then CPU parallelised using the OpenMP library. As in the serial code, the OpenMP version reads and normalises a CPL scan file and reads the protein sequences into an array where each index is a pair containing a protein name and a protein sequence, both held as strings. The serial version the code iterates through each of these pairs, scores all the peptides and dynamically inserts them into a results array if the ALS score is higher than the pre-existing contents. In the OpenMP implementation the array is split equally across each OpenMP thread. The scoring is performed identically as in the serial version, but with each thread having its own results array to save on race conditions. Then when all operations are finished, the results array is sorted, and the top n results are returned along with corresponding data.

7.2.1.4 Implementation of CUDA parallel CPL scanning

The CUDA implementation had some code alterations to allow for improved accessibility on the GPU device. The CPL scan file was read in as before but converted to a flat 1D array to allow for easier copying to the GPU. The FASTA data was converted to a single array of *char* values as CUDA does not allow for the std library strings, along with this char array, an array of *type int* is generated to denote the length of each protein sequence so that each CUDA thread can act across a single protein sequence without overlapping onto each other. Data is passed in chunks of 30,000 proteins at a time (if the input is greater than 30,000 sequences) in order to guarantee that all the data can be accommodated by GPU memory. Another array of *int* is used to denote which protein name corresponds to each peptide in the char array. The first step on the CUDA copies the CPL lookup table and the char array, generating a full lookup table for every protein. This is described in the pseudocode below:

```
peplen = length of k-mer epitope

CPL = CPL scan array

idx = CUDA device index

AA = amino acid index

scores = array of positional scores corresponding to protein sequence

sequence = amino acid sequence of protein


for residue in sequence do:

    for i in 1:23 do:

        if sequence[idx] == AA[i] then:

            scores[idx: idex + peplen] = CPL[i: i + peplen]

        end if

    end for

end for
```

Then the pre-existing character array is deleted from GPU memory and the protein name array and output array are also copied into GPU memory. The peptides are then scored by each CUDA thread scoring every peptide in a given protein as described below:

```
peplen = length of k-mer epitope

scores = array of positional scores, generated above

idx = CUDA device index

name idx = integer array corresponding to protein of origin of a given peptide

scores out = array of scores of peptides

names out = array of parent protein indexes corresponding to the peptide


for i in 1:23 do:
    if sequence[idx] == AA[i] then:
        scores[idx: idx + peplen] = CPL[i: i + peplen]
    end if
end for
scores out = scores[idx: idx + peplen] = CPL[i + peplen]
names out = name idx[idx]
```

This output is then sorted using an inbuilt radix sort in the thrust library (https://docs.nvidia.com/cuda/thrust/index.html). In short, the radix sort was chosen as it has previously been described to be a generally very efficient sorting algorithm that scales well on GPUs[137]. As the data is broken into chunks, in any operation beyond the first one is performed on a combination of the chunk outputted in that iteration and the previous scores. Both arrays are sorted together, and the top n are kept for the next iteration. On the final step the final sorted array is written to a tab-delimited text file.

7.2.1.5 Profiling

Profiling was performed both using Valgrind (https://valgrind.org/) to check for memory leaks and the NVIDIA Profiler (https://developer.nvidia.com/nvidia-visual-profiler) to measure how long was being spent on the CUDA operations versus those that performed on the CPU.

7.2.1.6 Benchmarking

Benchmarking was performed for the Serial implementation of the code, on OpenMP multithreaded using 4 and 8 cores and on the CUDA parallel device. The benchmarking was performed in two ways. The first was increasing the size of the database being scored against by ranking the same CPL scan dataset against an artificially generated dataset of proteins of length 311 amino acids (the average protein length seen in the pre-existing databases). Each implementation was run 5 times on databases ranging from 10^6 to $2x10^7$ sequences in length in increments of 500,000.

The second benchmark was to measure the effect of the number of results returned. This was known to place a significant burden on the performance of the code as the sorting of large results arrays were computationally expensive. The database size was fixed to 1.5x10$^{6 \text{ and}}$ were ran returning results ranging from 100 to 10^{6}.

All code was benchmarked on a machine running Ubuntu GNOME 16.19 with a NVIDIA QUADRO K1200 graphics card with 512 cores, an Intel Core 6700K processor and 16GB of RAM. All CUDA code was compiled with the NVCC compiler (version 8.0.44). The serial and OpenMP code was compiled with the GNU g++ compiler (version 5.4.0) using the C++11 ANSI standard. All versions, both CUDA and standard C++ were compiled with the highest optimisation flag (-O3). The OpenMP code was parallelised in the same fashion as the serial code, with the -fopenmp compiler directive to parallelise it.

7.2.2 Peptide ranking

The peptide ranking code was generated such that the code would accept two user inputs, a valid CPL scan file (a line delimited text file of MIP1β expression value of length 20 x peptide length) and a peptide of interest for ranking. Ranking is defined as the how high the ALS score is for the test peptide versus every other peptide of that length, irrespective of if it exists in a biologically meaningful sense. For a scan of 9-mer peptides this means ranking 20^{9} unique peptides.

While ranking could be meaningfully framed as a search problem or optimisation problem across search space, leaving it as an activity of scoring every possible peptide leaves this as an “embarrassingly parallel” problem that can be written, debugged and assessed quickly and still offer a performance increase compared to a single threaded or CPU-parallel implementation.

The overall step is shown in Figure 36. The CPL scan is read in as a two-dimensional array of floats. The index peptide is checked to see if it is a valid input (a string of the same length as the CPL scan file, containing the 20 standard amino acids) and scored. Unlike the database scan, the ranking code has the advantage of being able to generate the dataset *de novo,* rather than having to read it from a file in disk. Taking the example of a 9-mer peptide, every possible 3-mer peptide is generated for each third of the peptide. For peptides of longer length, the peptide is still split into thirds, just of larger chunks.

These 3-mers are copied to the CUDA device, along with the rank index value and the CPL scan array. Compared to the database scanning code this is significantly quicker as for a 9-mer peptide this means copying only 24,000 3-mer peptides into GPU memory compared to ~30,000 proteins, along with information linking each peptide to the position in the protein and the protein's origin or name.

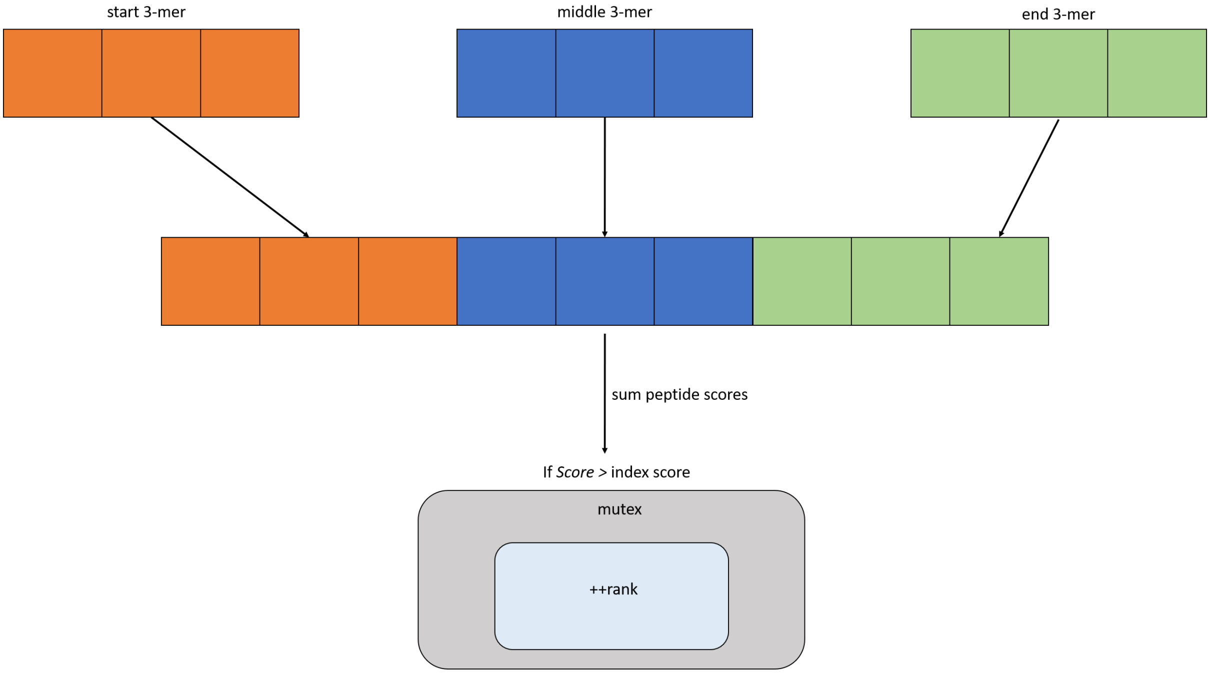


Figure 36 Overview of the steps taken in peptide ranking. For a 9-mer peptide, all possible 3-mers are generated and then copied into three pools. The CUDA operation then in parallel generates 9-mer peptides from these three pools. The peptide's ALS score is calculated and then compared against the peptide to be ranked, if the score is greater than that of the index then one is added to the rank in an access safe method using the CUDA API's inbuilt atomicAdd function.

Once copied into memory, threads are called in three dimensions on the CUDA device, e.g. across a given block, there are independent threads in the X, Y and Z "directions". These are then leveraged so that each independent thread is called in parallel for each "chunk" or third of the peptide being scored. In real terms this means that the X thread is responsible for allocating scores for the first third of the peptide, the Y thread for the middle and the Z thread for the end. These three sub-peptides are scored independently and then summed together. This peptide score is compared to the peptide of interesting and if the score is higher than the peptide then the rank value is added to using the atomicAdd function in the CUDA API. The atomicAdd function is used to modify a value in a memory safe way.

7.2.2.1 Benchmarking

This CUDA parallel implementation of the ranking was benchmarked against a single threaded implementation performing the same function. As the problem could be described as "embarrassingly parallel" the code was not optimised for performance in the serial version simply because the likelihood of good parallel performance was extremely high. Due to there being a memory safe operation in adding to the rank, it was expected that there would be a time difference between ranking the best possible peptide as there would be no race conditions, versus the worst possible peptide which would require access from every thread. In anticipation of this, ranking was performed on the best and worst peptide possible for the CPL scan of the INsB4 T-cell in both serial and CUDA implementations, along with some biologically interesting peptides generated by the CPL database scan.

7.2.3 Peptide alignment

In addition to outright using CPL scans to gain insight into biologically relevant peptides, a CUDA parallel code to perform a simple alignment was developed. The idea behind this was to find out how close or distant, in terms of alignment a given peptide was against a FASTA file of protein sequences. An interesting example of this would be in attempting to approximate potential cross-reactivity in terms of how similar a peptide is to the human proteome. To do this one would supply a list of peptides of interest in FASTA format and a FASTA of the human proteome. The script then calculates the average PAM30 alignment for each peptide in the query file against all peptides in the dataset.

Alignment is performed in C++ and CUDA. The code takes 3 arguments to be supplied by the user, a "query" FASTA file consisting of peptides the user is interested in aligning and a "reference" FASTA consisting of sequences of which the query is to be aligned to. The final option is the peptide length to be considered.

The FASTA files are read into memory and converted to peptides of length specified and a PAM30 alignment matrix is generated from a hardcoded set of data. The code then calculates how many device operations are required. The query peptides are handled by threads in the X direction of the block and the reference by Y threads. The peptides are operated in on chunks, passing each chunk onto the device, scoring the alignment between them before being deleted. This is because it is not guaranteed everything can be fit into GPU memory for large inputs. The average alignment is calculated by performing the PAM30 distance scoring in CUDA, before writing the alignment to a two-dimensional matrix where the rows represent each query peptide and the columns each reference peptide. Each row is then averaged to give how “close” the query peptide is to the entire proteome held in the reference.

7.2.4 Utilities

In addition to the parallel database searching and alignment codes, additional utility scripts were developed to help aid in identifying biologically interesting peptides. One of which was developed to query the IEDB to find if a peptide has been identified before. To do so the script downloads a zipped CSV file of all T-cell epitopes known in the IEDB (https://www.iedb.org/downloader.php?file_name=doc/tcell_full_v3.zip) unzips it. The script then takes the output file of PICPL, a tab delimited file containing the peptide, its parent protein(s) and its ALS. The script extracts this information and finds matching T-cell epitopes in the IEDB file. A file is returned that contains a combination of the ALS and database scan results as well as the experimental assay information from the IEDB.

7.3 CODE AVAILABILITY

The executable version of the C++ and CUDA versions of PICPL are available at https://github.com/whalleyt/PICPL. The source code is not available due to licensing constraints. The utility search script is available at https://github.com/whalleyt/CPLutils and the alignment code is available at https://github.com/whalleyt/peptide-aligment.

8 RESULTS: APPLICATION OF PARALLEL COMPUTING TO CPL DRIVEN EPITOPE PREDICTION

8.1 GPU ACCELERATED EPITOPE PREDICTION

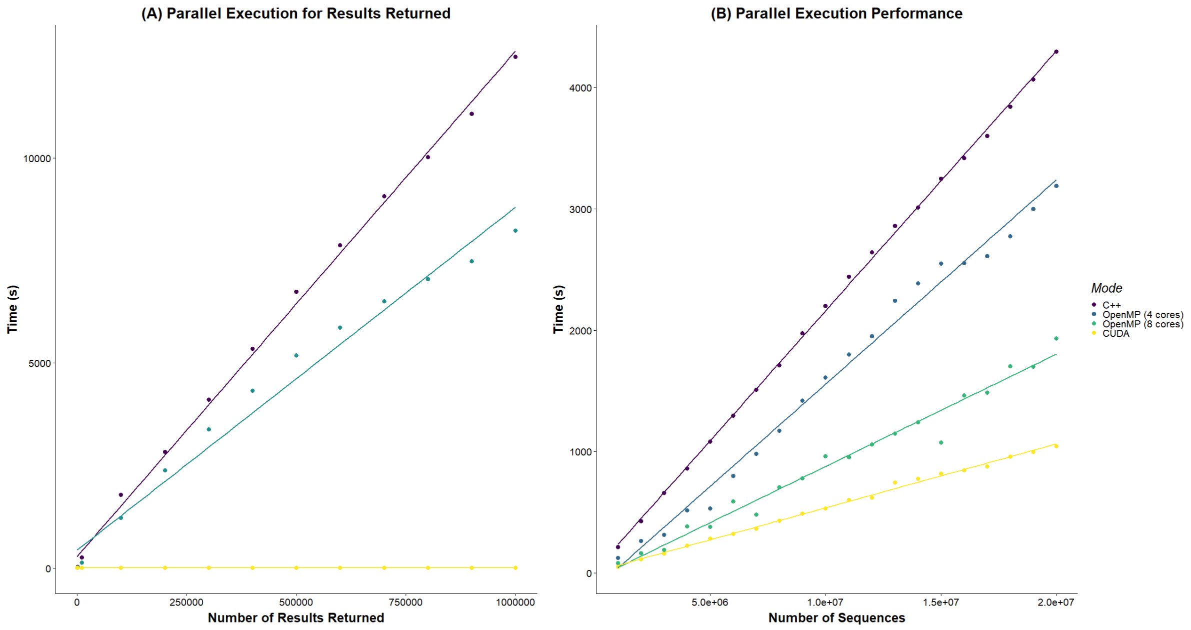


Figure 37 A) Run time for databases containing n number of sequences for the serial implementation in C++, the CUDA implementation and two OpenMP implementations using 4 and 8 cores respectively. B.) Run time for fixed number of sequences with varied number of results. Compared to serial C++ code, the CUDA implementation was 4.5x quicker.

As shown in Figure 37B, The CUDA implementation was significantly quicker than any other implementation. Compared to serial C++ the compute time was decreased by a factor of approximately 4.5. This appears to be linear also, meaning that this metric remains feasible well into the millions, if not billions of sequences. The CUDA implementation analysed 18,589 sequences per second versus 4,624 sequences and 11,800 sequences per second in the serial (C++) and OpenMP implementations, respectively.

Perhaps more importantly, the CUDA code was significantly quicker when increasing the number of results scored (Figure 37A). It appeared to only have a marginal increase in run time, unlike the serial and OpenMP versions which were slowed down significantly. This is important as this opens up the search depth of the data and allows the user to query lower scoring peptides without a marked increase in runtime. This increase in runtime is due to the inbuilt thrust library's radix sort.

8.2 THEORETICAL RANKING

The speed up of theoretical ranking between GPU and C++ was also significant. When comparing two extreme cases of the best possible peptide for the CPL scan of the INSB4 TCR (LLIENILFV) and the worst scoring peptide (GGVAADCDC) there was a significance increase in run time. This is because every time the queried peptide scores lower than a given peptide the CUDA threads cease, synchronise and perform an atomic operation meaning there is a delay. The more times this happens the slower the performance is. The highest scoring peptide took 82s whereas the worst took 1117 seconds, leading to a greater than 10-fold increase in runtime. However, it should be emphasis that even at its worst performance, the GPU accelerated theoretical ranking performed better than the serial C++ version which took 16088 seconds and 16144 seconds in the base and worst-case scenarios. So even when taking the worst case, the GPU optimised code is still ~16 times quicker.

8.3 PEPTIDE ALIGNMENT

Peptide alignment scales as function of database size causing exponential runtime growth. The notion of developing this script was first decided after runs in serial and parallel C++ were deemed unfeasible, meaning that no benchmark could be performed.

However, the script was benchmarked with two FASTA files of 10,000, where this run took on average 1 hour 25 minutes to score the average PAM30 distance between each peptide of the reference and query sets. This was stopped after a day in serial C++.

9 Background: RECIPIENT- A pipeline for pangenome reverse vaccinology

9.1 Reverse vaccinology

RV is centred around the usage of genome sequencing technologies and other high-throughput bioinformatics workflows to predict vaccine targets[122]. There is a strong demand for this as microbes are rapidly evolving and becoming drug resistant making existing therapies ineffective. This puts a burden on those developing new therapies both in terms of cost and time.

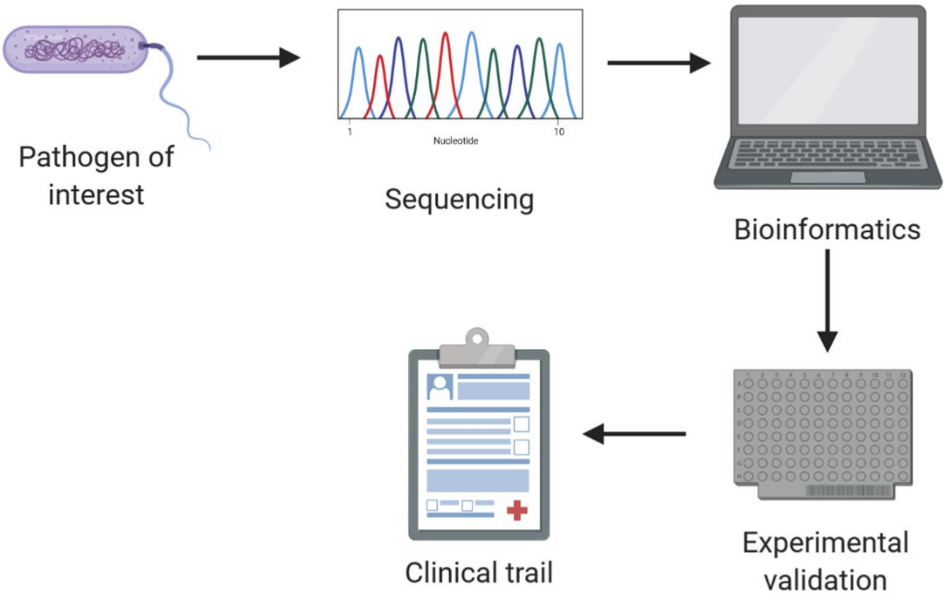


Figure 22 a schematic of the basic pipeline of RV, a pathogen species is identified, sequenced before a number of bioinformatics analyses are applied and candidate targets are selected. These candidates are taken forward for experimental validation and potentially clinical trials.

Fortunately, the rise in demand for therapies has coincided with a rise in technologies. Estimates suggest that there were well in excess of 20,000 whole genomes publicly available in 2015; and this number has surely grown since[123]. To compliment the explosion in genomic sequences available there is a growing number of bioinformatics tools to help with analyses in addition to an increasing number of bespoke databases, including those for antigen receptor data.

There is a breadth of RV tools[124–12], which contribute to the "bioinformatics" step of Figure 22, however many are not open-source, scale poorly in terms of high performance computing (HPC) and miss important biological parameters. Another effort that is missing is designing RV pipelines with broad population responses in mind. At the time of writing only one tool exists that targets the pangenome[129], that is the core genomic features shared between a bacteria/virus.

9.1.1 Microbial genomics

9.1.1.1 Microbial databases

RV has been aided by an increased number of databases and tools to support microbial genomics. Given the number of bacterial genomes available, it is often difficult to bulk download genomes. There exists a number of databases for downloading families of the same pathogen (e.g. species, serotype); one example being Enterobase[130]. Enterobase allows users to select and download genomes by their multilocus sequence type (MLST), serotype and other types of metadata. It supports several genuses, namely *Escherichia*, *Salmonella* and *Clostridioides* among others. Another example is the Global initiative on sharing all influenza data (GISAID)[131] which does a similar process for influenza viruses and biologically similar diseases (e.g. COVID-19). These tools are important in RV as it is often difficult to collect genomes from other public sources without a large degree of manual curation.

9.1.1.2 Genome annotation

There are also numerous bioinformatics tools that are essential for the basic characterization of genomes that underpin several RV pipelines. Once the user has a set of pathogen assemblies it is important to annotate them to identify genes and other important features. While this is simple in humans and other higher order eukaryotes as their genomes are relatively well-conserved, it must be done on a sample-by-sample basis with bacteria and viruses as their genomes are constantly evolving and a number of horizontal transfer events will be happening simultaneously.

The most well-adopted annotation software is Prokka[132] which packages a number of pre-existing feature prediction tools that predict the presence of certain features de novo. In the case of protein/peptide vaccines the important features are coding sequences (CDS) and signal peptides, predicted by Prodigal[133] and SignalP[134] respectively. This is then followed up by finding and annotating these features with pre-existing data using BLASTp[55]. If there is no direct match, then the CDS is matched to a protein family take from Pfam[135] or TIGEFAM[136] using HMMER[107].

Prokka may be the most widely used, but other tools for genome annotation exist, for example PGAP, which aims to reconcile the differences between *ab initio* predictions and database searches seen in other tools. There also annotation tools that prioritise speed[138].

Aside from classical genome annotation, there is also metagenomic taxonomy annotation, that aims to annotate metagenomic data and inform the user information about the make-up of species in a sample. This is also useful in RV as it can be used as quality control to detect contaminants in samples. An example of this type of software is kraken[139].

9.1.1.3 K-mer sketching.

Similar to this concept of taxonomic classification is the idea of using k-mer sketching to estimate the distance between samples. The distance measure between samples is useful again for detecting anomalies as well as describing the overall population of samples. Softwares such as Mash[140] and Dashing[141] estimate similarities in genomes by randomly sampling k-mers of DNA, hashing them and computing the Jaccard Similarity between samples in a pairwise fashion.

9.1.2 Pangenomes

The pangenome is the collection of genes shared across all samples in a species or subspecies. The pangenome is classically split into two main groups: core genes, which are shared by the majority of the isolates; and the accessory genes which are present in numerous isolates but not all[142]. This is summarized in Figure 23.

There are a number of pangenomic pipelines[142–145], all of which work in a broadly similar way. Sequences of the same family are detected by a homology search and their presence is detected in each genome. Paralogs and functionally similar groups are joined together to act as a similar gene.

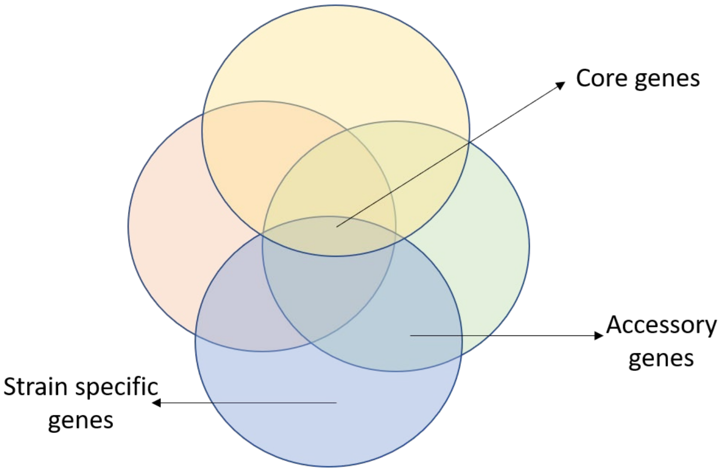


Figure 23 The basic concept of the pangenome. Each circle represents an entire genome. The centre, where all these genomes overlap can be considered the core genome. Others which have some overlap but not a large amount can be considered the accessory genome and final those with no overlap can be considered to not be part of the pangenome, instead just being specific to a given strain or isolate.

9.1.3 Population genetics measures of genes and proteins

Population genetics tests of neutrality are helpful for detecting whether a gene is undergoing natural selection. If a gene is dominated by a small number of highly abundant variants it is said to be undergoing negative or purifying selection. The reciprocal of this, positive selection is represented by a gene having a high number of low frequency alleles, suggesting that it is expanding and undergoing diversification. In the absence of selection in either direction the selection pressure is said to be neutral[146]. In the case of RV, it could be argued that genes undergoing negative selection are advantageous to being targeted as a vaccine as the alleles/variants under negative selection will remain in the population in high levels.

There are several tests for selection. Perhaps the most popular is Tajima's D[147]. Tajima's D is calculated as follows:

$$D = \frac{d}{\sqrt{\hat{V}(d)}}$$

Where *d* is the difference in two methods of quantifying diversity, the number of segregating sites and the number of mutations outright and $\hat{V}$ represents the variance. The number of segregating sites is the number of bases in the set of sequences that have more than one DNA base on them. These two measures are shown in Figure 24.

ATCGATGCGTGATCGTAGCTAGC
GTCGATGCGTGATCGTAGCTTGC
ATCGATGCGAAATCGTAGCTCGG
ATCGATGCGTGATCGTAGCTGGA

Segregating sites = 5
Total mutations = 8

Figure 24 Example of calculating the number of segregating sites and total mutants.

Tajima's D is a boundless number. If D is zero, then selection is said to be neutral. If D > 0 then there is said to be scarcity of rare alleles and negative selection is occurring. If D < 0 then there is said to be an abundance of rare alleles and the population is expanding, hence positive selection. Tajima's D should be assessed on a case-by-case basis, however if D is normally distributed then 95% of the values for D should fall within [-2,2]. Therefore if |D| > 2 then it is said to be undergoing a strong positive or negative selection as a rough rule. However, there is no literature supporting this further. The difference between a positive value of D and a negative one is shown in Figure 25.

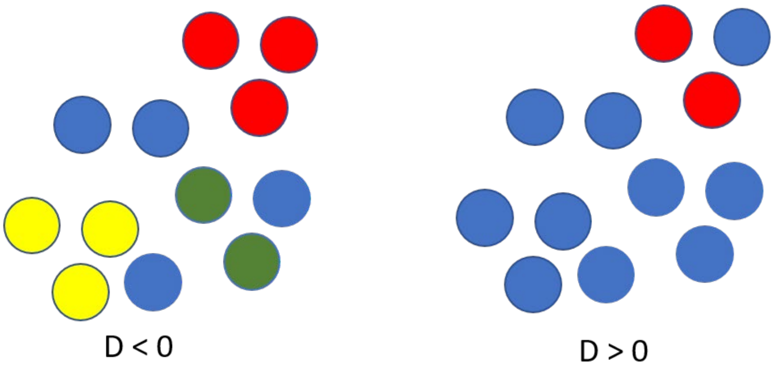


Figure 25 Cartoon demonstrating the interpretation of Tajima's D. Each circle represents the same gene in a different individual in the population. Each colour represents a different allele. The population on the left has an abundance of low occurrence alleles hence is undergoing positive selection (D < 0). Conversely the population on the right has a dominant allele and rare alleles are infrequent meaning negative selection is occurring (D > 0).

Other measures also exist for calculating neutrality. Fu and Li's D can also be used to test for selection. Fu and Li's D is used in cases where it assumed that all that data comes from one coalescent population, meaning that the genes in each sample derived from a common ancestor[148]. If RV were to be applied to broad groups of pathogens common ancestry cannot be assured with prior knowledge. Fay and Wu's H can be used to take into account the number of high frequency variants relative to intermediate frequency ones[149].

A final method of sequence diversity not rooted to the concept of population genetics is the entropy of a sequence. Although rooted in information theory and often applied to information and communication theory, entropy is often applied to sequences also[150,151]. A commonly used measure of entropy is Shannon's entropy[150], defined as:

$$H_n(p_i, p_1, p_2 \dots, p_n) = -\sum_{i=1}^{n} p_i \log_b p_i$$

Where n represents the number of possible states a value can take, p is their probability of occurrence and $\log_b$ is the logarithmic base of the user's choosing, in the case of sequence analysis typically 2.

9.1.4 Immunological considerations

The final facet for consideration in RV is the immunological context in which the protein takes. Immunogenicity can be measured in a number of ways. The first route of investigation would be MHC processing. The Immune epitope database (IEDB)[151] host a number of predictive tools based on neural networks to predict the likelihood of presentation of a peptide on a MHC class I molecule[154], MHC class II molecule[155]. This is complimented by predicting overall immunogenicity[154] and the likelihood of proteasomal cleavage for a peptide in the context of MHC class I[156]. As well as those supported by the IEDB there are number of other MHC class I prediction tools[157–159] and likewise for MHC class II[86,158]. Most models assume that good MHC binding will lead to some degree of T-cell response.

The other facet of RV epitope prediction is predicting B-cell epitopes. The IEDB also hosts tools for the prediction of linear B-cell epitopes[152]. However, as B-cell epitopes can be non-linear, the majority of these predictive methods require a structure. This, however, is out of the scope of most RV applications.

9.2 Salmonella enterica serovar Typhi

Salmonella is a bacterial genus of the family *Enterobacteriaceae*. Most *Salmonella* diseases in animals and humans is caused by serovars within the *Salmonella enterica* subspecies. These disease presentations can range from local gastroenteritis to fatal systemic disease. The exact outcome of the infections depends on both the physiology of the *Salmonella* serovar, but also on the host's immune status[153].

Salmonella serovars are varied in their ability to infect hosts; some have a wide-ranging variety of mammalian hosts. In contrast, *Salmonella* serovars Typhi and Paratyphi (*S.* Typhi and *S.* Paratyphi) have a limited host range, infecting only humans[154].

Typhoid fever is caused by infection with serovars Typhi or Paratyphi. As of 2017 there were 14.3 million cases of typhoid and paratyphoid fever occurring worldwide with an estimated 135,900 deaths[155]. Aside from improvements in sanitation and infrastructure, initial attempts at mitigating typhoid began with the administration of inactivated whole cell vaccinations. The whole-cell vaccinations however, were unsuccessful due to side effects[156].

There are now alternative types of typhoid vaccines available. The first, Ty21a is an orally administered attenuated strain. It contains the live-attenuated strain based on the pathogenic strain Ty2[157]. Ty21a has an inactivated galE gene, leading to it being unable to produce Vi antigen and other lipopolysaccharides[157]. The alternatives both use the Vi antigen to confer immunity. The first, Vi-tetanus-toxoid conjugate (Vi-TT) is the Vi antigen to the tetanus toxoid[158]. The second, Vi-PS is the purified Vi polysaccharide with no conjugate[159].

9.2.1 Studying the pathogenesis of Salmonella infection

It is important to note that due to the fact that *Salmonella* Paratyphi and Typhi only infects humans, it is difficult to study in typical model organisms. While humanised mouse models exist there are no good *in vivo* systems[160]. This means that most studies interested in the mechanism of typhoid fever study *Salmonella* Typhimurium. Serovar Typhimurium infected mice do however show a similar pathophysiology compared to humans, at least in terms of lesion placement in organs as well as the distribution of bacteria in tissues[161].

The transmission of *Salmonella* serovars happens predominantly through the faecal-oral route via the consumption of contaminated food or water. After being ingested, *Salmonella* invades the intestinal epithelial cells in the distal ileum[162]. Notably, *Salmonella* can invade via Microfold (M) cells. M cells are specialised intestinal epithelial cells that are involved in sampling luminal microbes to aid in mucosal immune surveillance[163]. M cells are found frequently over lymphoid structures known as Peyer's Patches (PP)[164] as well as other smaller lymphoid aggregates, for example solitary intestinal lymphoid tissues[165].

Salmonella's ability to access intestinal epithelial cells is conferred by several virulence genes encoded by the *Salmonella* pathogenicity island 1 (SPI-1). The protein products of SPI-1 form a Type III secretion system (T3SS) that allows the transport of several bacterial proteins into the host cytosol[166]. These proteins can then induce changes in the host cells leading to the rearrange of the cytoskeleton and cell membrane as well as the disconnection of epithelial cell junctions with facilitates *Salmonella* invasion[162]. Once they have accessed M cells, *Salmonella* can access the inner structure of the lymphoid tissue. This lymphoid tissue is dense in phagocytic cells meaning it is the initial site for intracellular infection[167]. Once this initial infection has occurred *Salmonella* can travel through the lymphatic system to mesenteric lymph nodes (MLNs) and gain access to the bloodstream and systemic tissues through efferent lymphatic vessels. This transport can be mediated by CCR7 in CD11c+ DCs[168].

Once spread, *Salmonella* is then able to replicate in phagocytes in the bone marrow, the liver and the spleen[169]. Another T3SS, this time encoded by *Salmonella* pathogenicity island 2 (SPI-2). SPI-2 allows for the evasion of macrophages by reducing the deposition of NADPH oxidases. This abrogation is dependent on the interference of the trafficking of oxidase containing vesicles to the phagosome[170]. *Salmonella* has also been shown to access DCs and

CD18+ phagocytes and disseminate rapidly to the blood, bypassing the need for lymphatic access. This pathway of action is suggested to be important for the rapid spread of *Salmonella* systemically[166].

9.2.2 Innate immune responses to Salmonella

The initial response to *Salmonella* often comes from epithelial cells which can initiate an inflammatory response and recruit phagocytes. The initial response to *Salmonella* in PP and MLNs involves neutrophils and inflammatory monocytes[169]. Neutrophil depletion in particular has been shown to be important in the regulation of *Salmonella* burden and spread; as neutrophil depletion has been shown to increase *Salmonella* load in the vasculature of the liver and spleen[171]. It has been suggested that NK cells play a role in producing IFN-γ in the early stages of *Salmonella* infection in mouse models. As some innate lymphoid cells express some NK cell markers, it is also suggested that some IFN-γ is released from them[172].

Overall, it is suggested that a multitude of different innate cells are involved in early *Salmonella* infection, centred around phagocytosis and IFN-γ production. Inflammatory monocytes are also recruited where they are routed to the PP and MLNs, producing factors such as iNOS, TNF and IL-1β leading to an inflammatory response[173]. Resident macrophages within the infected tissues have been shown to phagocytose *Salmonella* through recognition of its flagellin. This is mediated by the NLRC4 inflammasome complex and induces the release of IL-1β and IL-18, both pro-inflammatory cytokines[174]. *Salmonella* flagellin and LPS has also been shown to be recognised by DCs, causing the expression of CCR7, CD80, CD86 and CD40. The maturation of these DCs leads to enhanced antigen presentation abilities and allows them to migrate to the T-cell rich area of the lymphoid tissue to engage an adaptive immune response[175].

9.2.3 CD4 T-cell response to Salmonella

The study of *Salmonella* specific T-cell responses is a difficult process because the abundance of the naïve T-cell repertoire is low and there are few known *Salmonella* epitopes with known MHCs; at the time of writing there are 99 epitopes for *Salmonella* in the IEDB[151]. Most commonly, techniques to overcome this involve transferring naïve T-cells into sites of infection in order to surpass the threshold of detection by flow cytometry and immunohistology. Early attempts of this study used ovalbumin (OVA) specific T-cells in response to a recombinant *S.* Typhimurium strain expressing chicken OVA[176]. A clear caveat to any conclusion from these types of recombinant analyses is that the conclusions are not drawn from responses to endogenous antigen; instead from heterologous antigen which is over-expressed. To overcome this, adoptive transfer systems into natural *Salmonella* epitopes were developed. In this model, CD4 T-cell activation was shown first in the PPs followed by the MLNs after oral infection. These CD4 T-cells expressed CD69, followed by IL-2[167].

In the early stages of T-cell activation, draining MLNs are also a key site of T-cell activation. T-cells specific to *Salmonella* can be detected after 9-12 hours. This is not seen in other secondary lymphoid tissues, pointing to the importance of the MLNs. Further to this, the removal of MLNs *in vivo* showed an increased bacterial load and dysregulated immune function in infected mice's livers[168].

CD4 T-cells also have an important role in protective immunity. Mice without a thymus, αβ T-cells, MHC class II and T-bet+ Th1 cells have all been shown to be unable to clear infection whilst lack of γδ T-cells and B-cells did not aberrate immune function in response to *Salmonella*[177]. The role of CD8 T-cells is not wholly clear, some evidence exists suggesting the CD8 T-cells are not essential for clearance of *Salmonella* as β2M deficient mice suffered persistent infection[177]. B2M deficient mice however will lack non-classical MHC molecules and CD1 and experiments using mice missing only MHC class I suggest that there is a small protective role from CD8 cells[178].

CD4 and CD8 T-cells both have an important role in bacterial clearance in response to a secondary infection. In adoptive transfer experiments this immunity is not conferred by transfer of spleen cells alone, it also required the addition of immune serum[179]. In concordance with this, mice lacking B cells were able to clear a primary infection but could not overcome subsequent secondary challenges, suggesting a role for antibodies in this clearance. In mice that were unable to isotype switch or secrete antibodies, however, it was shown that B-cells did still play a role in this clearance, suggesting that there was also an importance in antigen presentation and cytokine release from B-cells[180]. The initial proliferation of T-cells involves communication with DCs[167], suggesting that the B-cell mediated antigen presentation occurs after the DC mediated presentation. Overall, there seems to a strong emphasis on the role of CD4 in acquired immunity to *Salmonella* infection, whilst being complimented by contributions from CD8 T-cells and B-cells.

CD4 T-cells undergo a large clonal expansion after *Salmonella* infection; these expanded CD4 T-cells are also able to migrate to non-lymphoid tissues such as the liver as well as gaining effector function[181]. This gain of function and expansion into other compartments is suggestive of a role for the CD4 effectors in regulating and controlling bacterial replication. *Salmonella* specific CD4 T-cells have been shown to be stable for over a year post-infection. The stability of these cells appears to be governed by small but important activity of pMHC class II complexes being presented on chronically infected phagocytes to CD4 T-cells[182].

Mice lacking T-bet, IFN-γ or its receptor IFN-γR are unable to clear *Salmonella* infection, suggesting a role in Th1 cells. IL-17 and IL-22 has been shown to be produced in the intestinal mucosa early on in *Salmonella* infection, suggesting that Th17 cells also have a role. The presence of these cytokines is not unique to Th17 cells, but Th17 cells have also been detected in the mucosal tissues following infection[183]. It has been suggested that Th1 cells are important in the clearance of bacteria in *Salmonella* infection, whilst Th17 cells play a role in protecting spread from the intestine.

Salmonella responsive CD4 T-cells can relocate to infected tissues and secrete effector cytokines. While most studies focus on cognate stimuli, there is a role for non-cognate activation. While this has been studied more in the framework of CD8 T-cell responses to viruses, typically involving IL-12 and IL-18[184], there is a growing pool of analysis surrounding responses to bacteria and *Salmonella*. The NLRC4 inflammasome is activated and IL-18 is released by CD8α+ DCs leading to OVA specific memory CD8 T-cells in *Salmonella* infection[174]. It has been shown that CD4 T-cell effector function can be elicited indirectly after injection of LPS, leading to IFN-γ release. The response is associated with induction via TLR agonists. The TLR agonism also requires inflammasome components NLRC4 and NLRP3 and leads to IL-18 release[185]. This route of innate stimulation of T-cells may aid in lowering the threshold for CD4 T-cell activation. It is speculated that this pathway may be useful in helping clear co-infections or super-infections, as co-current infections are likely to occur naturally in an in-vivo setting[166].

9.2.4 Antigen localisation in Salmonella

There are few known antigens of *Salmonella*. However, in a mouse typhoid model it was shown that antigens for *Salmonella* were preferentially found to be on the outer membrane of the bacteria. Surprisingly, this is not thought to be related to any sort of inherent immunogenicity[186].

One hypothesis is that surface antigens might become more rapidly available and internal antigens only become available once an immune response is already engaged. Another proposed model is that many live *Salmonella* were either found alone or not with other dead *Salmonella*. As there is no internal antigen as there are no dead *Salmonella* nearby meaning that the immune response must be modulated by an external antigen.

9.3 Enteropathogenic *Escherichia coli*

Diarrheal disease is a prominent cause of morbidity and mortality, especially in children under five years old. The burden of the disease is primarily placed on populations in the developing world where sanitation, water access and logistical avenues for medical interventions are lower than in developed countries. However, cases also do exist in developed nations[187].

Escherichia coli (E. coli) is reported frequently in the developing world. *E. coli* is a versatile bacteria, with life cycles ranging from commensal to invasive. Enteropathogenic *Escherichia coli* (EPEC), an *E. coli* pathovar is a common causative agent of diarrhoeal disease.

Classically, *E coli* pathotypes were typed by 3 antigens: the O (somatic), H (flagellar) and K (capsular) antigens. Now serogroups O39, O88, O103, O145, O157 and O158 are considered EPEC pathotypes. Of the H antigen, H2 and H6 are commonly considered EPEC, whilst other minor H antigens can also be EPEC. However, many EPEC strains do not possess the H antigen at all**.** These strains of EPEC are known as non-motile[188]. Due to the increasingly apparent diversity of the O, H and K antigens, serotyping in this way is not always considered a useful diagnostic, with WGS methods instead being employed[189].

Now EPEC is defined based on its virulence factors and phenotypically as a diarrhoea causing strain of *E* coli that can produce attaching and effacing (A/E) lesions, but cannot produce Shiga toxins, heat labile (LT) or heat stabile (ST) enterotoxins[190].

EPEC is characterised by the locus of enterocyte effacement (LEE). LEE encodes an adhesin intimin, a T3SS (composed of EspA, EspB and EspD) alongside six other effectors (Tir, EspF, Map, EspG, EspH and EspZ)[191].

A/E lesion formation requires LEE. A/E lesions are characterized by attachment of bacteria to the apical plasma membrane of intestinal cells, local accumulation of F-actin and effacement of the brush border microvilli[9]. The formation of A/E clusters is mediated by intimin. The attachment process is facilitated by Tir (translocated intimin receptor) which inserts into the host plasma membrane where it acts as a receptor for intimin[192]. Once attached the T3SS can inject effector proteins into the host cells.

5 of the 6 LEE effectors are inducers of cytotoxicity, reorganisation of the cytoskeleton and of electrolyte imbalances leading to diarrhoea. The remaining effector EspZ integrates into the plasma membrane and regulators effector translocation, protecting infected cells from cytotoxicity[193].

EPEC can be further grouped into typical EPEC (tEPEC) which has the plasmid encoded bundle forming pilus (BFP); and atypical EPEC (aEPEC) which does not contain BFP[194]. EPEC has been shown *in vitro* to form 3D microcolonies known as localised adherence (LA) patterns. BFP mediates LA pattern formation, as well contributing to antigenicity, auto-aggregation and biofilm formation[195–197].

However, even this sub-categorisation into the tEPEC:aEPEC binary is an over-simplification. EPEC infection can range from lethal to non-lethal and non-lethal infections can range from having severe symptoms to being asymptomatic. It has been observed using BLAST score ratios (a technique used to create a distance matrix between genomes)[198] that no single gene cluster could be attributed to a single clinical outcome, suggesting that EPEC pathogenicity is a multi-faceted arrangement[199]. A similar study by *Hazen et al.*[200] had a number of EPEC samples that did not cluster with other samples. The authors also observed that *bfpA,* a gene encoding for a subunit of the BFP would seemingly undergo several independent loss and acquisition events in different lineages.

Aside from the LEE effector locus, there are a series of other effectors found elsewhere in the EPEC genome. A number of these effectors follow the nomenclature of "non-LEE encoded effector" (Nle) A number of these are found on prophage regions. Prophage 2 (PP2) contains 3 effectors NleH1, cycle inhibiting factor (Cif) and espJ. nleH1 has been shown to prevent the translocation of ribosomal protein S3 (RPS3) inhibiting NF-κB[201]. Cif is a member of a family of toxins known as cyclomodulins which modulate host cell cycle. Cif has been shown to both induce cell cycle arrest but also delay apoptosis in epithelial cells. Cif is delivered by the T3SS[202]. EspJ is known to prevent phagocytosis[203].

PP4 contains 4 Nle genes: NleG, NleB, NleC and NleD. NleG shows a lot of functional diversity. It is known to target host ubiquitination machinery, however the specific function of members of the family as well as their targets of ubiquitination remain unclear. NleG5-1 has been shown to localise to the host nucleus and target the MED15 subunit (mediator of RNA polymerase II transcription subunit 15) and NleG2-3 appears to localise to the host cytosol where it leads to the degradation of hexokinase 2 and SNAP29[204]. NleB represses

NF-κB activation, but the mechanism at which is unclear, although it is thought to target glyceraldehyde 3-phosphate dehydrogenase (GAPDH)[205]. NleC is a metalloprotease. NleC is also shown to repress NF-κB function, this time it appears to be mediated by the cleavage of p65 which affects downstream p65 interaction with RPS3[206]. Finally, NleD, another metalloprotease inhibits mitogen activated protein kinase (MAPK) signalling proteins JNK and p38 by translocating into host enterocytes where it cleaves and inactivates them[207].

PP6 contains NleA/EspI, NleH2, NleF, espO. NleA has been shown to inhibit NLRP3 inflammasome activity by the inhibition of caspase 1[201]. NleH2, like NleH1 has been shown to attenuate NF-κB activity through modulation of MAPK signalling via p38[208]. NleF binds to caspases 4,8 and 9. This caspase inhibition leads to apoptosis inhibition[209]. EspO also inhibits apoptosis. This inhibition is mediated through HS1-associated protein X1 (HAX-1)[210].

9.3.1 EPEC vaccination efforts

Natural immunity after EPEC infection has been observed and antibodies have been shown to be protective against future infections. Studies in developing countries have shown IgA antibodies against intimin, EspA, EspB, EspC and BFP[211]. Additionally, antibodies responsive to EsPA, B, C and D can confer protection against EHEC pathotypes expressing the LEE locus. BfpA and B, although not always expressed on EPEC, are also potential targets. IgA antibodies against them have been detected in the faeces of children who had been breastfed and had acute diarrhoea but not in those who were not breastfed[212].

EspB has been of particular interest as a target of vaccination. However, it is difficult to target as there are three main variants of EspB: α, β, and γ, with the α variant having 3 further subcategories[213]. However, this could possibly be overcome by targeting of particular conserved epitopes in the EspB sequence.

9.4 Hepatitis B

Hepatitis B virus (HBV) infection results in substantial human morbidity and mortality, especially through the consequences of chronic infection. Estimates of people chronically infected with HBV range from 240 million to 350 million globally[214]. HBV was estimated to have contributed to 786,000 deaths annually, with 341,000 of those being liver cancer and 312,000 being liver cirrhosis. This places HBV infection 15th among all causes for human mortality[215].

9.4.1 Genome structure

The HBV is a member of the *Hepadnaviridae* family. I t is a small DNA virus that replicates through an RNA intermediary and can integrate into the host genome. This unusual method of replication allows the virus to persist in infected cells. HBV is categorised into 8 different genotypes A to H, these genotypes have a distinct geographic distribution. HBV's genome encodes four overlapping ORFs (S, C, P and X)[216].

The S ORF encodes for viral surface envelope proteins. The C ORF encodes for the viral nucleocapsid HbcAg or the hepatitis B e antigen (HbeAg) depending on where translation is initiated at the core or pre-core region. The core protein can self-assemble into a capsid like structure; while the smaller pre-core region codes for a signal peptide that directs the translation product to the ER where the protein is processed to form HbeAg[217]. The P ORF encodes for the polymerase, a large protein of ~800 amino acids in length. Finally, the X ORF encodes for HBxAg, a protein with multiple functions, including signal transduction, transcriptional activation, DNA repair and protein degradation. Although it is known that HbxAg is required *in vivo* for infection to occur, the reasoning is unknown. HbxAg has, however been implicated in a number of biological processes including signal transduction, activation of transcription, DNA repair and inhibiting protein degradation[216].

9.4.2 Immune responses

Understanding of innate immune responses to HBV is limited by epidemiological issues, very few patients are recruited when they are undergoing early infection as they are often asymptomatic. Likewise, *in vitro* systems are inefficient and replication of the HBV virus is low in them[218].

However, studies into the innate immune response to HBV has yielded insights. Weak activation of innate immunity is typical of acute HBV infection in adults. Levels of pro-inflammatory cytokines are low in the first 30 days of infection[219]. It is thought this low-level innate response is a result of HBV escaping innate regulation by having covalently closed circular DNA (cccDNA) to the cell nucleus and having intermediates of replication (both in RNA and DNA form) to the cytoplasmic core particles[220]. However, this is by no means the accepted paradigm, *Durantel and Zoulim* for example, argue that HBV actively suppresses the innate immune system[221].

The adaptive immune system has been recognised as a key player in the clearance of HBV infection. Here, CD4 T-cells produce large quantities of cytokines and are essential for the development of CD8 CTLs and B-cell antibodies. CD8 T-cells clear HBV infected hepatocytes through both cytolytic and non-cytolytic activity whilst B-cell antibodies neutralize free viral particles, preventing reinfection[222].

Not much is known about the induction of B-cell response in acute HBV, however CD4 and CD8 T-cell mediated response are generally detectably around the point where HBV begins to exponentially replicate within its host. Usually, this point occurs 4-7 weeks after infection[223]. Of known epitopes, CD4 T-cells seem to prefer peptides from the capsid protein, where CD8 T-cells seem to recognise a wider array of peptides[224].

During acute infection, HBV is often self-limiting leading to a residual infection that can return during immunosuppressive events. HBV DNA levels decline by up to 90% in a 2–3-week period after the peak levels of replication. This decline occurs with very little indication of liver damage, suggesting that mechanisms are mediated in a non-cytopathic manner by means of IFN-γ and TNF release by CD8s[225]. There is however, a small portion of action that is cytopathic. Recruitment of HBV specific CTLs is promoted by the secretion of CXCL-10 and platelet activation and leads to the killing of infected hepatoycytes[226].

Once infection is successfully controlled, the maturation of T-cell memory occurs assertively[227]. Between this maturation of memory and the initial clearance of infection, there is a functional impairment of CD8s. At this stage, T-cells are activated but struggle to proliferate and show signs of exhaustion. Normally, this is associated with the peak of IL-10 production and the release of arginase from dying hepatocytes[219,228,229]. The release of arginase is thought to contribute to the down-regulation of the CD3ζ chain by depleting L-arginine.

In chronic HBV infection, the T-cell response is much weaker. Seemingly irrespective of the cause of the infection being chronic (maternal transmission, abundance of antigen, MHC profile etc.) the prolonged expression of antigen can aberrate T-cell response. This dampening down of immune response is mediated by expression of PD-1, CTLA-4 and Tim-3 leading to less proliferation, cytokine production and apoptosis[230].

The length of chronic infection is also a strong influence on T-cell function. Young patients with a lower length of infection show less T-cell exhaustion than older patients with longer infections[231].

9.4.3 HBV antigens

HBV infection leads to the persistent release of the soluble form of HbsAg and HBeAg. Both are derived from the C ORF. HBsAg has been suggested to impair the abundance and function of DCs by modulating TLR-2 expression as well as interfering with TLR mediated cytokine release[34]. Soluble HBV antigens have also been understood to inhibit antigen presentation function, the interference of cytokine production and inhibition of T-cell response[35]. It remains unclear why these responses only exist in response to HBV as one would expect a chronic HBV infection to correlate with more opportunistic infections.

However, there are a number of caveats to this model. The first is that experiments tend to be performed *in vitro* with proteins based on yeast or *E. coli* expression systems; or were purified from the sera of chronic HBV patients, potentially leading to LPS and other contaminants being present. LPS induced tolerance of APCs via TLR agonism is a known phenomena, meaning that the outcome of the experiments could have been impacted[232]. Also, patients chronically infected with HBV often have high levels of IL-10 and liver enzymes, meaning they could be modulating immune response, rather than direct action from HBV antigens. In patients with high levels of HBsAG but no or mild levels of liver inflammation the frequency of T-cells, their function and the level of circulating professional APCs (DCs, monocytes and B-cells) did not change[233].

9.4.4 Current HBV vaccine treatments

The majority of vaccine therapies aim to induce functionally effective HBV specific T-cells. T-cell response against HBcAg is important for resolution of HBV infection, but most targets are only aimed at envelope proteins. This has been explored with woodchuck model's and a DNA prime-adenovirus vaccine containing HBcAg[234].

9.5 SARS-CoV-2

Severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2) accounted for 4.7 million deaths and 230 million known cases as of September 2021 (https://covid19.who.int/). Viral genome sequences available via public initiatives such as the Global Initiative on Sharing All Influenza Data (GISAID) exceed 1 million permitting unparalleled levels of genomic surveillance[235].

SARS-CoV-2 is a positive sense single stranded RNA virus. Its genome is about 30kb in length and is mostly protected by a fatty envelope layer. SARS-CoV-2 is a member of the *Coronaviridae* family. Members of the *Coronaviridae* family are further split into 5 genera, 4 of which are members of the subfamily *Orthocoronavirinae* lettered from α to δ. α and β have the ability to infect humans, whilst γ and δ mostly infect birds and pigs[236]. SARS-CoV-2 is a β coronavirus.

Similarly, to SARS-CoV and MERS-CoV, the genome of SARS-CoV-2 is comprised of 12 ORFs. At the 5' end of the genome, two overlapping ORFS named 1a and 1b encode for the RNA polymerase and other non-structural proteins. These two ORFs occupy approximately 2/3 of the genome. Structural proteins such as the spike (S), Membrane (M), envelope (E) and nucleocapsid (N) are present in the remaining third, stretching to the 3' terminus[237].

SARS-CoV-2 uses the angiotensin converting enzyme 2 (ACE2) receptor to gain access into human cells. The binding of the ACE2 receptor governs its pathogenicity. The binding between the S protein and ACE2 receptor is 10-20 times stronger than that of the S protein of SARS-CoV[238].

Coronaviruses gain entry inside the target cell by engaging the host receptor with the S glycoprotein. The region of the S protein containing the receptor binding domain (RBD) is present on the S1 subunit. Once in the cell, the virus releases its positive sense single stranded RNA genome into the cytoplasmic compartment where the translation of ORF1a and b begins. This translation leads to the production of two polyproteins, pp1a and pp1ab. Three functional proteases then cleave these polyproteins into 16 non-structural proteins (NSP1-16) which create the viral polymerase and other assembly proteins. The E protein is incorporated into the rER or Golgi apparatus. The RNA combines with capsid protein to form the nucleocapsid and then the assembled virus particles are budded from the ER-Golgi Intermediate Compartment (ERGIC). The virus laden vesicles are fused to the cell membrane for shedding. These virions are accessible to infect nearby healthy cells[239].

9.5.1 Immune response to SARS-CoV-2

Humoral immune response to SARS-CoV-2 is mediated by antibodies targeting the spike glycoprotein and the nucleocapsid protein predominantly. The S1 subunit in particular seems to be an important target for neutralising antibodies[240]. The major role of neutralising antibodies is for antigen binding and interacting with cells bearing Fc γ receptors to modulate immune responses and as such IgG responses have been detected against the nucleocapsid, S1, ORF9b and nsp5 among others by means of proteomic microarray[241].

Patients with infected with SARS-CoV-2 or recovering from it also have IgM and IgA responses. The antibody response is characterised by seroconversion of IgM and IgG one to two weeks after symptom onset and antibody concentrations persist for weeks or months after viral clearance. In longitudinal studies, IgA antibodies were produced early, followed by IgM[242]. In a study of patients with mild COVID-19, a decline in IgG titres specific to the RBD of the spike decline after 2-4 months[243]. However, numerous other papers report that antibody kinetics are much more persistent, robust responses[244,245].

SARS-CoV-2 specific T-cells express perforin 1 and granzymes upon *in vitro* restimulation with SARS-CoV-2 antigen. Using expression levels of activation markers (4-1BB ligand receptor and CD40-L) as a measure for CD4 T-cell activation, *Braun et al.* demonstrated that 83% of patients with COVID-19 had spike epitope specific CD4 T-cells. Perhaps more notably, they identified T-cells reactive to spike glycoprotein in 35% of their patients who had not had COVID-19[246]. In another study, this CD4 T-cell response was shown to be predominantly mediated by Th1 cells characterised by their high levels of IFN-γ release. They were specific for the spike glycoprotein, the membrane protein and the nucleocapsid protein predominantly, but there were also lesser responses to non-structural proteins[247].

10 METHODS: RECIPIENT- A PIPELINE FOR PANGENOME REVERSE VACCINOLOGY

10.1 AIMS

As discussed previously in the introduction, the wealth of genomic data means that reverse vaccinology (RV) practices are becoming more applicable on a large scale. This chapter describes the development for RECIPIENT (REverse vacCInology for PotentIal vaccinE caNdidaTes) a pipeline for the design of interesting protein and peptide vaccine candidates based on immune recognition and evolutionary conservation.

10.2 DATA COLLECTION

The pipeline was tested on four datasets, two sets of bacterial genomes and two sets of viral genomes. These four datasets comprised of *S.* Typhi, EPEC, SARS-CoV-2 and HBV.

As discussed in 9.2-9.5, all four of these pathogens present a significant burden in terms of mortality and the required healthcare interventions. Furthermore, the four datasets represented a diverse set of biological challenges to benchmark RECIPIENT against. The *S.* Typhi and EPEC datasets both consisted of bacterial genomes. Bacterial genomes in the context of RV are interesting because typically their genomes have a larger number of genes or ORFs to search against. Computationally, this increases the demands on the pipeline to deliver results in a reasonable timeframe. Biologically, this increased number of potential targets makes the need for a computational pipeline that can effectively filter candidates a necessity. The viral datasets provide an interesting contrast to this, as they have a smaller number of targets to select from.

EPEC has a number of effector proteins known to be unique to it compared to commensal *E. coli*, meaning that the selection of it as a test dataset gave a number of specific gene targets to search for in validating the pipeline. In contrast the *S.* Typhi, while still having a number of well-described immune targets, had a number of potential undiscovered effectors that were unique to *S.* Typhi but not described in a RV context.

The first dataset was collected from Enterobase (https://enterobase.warwick.ac.uk/)[248]. This dataset consisted of 785 genome assemblies of *Salmonella enterica* serovar Typhi. The complete list of accession Ids for Enterobase can be found in supplementary list S2.

The next dataset consisted of 58 EPEC genome assemblies taken from Enterobase. The accession IDs can be found in supplementary list S3.

In contrast to the two datasets above, The third dataset was collected from GISAID[235] and encompassed 123 genomes from across the 2020 global pandemic of COVID-19. SARS-CoV-2 was used to show the tool can work on RNA viruses and those with little functional annotation compared to bacteria. The accession IDs are shown in supplementary table S4.

The fourth dataset comprised of 61 genomes of *Hepatitis B* virus, collected from HBVDB[249]. These two choices were selected to provide a complementary analysis to the SARS-CoV-2 dataset and demonstrate the tool can work on both DNA and RNA viruses. The accession IDs are shown in supplementary table S5.

10.3 Pipeline

10.3.1.1 Implementation

The RECIPIENT pipeline was implemented in the Nextflow[250] pipeline manager. There were several reasons for this. Firstly, it allows for reproducibility of the dataset, log files describe exactly how the pipeline was executed and what errors were hit. These are then cached so the errors can be replicated or if the workflow finishes successfully, they can be re-ran by another user without having to fully compute them.

Nextflow also allows for the use of Singularity (https://singularity.lbl.gov/), Docker (https://www.docker.com/), Conda (https://docs.conda.io/en/latest/) containers and environments. The benefit of this is two-fold. The first is that it allows for greater reproducibility as it guarantees that versions of different packages are the same, meaning that subtle changes between versions are avoided. The second is that it means that the user does not have to install each software package individually which can prove difficult to ensure tools can run on different platforms. Similarly, many HPC environments make it difficult for a user to install third party software. Therefore, RECIPIENT comes with a series of Singularity containers to expediate installation. Singularity was chosen because it is very commonly used in HPC environments and it is assumed most HPC users will have access to it. Likewise, Singularity is also supported on local systems. In both it does not need to be ran with administrator/root privileges meaning that it was chosen over Docker.

Nextflow also automatically distributes jobs. It creates a directed acyclic graph (DAG) by pairing task input to other tasks outputs, meaning it can schedule jobs in the correct order and optimise for CPU core usage. This is pertinent again in local and HPC environments. RECIPIENT makes use of Nextflow's labelling system meaning that highly parallel or memory hungry tasks are assigned more resources. This is assigned by default; however, users can modify this according to their environment.

As well as job distribution, RECIPIENT makes use of Nextflow's job scheduling capabilities. On release, RECIPIENT comes with 3 profiles designed for local execution or HPC execution on systems using the SLURM and PBS job schedulers.

10.3.2 Workflow

The workflow itself is described in Figure 38. The pipeline derives a number of measures that are amenable to selecting candidate proteins or peptides for vaccinations. The initial first step is one of quality control (QC).

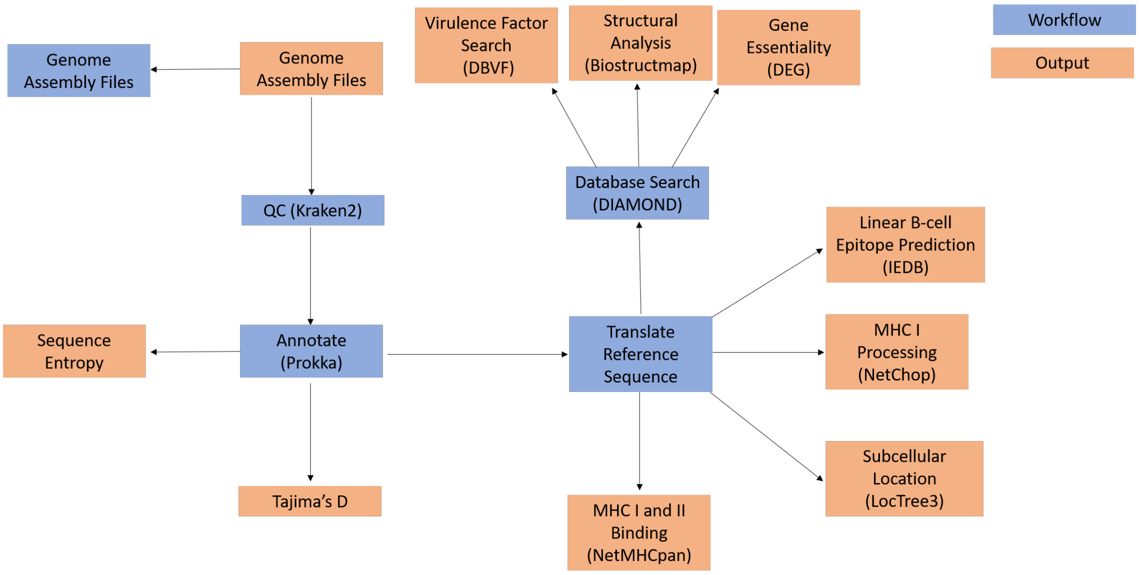

Virulence Factor Search (DBVF)
Structural Analysis (Biostructmap)
Gene Essentiality (DEG)
Workflow
Output
Genome Assembly Files
Genome Assembly Files
Database Search (DIAMOND)
QC (Kraken2)
Linear B-cell Epitope Prediction (IEDB)
Sequence Entropy
Annotate (Prokka)
Translate Reference Sequence
MHC I Processing (NetChop)
Subcellular Location (LocTree3)
Tajima's D
MHC I and II Binding (NetMHCpan)

Figure 38 Overview of the analysis pipeline. Boxes highlighted in orange represent steps which generate an output file, while blue shows steps used only for part of the pipeline

QC is performed by first classifying the taxa found in each sample to detect either mislabelled or contaminated sequencing data which may affect the creation of a pangenome.

In RECIPIENT's case it is done with Kraken2[139]. This is then complimented by a measure of Kmer distance between samples to detect outliers that will again affect the creation of a pangenome. K-mer distance calculation is performed using MASH[140].

Once QC has been performed the dataset is annotated using Prokka[251] and a pangenome is created using Roary[252] Following this there are two sequence diversity measures calculated first Tajima's D[147] then the Shannon Entropy[150] of each set of genes in the pangenome.

Each pangenome sequence has a reference, this then translated where it is passed into MHC class I[253], MHC class II[254] binding prediction; prediction of linear B-cell epitopes[152], proteasomal cleavage[255] and TCR binding prediction[79] (should the user choose to investigate specific TCRS). Subcellular localisation is then predicted for each gene[256]. A number of physiochemical parameters for each sequence is also calculated, ranging from purely physiochemical factors[257] to latent representations of sequences that are shown to have immunogenic interactions[258]. The sequence of each reference is also BLAST searched against for PDB homologues, then various physiochemical parameters are collected using the biostructmap package[259]. Transmembrane helices are also predicted[260]. Finally these genes are then annotated for essentiality[261] and virulence factors[262].

10.3.3 Quality control

QC was broken down as described in the sections below. The two methods used were read classification with Kraken2. Kraken2 was used to identify foreign reads and contaminants in the assemblies. MASH was used to calculate the distance between all genomes in a dataset.

10.3.3.1 Taxonomic read classification with Kraken2

As the data supplied is a whole genome assembly, it is assumed that quality control has been performed to some degree on the read level data. Metagenomic quality control on each assembly file is generated using Kraken2 [139,263] against the minikraken2 database. Kraken2 is a software designed to identify metagenomic reads in a dataset and assigned a taxonomic classification to them. In many use cases, this can be used to assess the ecological diversity of metagenomic sequencing, but it has also been applied to classical genomic sequencing to search for mis-assigned files or files with a high level of contamination with other species. Colloquially, this has also been described as a reason why constructions of pangenomes fails in Roary, also (https://sanger-pathogens.github.io/Roary/).

Kraken2 masks Kmer sequences (of a length of 35 bases by default, but the user can change length according to their need) into a spaced seed map. A spaced seed map is a representation of a string in which there are a number of "wildcard" base positions which are assumed to be degenerate[72,173]. This significantly increases the speed and sensitivity of homology searches[266]. Kraken2 optimises for memory then by creating a probabilistic hash table. A hash table is a data structure that maps keys, in this case the original Kmer seed map, to a value, in this case a taxonomic identification. It does this through a hash function which maps variable length strings to a more compact representation. In the case of Kraken2 this is done probabilistically, meaning there is a small chance that there will be false positives as there is a chance two distinct Kmers will be hashed into the same sequence. The likelihood of this happening, however, is small; happening < 1% of the time meaning that results are the same or negligibly different. However, Kraken2 performs much quicker than its counterparts, with lower memory consumption meaning that this method is often more optimal.

It was for the above reason Kraken2 was chosen, high level accuracy was not needed because the tool is not classifying metagenomic reads, it's looking at a larger scale for outliers and total accuracy is not required as it assumed the user has in some way generated or curated their dataset prior to their analysis.

Specifically kraken2 was chosen over other tools such as kraken, metaphalan[267], TIPP[268] showed comparable performance, while Kraken2 is much more computationally inexpensive, both in terms of memory and processing time.

By default, RECIPIENT is designed to run with the pre-compiled 'minikraken' database (https://ccb.jhu.edu/software/kraken2/downloads.shtml) which is an 8GB database built from the Refseq bacteria, archaea and viral libraries as well as the GRCh38 human genome. This was decided to be the default as the full Kraken2 database is ~30GB in size, meaning that local users would struggle to fit the data into memory. The configuration file of RECIPIENT however allows the user to point to their Kraken2 database of choice. Similarly, there is an additional workflow step to build a database if so required, however the pre-compiled Kraken2 databases are heavily encouraged to be used.

10.3.3.2 Kmer distance with MASH

Kmer Sketching was also chosen to be included in the pipeline to allow for insights of how much the genomes varied within the dataset. For example, should the user spot certain samples be much more distant than expected then they can remove them. Also, as RECIPIENT relies on the creation of a pangenome, it is also valid to assess whether the genomes being used are close to one another in sequence space.

The Kmer sketching software MASH was chosen for this purpose. MASH was chosen as it supports assembled sequences rather than unassembled fragments. MASH hashes each Kmer into a 32-bit value and then the Jaccard index between each samples Kmer hashes is calculated by randomly sampling the data as follows:

$$J(A,B) = \frac{|A \ \cap \ B|}{|A \ \cup B|}$$

Where A and B represent two sets of genome's Kmers.

This random sampling process is relatively quick, hence why it was chosen for QC in RECIPEINT. MASH RECIPIENT's main goal is predicting and assessing vaccine targets. MASH was shown to be able to compute the pairwise difference between the NCBI Refseq in 33 CPU hours. This total of ~1.5 billion pairwise comparisons[140] massively exceeds the expect amount of comparisons needed for RECIPIENT, meaning that it guarantees completion for most users. While knowing the K-mer distance between samples is helpful in assessing the overall structure of the pangenome and identifying outliers, a more accurate or non-approximation-based method is not necessary.

By default, MASH is run on 40 cores, with a Kmer size of 21 and a maximum number of 1000 hashed Kmers per genome. The input is seeded to ensure reproducibility (however generally speaking the results are expected to converge regardless of seed). DNA stranding is ignored. All parameters can be changed by the user in the configuration file.

RECIPIENT accepts the output of MASH, a triangular matrix in Phylip format and converts it in to an easier to read square matrix in tab-separated value (TSV) format. This is then read into

R and plotted in the Superheat package (https://rlbarter.github.io/superheat/) to plot as a heatmap.

10.3.3.3 Gene and pangenome annotation

10.3.3.3.1 Single genome annotation with Prokka

Genome annotation is the first non-QC step of the pipeline. There are number of genome annotation tools to be used, in RECIPIENT Prokka[251] was chosen.

Another reason Prokka was chosen to be included is that it is very highly used in the microbial genomics community and it is relatively easy to install as all the constituent sub-packages are bundled into it. It also comes prebuilt in Conda, Docker and Singularity giving local users of RECIPIENT more options should they want to avoid installing anything. This rationale meant that it was chosen for genome annotation. The preceding package in the pipeline, Roary is also designed to take the general feature format (GFF) files from Prokka in its pipeline.

In short Prokka annotates a genome and breaks down the contigs into coding sequences of the genome by predicting open reading frames (ORFs) with Prodigal[269]. These putative genes are then assigned a name or function against a database of UniProt[270] proteins, proteins from RefSeq[271], failing that they are given protein family predictions based on Pfam[272] and TIGRFAM[273].

Prokka comes with several parameters that allow for the customisation of annotations based on the user's wants and needs. The only requirement needed by the user is to specify the kingdom their species comes from (namely eukaryote, bacteria, or virus) and optionally what gram stain their bacteria corresponds to, if they are analysing bacteria. The only other options enabled are that tRNA and rRNA will be ignored and certain parameters related to the output are set to remain consistent and avoid errors. There are parameters corresponding to the genome of the sample being annotated but these are intentionally left out to stop users modifying them and stopping the pipeline from being able to annotate. This will increase run time, but that is preferable to poor annotation.

One unfortunate design decision was to leave out SignalP[274] a tool for prediction of signal peptides from the Prokka pipeline. Due to licensing constraints, SignalP cannot to re-distributed in wider packages. Should users want to run SignalP within their pipeline in

Singularity, a Singularity recipe is available at https://github.com/WhalleyT/singularity-recipes which will copy a user's license compliant version of SignalP and make it available to Prokka within the container.

10.3.3.4 Pangenome identification

Pangenome identification was applied by Using Roary[252]. Roary extracts CDS sequences from the supplied Prokka GFF files, translated into protein and filtered to remove truncated sequences. They are then iteratively clustered using CD-HIT[275] and MCL[276] giving a smaller and less redundant set of sequences to work with. These remaining groups are split so that paralogs are in different groups, leaving sets of true orthologs.

Roary was chosen because compared to other pangenomic softwares such as BPGA[277] as like many of the other tools it is relatively quick and efficient. The original publication of Roary notes that 128 samples can be analysed in 1 hour with only 1GB of RAM. This and the fact the tool is relatively easy to install through apt, Docker or Conda means that it is a safe choice in that local users are able to download the software.

RECIPIENT keeps the Roary description schema for how common a gene family is found in the pangenome. It is described below in Table 5.Table 5 Pangenome presence annotations used by RECIPIENT and based on the Roary annotation

Table 5 Pangenome presence annotations used by RECIPIENT and based on the Roary annotation

Name	Percentage of genomes gene is found in
Core	between 99% and 100%
Soft	between 95% and 99%
Shell	between 15% and 95%
Cloud	less than 15%

These data are then passed into R where they are used to generate some plots about the pangenome. Namely the number of genes found in each pangenome category are plotted in a bar plot and pie chart. There is also a heatmap of the presence of each in each sample. This is particularly useful for readers who are interested in finding if certain groups of samples are missing the same gene or not. All of this is performed using the R packages dplyr and ggplot2 (tidyverse.org).

Roary is ran with default parameters, with the option to create a multi-gene alignment switched on. This is performed using PRANK[278]. The multi-gene alignment creates a multiple sequence alignment (MSA) of each gene. These MSA FASTA files are what are passed into the sequence diversity steps (Tajima's D and Shannon entropy).

10.3.3.5 Translation and filtering of the data

Immediately following annotation with Roary, the reference pangenome genes are translated to amino acids using the Biopython SeqTools module[92]. Then extraneous sequences are removed. Extraneous sequences are ORFs that are unlikely to code for whole proteins (determined here by sequences less than 20 amino acids long).

10.3.4 Subcellular location annotation

The reference sequence taken from Roary is then passed into Loctree3[256] for subcellular localisation prediction. Subcellular localisation helps the user in assessing the role and function of a protein, as well as helping inform if it is likely to interact with the immune system. Loctree3 makes use of a support vector machine to classify genes into localisations based on known gene's gene ontology (GO) annotation. The output file, containing the highest probability location and its likelihood scored are parsed with a Python 3 script and read into a TSV file.

Loctree3 was chosen as it was shown to be more performant than its competitors, including Cello[279], PSORTB[280], WOLF-PSORT[281] and YLoc[282]. Again, it was also selected because it comes supplied with a Docker container installation for local users and has been demonstrated to be able to run on a local desktop machine, opening it to local users.

10.3.4.1 Sequence diversity measures

10.3.4.1.1 Tajima's D

There are several tests for evolutionary selection. Perhaps the most popular is Tajima's D [147]. As discussed earlier, Tajima's D is calculated as follows:

$$D = \frac{d}{\sqrt{\hat{V}(d)}}$$

Where d is the difference in two methods of quantifying diversity, the number of segregating sites minus the number of mutations outright and $\hat{V}$ represents the variance. The number of segregating sites is the number of bases in the set of sequences that have more than one DNA base on them.

There are currently several tools that calculate Tajima's D already. However, none were suitable for being included in the RECIPIENT pipeline. Most implementations required some degree of conversion of the sequence data into different formats. Two leading examples are vcf-tools which requires the FASTA file to be converted to variant call format (VCF) format and DendroPy[283] requires the data to be converted into some form of phylogenetic data structure; usually a nexus file or a Newick tree graph. Some other leading population genetics softwares like DivStat[284], MEGA[285] and DnaSP[286] accept FASTA format directly, but require installation of a large package that can only be accessed by a GUI meaning that it is inappropriate for automated analysis.

With this in mind, the Tajima's D was calculated with a custom Python 3 script to reduce run time, simplify error reporting and aid transparency as the script can be packaged with the software allowing the user to read the source code. The script reads a FASTA file, calculates the number of mutations and segregating sites and estimates the variance before calculating D.

The number of segregating sites is calculated by the snippet below:

```
def _calculate_segregating_sites(sequences):

    combos = combinations(sequences, 2)
    indexes = []
    for pair in combos:
        seqA = pair[0]
        seqB = pair[1]
        for idx, (i, j) in enumerate(zip(seqA, seqB)):
            if i != j:
                indexes.append(idx)

    indexes = list(set(indexes))
    S, n = len(indexes), len(sequences)
    denom = 0
    for i in range(1, n):
        denom += (float(1) / float(i))
    return float(S / denom)
```

The calculation of segregating sites makes use of the itertools combination function, to generate of possible pairs of sequences to be iterated through. Then each pair of sequences are checked for differences, the index of these differences is then added to a set object so the number of unique sites can be calculated by taking the length of the object. The combinations function creates a slight memory burden; however, memory usage was kept below 4 GB for 10,000 genomes. If a larger dataset would be used, then typically the user would be using the dataset in an HPC environment and there is much heavier usage used elsewhere in the pipeline.

The function to calculate the number of pairwise mutations is very similar:

```
def _calculate_pairwise(sequences):
    """Calculate pi, number of pairwise differences."""
    for seq in sequences:
        if len(seq) != len(sequences[0]):
            raise("All sequences must have the same length.")

    numseqs = len(sequences)

    num = float(numseqs * (numseqs - 1)) / float(2)

    combos = combinations(sequences, 2)
    counts = []
    for pair in combos:
        seqA = pair[0]
        seqB = pair[1]
        count = sum(1 for a, b in zip(seqA, seqB) if a != b)
        counts.append(count)

    return(float(sum(counts)) / float(num))
```

Variance estimation is the final calculation. Other methods allow for a greater number of assumptions about the data, however in aid of simplicity, the variance was assumed to be coming from a normal distribution and as such calculated as follows:

$$S^2 = \frac{\sum(x_i - \bar{x})^2}{n-1}$$

Which was implemented in standard Python 3 to avoid importing any external libraries. This led to the final calculation of D. Within the script there were extra methods of error checking, for example FASTA files of less than 3 sequences are not calculated, nor are sequences of uneven length. Gapped sequences e.g. those created by an alignment of unequal sequences are calculated, where an insert in the multiple sequence alignment is counted as a mutation/segregating site against standard DNA bases.

If it is assumed that the distribution of Tajima's D scores is Gaussian, then 95% of the data falls within the range [2, -2]. This means that as a default RECIPIENT assumes a gene with a Tajima's D > 2 to be undergoing strong negative selection, meaning that the gene sequence is being conserved. However D has been shown to vary as a result of sample size[287] meaning that the user should introspect on their data and assign a cut off relevant to them. This cut off is modifiable in the pipeline.

10.3.4.2 Shannon Entropy

The aligned sequences are scored for their diversity. Sequence Shannon entropy[288] is calculated on the aligned DNA sequence of each core gene. Shannon Entropy is essentially a measure of information within a given sequence and indicates whether a sequence is diverse or not. It is defined as:

$$H_n(p_i, p_1, p_2 \dots, p_n) = -\sum_{i=1}^{n} p_i \log_b p_i$$

Where n represents the number of possible states a value can take, in this case it is fixed to 4 (representing the 4 DNA bases; A, T, G, C); p is their probability of occurrence and b is the logarithmic base of the user's choosing. In this case we have implemented it at log base 2.

Shannon Entropy is available in a number of different formats, however again in the aim of simplifying the conversion of file formats, a Python 3 script was generated. It read in each sequence in a FASTA file, calculates the entropy at a given position and iterates through every position in the sequence aligned FASTA file:

```
def shannon(column):
    dna_bases, M = set(column),len(column)
    entropy_list = []
    # Number of residues in column
    for base in dna_bases:
        number_of_bases = column.count(base)
        probability = number_of_bases/float(M)
        entropy = probability * (math.log(probability,2))
        entropy_list.append(entropy)
return -(sum(entropy_list))
```

This is then averaged across the entire sequence to give the entropy of the sequence.

10.3.4.3 Structural profiling

In order to ascertain more information about the protein, coded for by each gene, they are searched against all of the sequences found in the PDB. In order to do this, first the PyPDB API[289] was used to get a list of all PDB files found in the PDB at the time of analysis. These IDs were used to create the relevant link to the FASTA file (for example https://www.rcsb.org/fasta/entry/<PDB_FILE>) which was the downloaded using the Linux package Wget (https://www.gnu.org/software/wget/). Upon completion these files were combined to make one FASTA file.

In order for these sequences to be searched against, the DIAMOND aligner[290] was used to create a database to search against. DIAMOND was chosen for its speed in performance, it is up to 20,000 times quicker than BLAST, whilst maintaining comparable performance in alignment. The database was created using default parameters. Then, the reference protein sequence of each gene in the pangenome was searched against the database, using a BLOSUM62 matrix and the highest sensitivity alignment.

The highest scoring output based on percentage identity (or “pident” as it is referred to in the software) was then taken forward for analysis, assuming this was above a cut off of 98% similarity.

The corresponding PDB file is then parsed from the ID of the DIAMOND output and downloaded using wget. The corresponding matching chain is then also extracted in the same way using a Python 3 script.

From here the PDB file is passed into the BioStructMap[259] package, where a number of parameters are calculated. These include the Kyte and Doolittle index of hydrophobicity; the normalised flexibility, the Hopp and Wood hydrophilicity; the Emini surface fractional probability and the Janin interior to surface transfer energy scale. These results are somewhat arbitrary as they are aligned against potentially a slightly different protein sequence, however they may prove useful in identifying specific sites. These results are then written to a PDB file where the parameters for each residue replace the B-factor value. This is a common method for visualising parameters over the residues in a protein structure. Due to many users having a preferred visualisation software, the output is a PDB file and not an image, allowing the user to create publication quality images using PyMOL[91], CCP4MG[90] or Chimera[100].

10.3.4.4 Physiochemical properties

As well as structural parameters, the sequence level data is used to calculate physiochemical properties. This is done using the Peptides[258] package in R. This is done by passing the reference sequence from the pangenome into a text file. The exact scores are shown in Table 6.

The physiochemical parameters are taken as the mean across the entire protein sequence to aid user readability, as the output document would be too large otherwise and would detract from readability for the user.

THMM is also used to predict transmembrane helices. Transmembrane helices are often conducive to a protein being difficult to purify experimentally, this is not advantageous to vaccine design[291]. THMM uses an HMM to detect motives associated with transmembrane helixes and predict sequences likely to contain them.

Molecular weight is also counted for each reference sequence using the Biopython ProtParam module.

Table 6 A list of the parameters used in the physiochemical property analysis, along with a description of what they are and the reasoning of why they are included.

Parameter	Description	Reasoning
Juretic hydrophobicity	Hydrophobicity scale reasoned through preference function	Important for stabilisation of protein for vaccine
PP1	PCA parameter based on polarity	Important for stabilisation of protein transport
PP2	PCA parameter based on hydrophobicity	Important for stabilisation of protein for vaccine
PP3	PCA parameter based on hydrogen bonding	Hydrogen bonding propensity is important for antigen recognition
Kidera factor 1	Helix formation	Shown to be descriptive of TCR and Antibody recognition
Kidera factor 2	Side chain size	Shown to be descriptive of TCR and Antibody recognition
Kidera factor 3	Extended structure preference	Shown to be descriptive of TCR and Antibody recognition
Kidera factor 4	Hydrophobicity	Shown to be descriptive of TCR and Antibody recognition
Kidera factor 5	Double bend preference	Shown to be descriptive of TCR and Antibody recognition
Kidera factor 6	Partial specific volume	Shown to be descriptive of TCR and Antibody recognition
Kidera factor 7	Flat extended preference	Shown to be descriptive of TCR and Antibody recognition
Kidera factor 8	Occurrence in alpha region	Shown to be descriptive of TCR and Antibody recognition

Kidera factor 9	pK-C	Shown to be descriptive of TCR and Antibody recognition
Kidera factor 10	Surrounding hydrophobicity	Shown to be descriptive of TCR and Antibody recognition
Z1	Lipophilicity	Membrane transport
Z2	Steric properties	Spatial presence/availability of residues
Z3	Electronic properties	Charged pockets important for interaction
Z4	electronegativity	Important for transport
Z5	electronegativity	Important for transport

10.3.4.5 Immunological

A number of immunological parameters are calculated. Instead of allowing the user to specify their HLA alleles of interest. The softwares netMHCpan[253] and netMHCIIpan[292]. The netMHCpan family use neural networks to predict whether a peptide will not bind MHC, bind weakly or bind strongly. By default, it does so across a range of HLA alleles that are said to be representative of the wider population, therefore avoiding the lack of broad applicability in some RV methods. A python script was generated to parse the outputs and count the number of strong and weak binders for each gene and read them to a TSV file.

Similarly, proteasomal cleavage was predicted for the peptides using a similar method employed by netChop[255]. The output was also cleaved such that the number of cleavage sites was predicted as well as counting the peptides cleaved. This was done using a Python 3 script.

Finally linear B-cell epitopes were predicted using Bepipred-2[152]. Like above, this also employs a neural network method to predict residues which will be recognised by B-cells in a linear fashion. A python 3 script was used to parse the output of the number of epitope residues per gene and was outputted into a TSV file.

The percentage of conservation of each antigen was also calculated. The pangenomic reference sequence was used to predict MHC I and II and B-cell binding. The epitopes predicted by this were then taken and matched to their corresponding sites in the alignment file containing individual samples. For MHC binding prediction a percentage was calculated for how many samples had the exact same matching peptide sequence. For B-cell epitopes this was calculated on a residue-by-residue basis as some linear epitopes could long (e.g. >50 amino acids in length) so calculating their conservation across the whole site would be difficult.

10.3.4.6 Gene essentiality and virulence factor identification

A FASTA file of essential genes was downloaded from the Gene Essentiality Database[261], as was one of virulence factors from the Virulence Factor Database[262]. Using the same process as described in 10.3.4.3 these two DNA databases were converted into a DIAMOND database, before the original DNA reference sequences in the pangenome were searched for. If there was a match with 98% similarity the gene was determined to be essential or virulent, regardless of the species of origin. The species was ignored because there is no guarantee that the other is using one particular species or subspecies and also that confuses user interpretation as to some extent this is dependent on how well annotated the databases are in the first place.

10.3.4.7 Outputs

Throughout the running of the pipeline, Nextflow automatically copies the results of each process into a parent directory, meaning that the user is able to explore the raw data themselves. On top of this, the tabulated data is read into R where it is used to generate a summary document of the parameters using Rmarkdown. The data summarised in the document are the Tajima's D; the predicted MHC I and I binding percentage; proteasomal cleavage; the physiochemical properties discussed in 10.3.4.4; the subcellular location prediction and the presence in the core genome.

The above data are both plotted using the ggplot2 library and also summarized both as a master document with interactive tables in the Rmarkdown document using the DT library, as well as a hard-coded output in TSV format. The Structure profiled PDB files are left out of the summary document as it is difficult to get a production ready image without prior knowledge about the protein. Instead, the output gives a set of PDB files with their B-factor overwritten with the scores of each analysis.

10.4 BENCHMARKING

Benchmarking was performed across all of the four datasets. This was done in order to demonstrate the scalability of the tool in terms of genomes analysed. On top of measuring outright usage in CPU usage, the Nextflow tracing feature, which tracks CPU, I/O and RAM usage across each process was also used to identify bottlenecks.

The tool was timed using 40 cores on multithreaded processes, with a total of 15 jobs available at one time. It was benchmarked on the Advanced Research Computing @ Cardiff University (ARCCA) Raven supercomputer.

10.5 AVAILABILITY AND LICENSING

The RECIPIENT pipeline is available at https://github.com/whalleyt/recipient. It is released under the GNU GPL V3 license. Where possible the tool makes use of Singularity containers. Due to licensing constraints to tools affiliated with the Technical University of Denmark, some of these tools have to be downloaded. If they are downloaded and the license is agreed for by the user, they are able to use the Singularity recipe scripts to generate their own Singularity containers as part of the pipeline at https://github.com/whalleyt/singularity_recipes

11 RESULTS: RECIPIENT- A PIPELINE FOR PANGENOME REVERSE VACCINOLOGY

11.1 EPEC

11.1.1 Pangenome identification

Roary predicted that the core and soft core pangenome was accounted for by 3542 of the total 13,698 available genes (Figure 1). The soft core accounted for 1243 genes and the core accounted for 2299 genes. However, only genes known to be specific to EPEC, as discussed in 9.3 were considered to avoid targeting commensal *E. coli* genes.

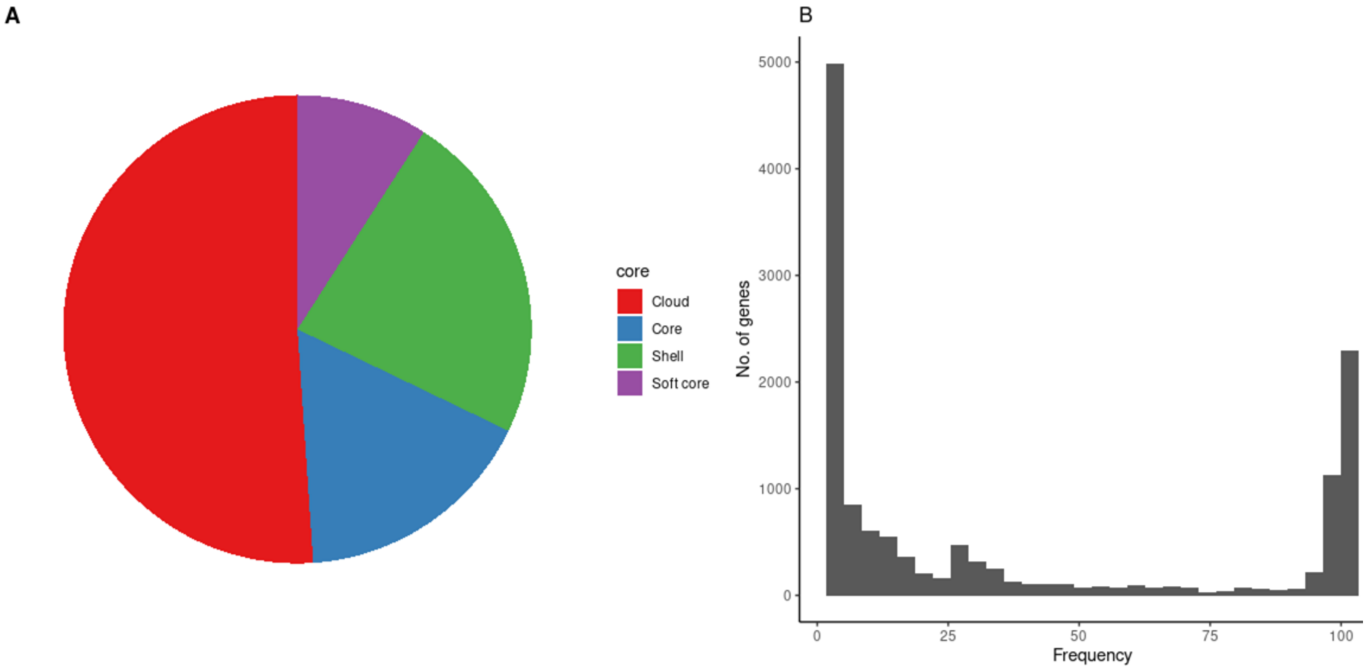


Figure 39 The identification of pangenome components corresponding to the EPEC genomes. A. shows a pie chart of the categorisation of the pangenome into the core, soft core, shell and cloud genome. B.) shows a histogram of the percentage a given gene is present across all genomes.

The EPEC pangenome creation was not successful in fully identifying different key EPEC genes. For example, some the gene products of the LEE locus (discussed in 9.3) were not successfully identified. Their presence in the pangenome is summarised below in Table 7. The Prokka pipeline was unable to identify several genes encoded for by LEE, namely EspA, EspB, EspD, EspF, EspG, EspH and EspZ, however EspC, EspP, Tir and Map were identified.

This was not rectified by any parameter changes during genome annotation and pangenomic analysis, including specifying the databases used by Prokka using the --*Genus* and –*Species* parameters, nor was it affected by raising and lower the E value parameter, the parameter used to determine a similarity cut off in BLAST searches. A second group of EPEC genomes from a study by *Hazen et al.*[200] was downloaded (accession numbers can be found in supplementary table S6) and the same issue arisen. Basic QC was performed: Kraken2 was used to check for contamination, MASH k-mer distance was used to assess genomic distance between samples and SQUAT was used to assess the quality of the assemblies and there were no issues discovered with the quality of the assemblies[293].

Table 7 Identification of Genes known as EPEC effectors, NA represents the Prokka pipeline being unable to annotate the gene, whilst any numeric values represent the percentage of genomes the gene was found to be in.

gene	presence in pangenome (%)
EspA	NA
EspB	NA
EspC	28.1
EspD	NA
EspF	NA
EspG	NA
EspH	NA
EspP	22.8
EspZ	NA
Tir	49.1
Map	100

However, there were multiple genes found on the LEE locus that were successfully mapped to the pangenome, albeit in lower levels than expected. The gene encoding for the Map effector protein was present in 100% of the samples, meaning that it was the only core gene present in the pangenome coming from the LEE locus. This is most likely a reflection of the quality of the Prokka annotation, so the other LEE gene products were still considered to be of interest, however this denotes an obvious flaw in working with lesser studied genomes, as the user may not have the prior knowledge necessary to make such a decision.

Of the EAF encoded genes, only BfpB was successfully annotated. It was present in 8.8% of the EPEC genomes. Due to the distinction of typical and atypical EPEC, this was not expected to be a member of the core genome. The major structural subunit of the BFP, bfpA was not identified at all. The second operon encoded on the EAF plasmid, the plasmid encoded regulator (Per) has three gene products perA, perB and perC and none were identified by Prokka. It has been noted that Prokka is not optimal at detecting plasmids from genome assemblies using the default settings (https://github.com/tseemann/prokka/issues/319). A wider issue surrounding plasmids is that a number of de novo assembly pipelines are not optimised to detect plasmids. For example, SPAdes[294], the software used to assemble the EPEC FASTA files has a sister software specifically to aid in detecting plasmids[295].

Finally, the annotations were checked for the presence of Nles. Two were successfully identified and annotated by Prokka. NleF was present in 64.9% of the pangenome and EspFU was present in 49.1% of the pangenome.

11.1.2 Tajima's D

Tajima's D was calculated for all genes in the EPEC pangenome. A number of key EPEC genes appeared to be undergoing negative selection based on their Tajima's D value (Table 8). Here a positive D value indicates negative selection.

Table 8 Tajima's D values for successfully annotated members of the EPEC effectors.

gene	Tajima's D
espC	0.2
espFU	24.7
espP	16.8
Tir	20.2
Map	3.8
BfpB	0.0
NleF	4.6

With the exception of two genes, all the successfully annotated effectors were undergoing negative selection based on their Tajima's D, meaning that there was a scarcity of rare alleles. There appears to be no other attempts in the literature at measuring Tajima's D in

EPEC genes. However, given the effector function of several of these genes, one might expect conservation and lack of rare alleles in the functional areas of the effectors.

11.1.3 Immune response prediction

As discussed earlier in Chapter 9 there are a multitude of adaptive immune responses from EPEC. Therefore, responses from CD8 T-cells, CD4 T-cells (by proxy by means of MHC I and II binding affinity) and B-cells were considered. The total number of 9-mer peptides predicted to bind MHC class I; the total number of 15-mer peptides predicted to bind MHC class II and the total number of unique peptides predicted to be linear B-cell epitopes as well as the total number of amino acids predicted to be part of the paratope (Table 9). A number of the potential genes of interest had a strong predicted immune response.

Table 9 The total number of good MHC I and II binders, the total number of linear B-cell epitopes, the total number of Amino acid residues predicted to be part of the paratope and the sequence length of the potential target gene.

gene	no. good MHC I binders	no. good MHC II binders	no. B-cell epitopes	total no. residues available B-cell	sequence length
espC	15	37	27	1007	1242
espFU	2	0	1	200	206
espP	16	35	23	1073	1326
Tir	4	16	16	389	537
Map	7	4	12	81	124
BfpB	14	10	12	363	538
NleF	3	7	9	118	188

Interestingly, the preference of MHC class I to MHC class II epitopes was relatively evenly split between the 7 genes, with 4 having more predicted MHC class II peptides and the remaining 3 having a stronger predicted preference for MHC class I. Despite some genes having relatively few good MHC binders predicted, these MHC binders were very highly conserved across the pangenome, suggesting that there was not a selection pressure to diversify. There were a total of 172 peptides predicted to be good MHC I or II binders of those only 13 were not conserved across every sample in the EPEC cohort based on a cut-off of >80% of the sequences containing that peptide. Based on the EPEC cohort at hand, this suggests that a CD4 or CD8 T-cell mediated immune response to these predicted peptides would be well conserved across all samples.

EspA, B and D all had known epitopes in the IEDB, however none from espC, FU and P were found. This could warrant further investigation. There were also no known epitopes mapped to BfPB, NleF or Map. One Tir peptide LTGGSNSAVNTSNNPPAP has already been described[296], but the sequence was not found in any of the sequences for Tir suggesting that this peptide is not well conserved.

11.1.4 Subcellular localisation prediction

The EPEC dataset and the set of known EPEC effectors gave a good chance to validate the effectiveness of subcellular localisation prediction on an observational level. These effector proteins are all relatively well-described and as such, have had their subcellular localisation described in all but one case (Map) as shown in Table 10.

Only the annotation for EspFU, and Tir were successfully annotated such that they matched the Uniprot annotation. The nature of RECIPIENT means that it is designed such that the user may not be working on a well-annotated pathogen, nor are they able to manually corroborate all of their predictions. However, it demonstrates a potential need for non-predictive methods for assigning subcellular localisation.

Table 10 Subcellular localisation as described by experimental validation, taken from Uniprot and LocTree3 subcellular localisation prediction.

gene	localisation prediction	experimentally observed localisation
espC	Unknown	Secreted
espFU	Host Associated	Secreted, host cytoplasm

espP	Unknown	Cell surface, secreted, periplasm
Tir	Host Associated	Secreted, host cell membrane
Map	Cytoplasmic	Not annotated
BfpB	Unknown	Outer membrane
NleF	Cytoplasmic	Secreted, host cytoplasm

11.2 *Salmonella typhi*

Unlike EPEC, which had a small number of known effectors which distinguished it from other commensal strains, *S.* Typhi is a member of a wholly pathogenic genus meaning that a different approach to target identification was needed. Whilst the small number of EPEC effectors meant that they could be searched for based on *a priori* knowledge, *S.* Typhi's entire genome could be viewed as a potential target. This meant that candidates were filtered based on their essentiality to the pangenome, their selection based on Tajima's D, subcellular location and predicted immunogenicity. As discussed in chapter 9.2, there is a strong CD4 T-cell mediated response to *S.* Typhi*,* so preference was placed on MHC class II binding predictions.

11.2.1 Pangenome identification

The pangenome of *S.* Typhi showed a similar structure to the pangenome of EPEC. The majority of genes fell either into the cloud pangenome or the core genome (47.2% and 45.2% respectively) meaning that most genes were either found in > 99% of the 785 samples or < 15% of them (Figure 40).

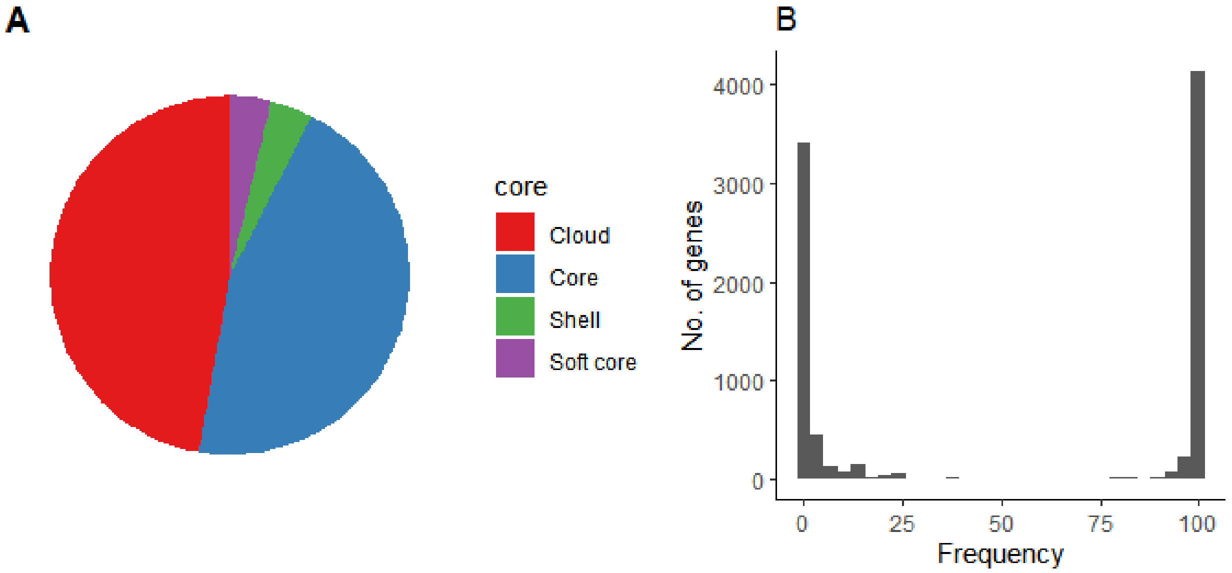


Figure 40 The identification of pangenome components corresponding to the S. Typhi genomes. A. shows a pie chart of the categorisation of the pangenome into the core, soft core, shell and cloud genomes. B. shows a histogram of the percentage a given gene is present.

The strong polarisation of the pangenome to being either core or cloud meant that a large number of genes could be discounted straight away as they were so rarely found across the pangenome that any targeting of them would be ineffective. Another factor that limited the scope of the pangenome annotation was that a large number of the genes annotated were of unknown function and were annotated as hypothetical genes, predicted through homology or ORF prediction. The majority of these also products had no gene label attached to them, meaning that they were predicted by functional group. There were 8915 annotated genes annotated by Prokka, of which there were only 609 that were not hypothetical genes.

Genes were filtered on the genome annotation and the presence of the gene in the pangenome. Any genes labelled “hypothetical” or “putative” were discounted. Furthermore, a cut off of 90% presence in the pangenome was thresholded. This left 3296 potential genes of interest.

11.2.2 Tajima's D

Tajima's D was widely distributed among the candidate genes. As the kernel density plot shows in Figure 41, most genes were not undergoing positive or negative selection; 2173 of the genes' D values were such that they satisfied |D| < 2 meaning that they were not undergoing positive or negative selection. Of the remaining genes, 207 were undergoing negative selection (positive D) and 154 were undergoing positive selection (negative D). 762 genes were either too short or were not fully aligned, meaning their D calculation was skipped.

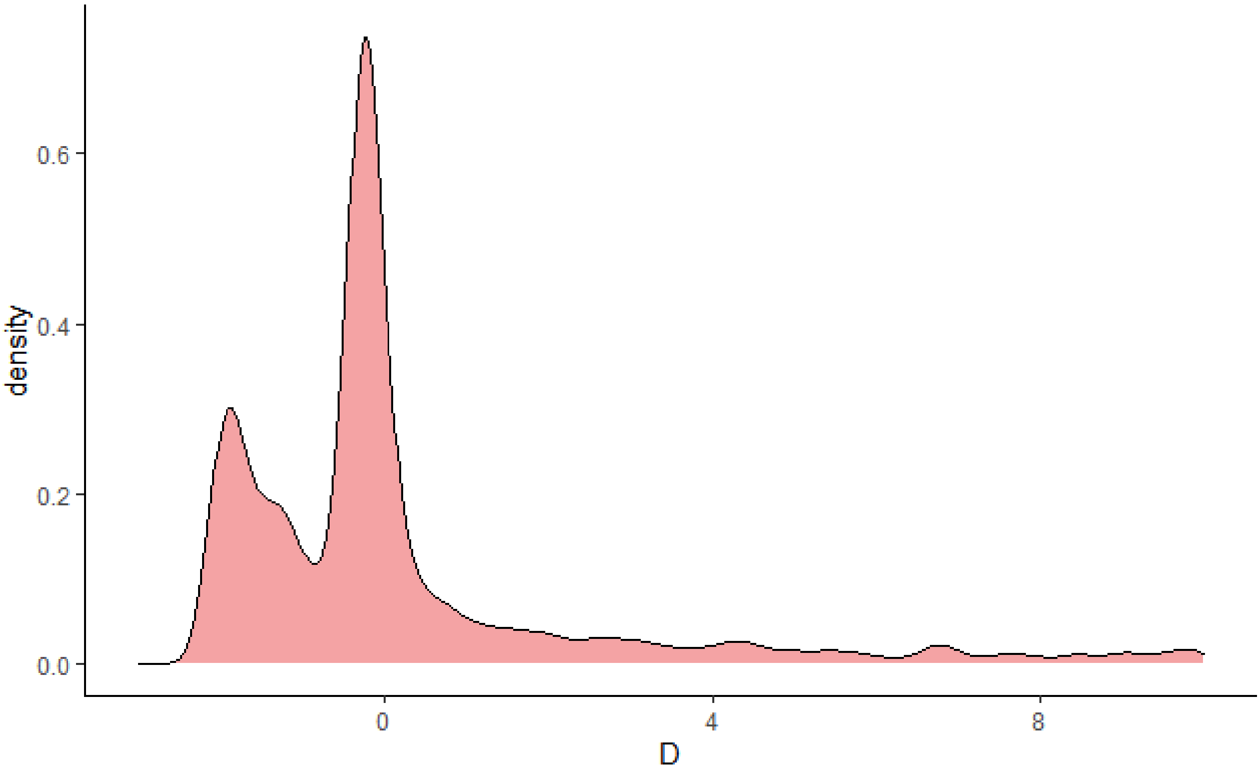


Figure 41 Kernel density plot of Tajima's D for all genes in the S. Typhi pangenome.

Therefore, a cut off of D > 2 was determined in order to filter out genes not undergoing negative selection, leaving 207 genes.

11.2.3 Subcellular localisation prediction

The outer membrane of *Salmonella* is thought to possess more antigens. This does not appear to be mediated through antigenicity directly, meaning that searching solely for good MHC binders would not find these targets. Therefore, gene products predicted to on the outer membrane were preferentially searched for. Based on the annotation predictions “extracellular”, “outer membrane”, “flagellar” “T3SS”, “host associated” and “fimbriael” were considered. The breakdown of subcellular location for genes found in > 90% of the pangenome, not assigned hypothetical and with D > 2 is described in Figure 42.

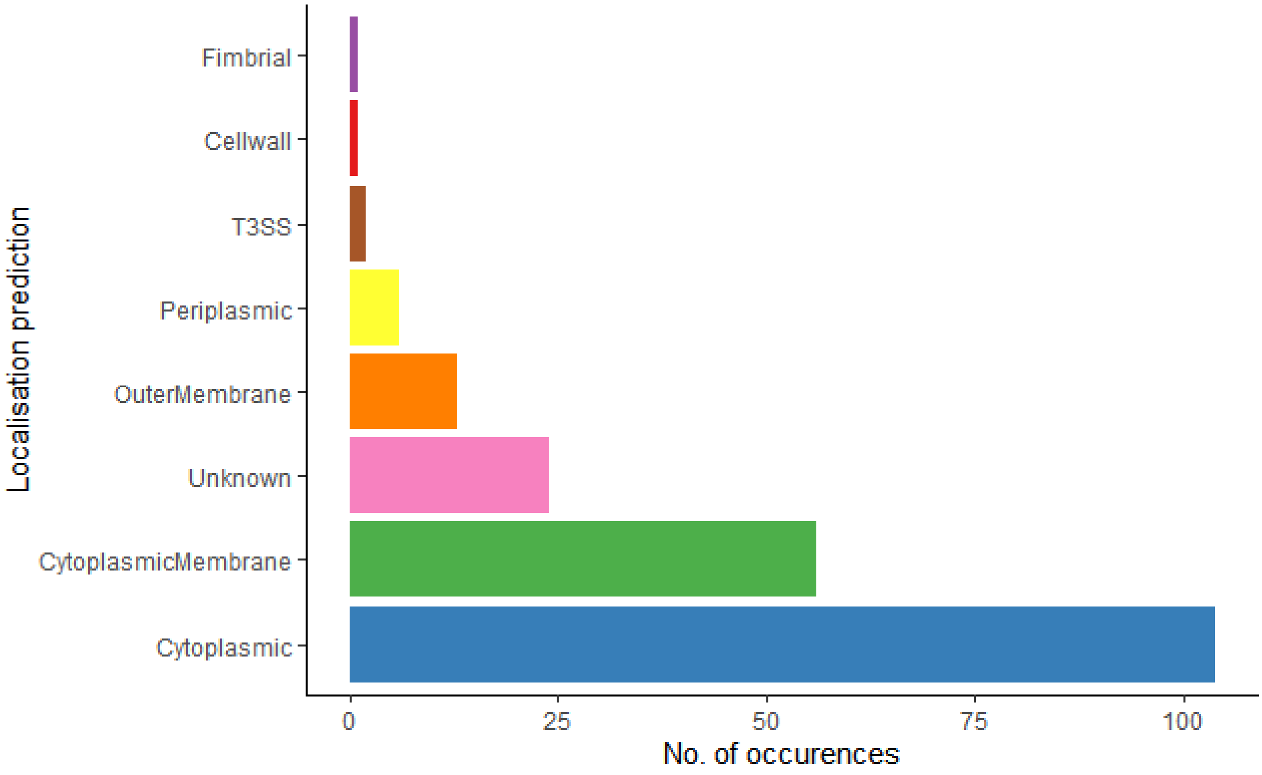


Figure 42 Subcellular location prediction by psortB for the filtered list of candidate genes.

Once filtering of the six subcellular localisations described above was applied then 15 potential genes of interest remained. Of the 15 extracellular genes predicted by psortB, 2 were predicted to be from the T3SS, 1 was predicted to be fimbriael and the remaining 12 were predicted to be from the outer membrane. These, along with their manually collected annotation from Uniprot are summarised in Table 11 predicted and manually curated for the 15 proteins predicted to be on the outer membrane for Salmonella Typhi, following filtering on pangenome presence and Tajima's D. All were the same as their manual annotation.

As discussed in the psortB paper[280], precision (e.g. the ratio of true positives to the total number of true and false positives) is high, meaning that while performance may be poor in assigning a category (e.g. lots of “unknown” predictions) there is a relatively low amount of false positives.

Table 11 predicted and manually curated for the 15 proteins predicted to be on the outer membrane for Salmonella Typhi, following filtering on pangenome presence and Tajima's D.

gene	subcellular localisation prediction	manually curated subcellular location
sefA	Fimbriael	Fimbriael
bamD	Outer membrane	Outer membrane
Caf1A	Outer membrane	Outer membrane (multi-pass)
fimD	Outer membrane	Outer membrane
kdgM	Outer membrane	Outer membrane
lptE	Outer membrane	Outer membrane
ompF	Outer membrane	Outer membrane (multi-pass)
pldA	Outer membrane	Outer membrane (multi-pass)
rcsF	Outer membrane	Outer membrane; lipid anchor
sadA	Outer membrane	Outer membrane; cell surface
ttgI	Outer membrane	Outer membrane

yehB	Outer membrane	Outer membrane
yfaL	Outer membrane	Outer membrane; secreted; cell surface
yscJ	T3SS	T3SS
yscU	T3SS	T3SS

11.2.4 Predicted MHC binding

As discussed in chapter 9, there is a wealth of literature suggesting there is a strong CD4 T-cell response to *Salmonella* infection. Therefore, predicted MHC class II binding prediction was used preferentially over MHC class I or B-cell mediated response. The MHC binding prediction was calculated based on the pangenome reference sequence, that is, a single sequence taken from the collection of sequences of that gene that are representative of all of them. Therefore, the presence of each peptide was counted across each individual sample. The intention of this was to give an indication of how well conserved that particular site was. This is summarised in Figure 43, where a peptide was considered conserved should it appear in 90% of its parent sequences.

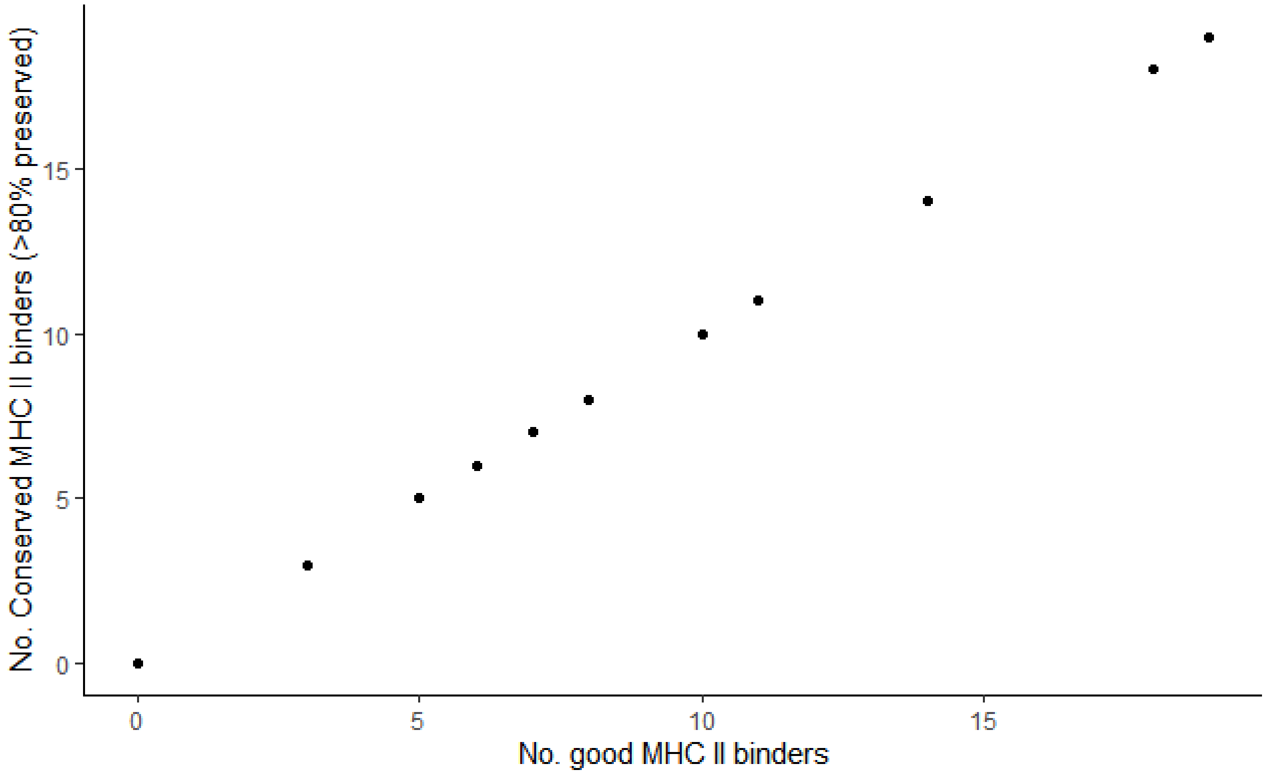


Figure 43 43 The number of predicted good MHC class II binders plotted against the total number of well-conserved good MHC class II binders, based on a binary classification of the peptide being found in >80% of all sequences for a given gene.

Apart from 3 genes, which no good MHC class II binders (yfaL, sefA and rcsF) the remaining genes had all of their predicted MHC class II binders highly conserved. yfaL, sefA and rcsF were then removed. The percentage of conservation for each individual peptide of the remaining genes is shown in Figure 44. Every peptide is well conserved (present in greater than 80% of sequences) and a large number (76 peptides out of 112) were very highly conserved (present in greater than 99% of sequences).

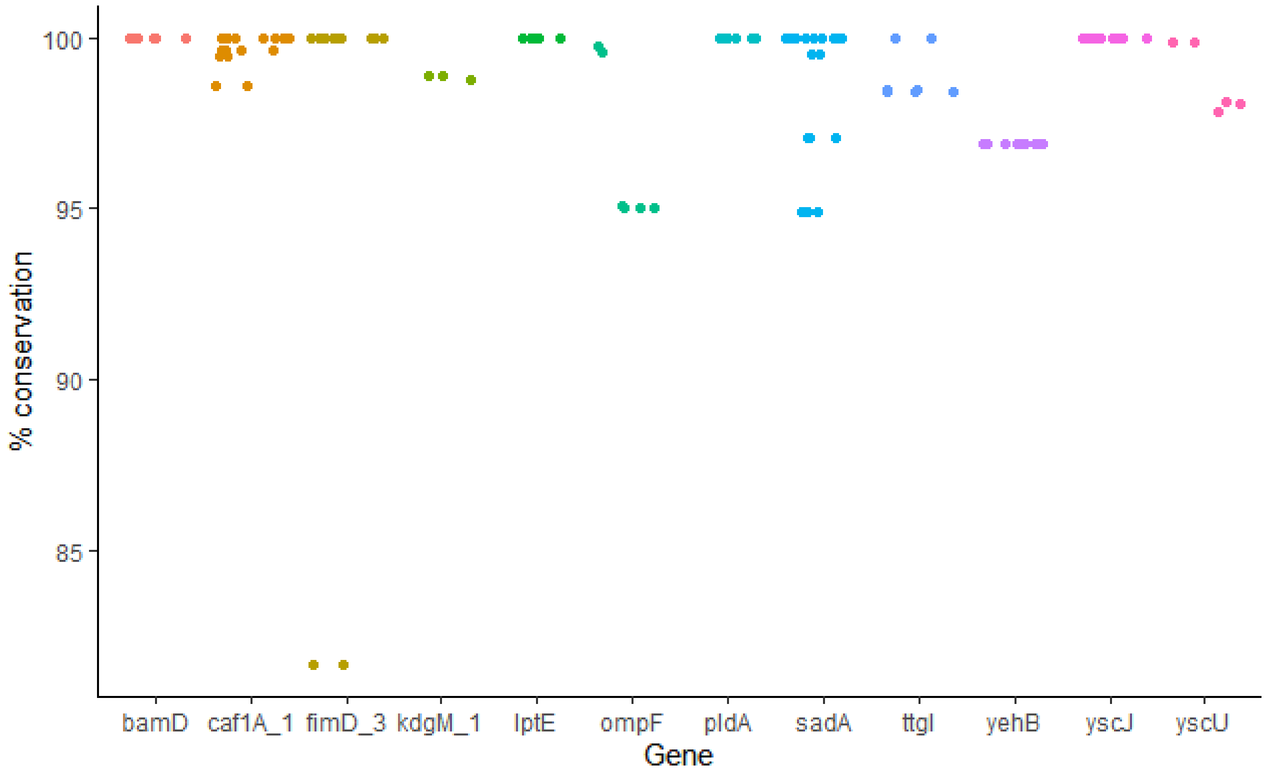


Figure 44 Percentage conservation of each peptide predicted to strongly bind MHC class II from the Salmonella Typhi pangenome.

All of the remaining proteins satisfy a number of conditions which would be conducive to the generation of an immune response. They are strongly conserved in the pangenome, they are relatively well-understood in that they are known genes with meaningful annotations; they have a number of conserved peptides which are predicted to bind MHC class II; they come from proteins undergoing negative selection meaning that they are unlikely to diversify, and they are proteins which come from the outer membrane of *Salmonella* meaning that they are from a preferentially targeted site. I will now discuss any pre-existing literature surrounding these genes as targets for immune response.

There is little information known about any immune response to bamD. Other RV pipelines have suggested members of the bam family to be good potential targets (for example bamA in *Moraxella catarrhalis*[297] and bamC in *Neisseria meningitidis*[298]*)*. However, outside of that there is little information pertaining to whether it elicits an immune response.

Caf1A is an outer membrane usher protein, which aids in localising the capsular antigen caf1 which is an important mediator in the inhibition of phagocytosis in *Yersinia pestis*[299]*.* Its role in *Salmonella Typhi* has not been well described. It has been shown that over-expressing

Caf1 is able to attenuate *Salmonella* Typhimurium[300] but this is not thought to be immune modulated and this was performed by a vector being transferred, so there is still no evidence that it is expressed in *Salmonella*.

FimD is a member of the fim fimbriael cluster. Together the fim cluster is involved in the biogenesis and structure of type 1 fimbriae (T1F). T1F is involved in adhesion, leading to the recognition and binding of high-mannose oligosaccharides which are carried and expressed on the host surface. T1F is then used to colonise the host cell. FimD is involved in chaperoning the T1F complex[301]. Other RV methods have also predicted fimD as a potential immune target in *Acinetobacter baumannii*[302] and *Acinetobacter nosocomialis*[303]. Again, however no functional insights have been followed up.

kdgM appears to have very literature in *Salmonella* specifically. Functionally, most information on kdgM is based on information gained studying *Erwinia chrysanthhemi* where it aids in the secretion of pectinases, which degrade the pectin polymers found in plant cell walls[304]. Thus, it is unlikely to be active during infection of human cells.

LptE is a protein that aids in the formation of the LptD, a protein which in turn is involved in the transport of LPS. It is found in most Gram negative bacteria[305]. LptE has not been directly implicated in any immunological studies or RV studies. However, lptD has been predicted in a number of Reverse vaccinology pipelines[297,306,307]. Also, in *Vibrio* species lptD has been implicated in being a potential vaccine antigen due to its high immunogenicity[308]. The exact mechanism of this remains unclear, but due to lptF's close proximity in interacting with lptD it may also be a fruitful target.

OmpF has been shown to be a highly protective antigen which is immunogenic and can stimulate both an innate and adaptive response without the need for any exogenous adjuvants. Response to them appears to follow a Th1/Th17 helper cell profile[309]. It has also been shown that ompF successfully could induce a sustained immune response in *Salmonella* Typhi[310]

PldA has not been described in *Salmonella* with any detail. In *Campylobacter coli* it has been shown to play a role in cell-associated haemolysis, destruction of red blood cells. It has been shown that LPS production is dependent on pldA. This is a result of pldA generating fatty acids in the outer membrane of the cell, these fatty acid products are processed where eventually they are used to produce LPS[311]. This is found in a large number of Gram negative bacteria meaning that it is possible that immune targets to pdlA would impact commensal bacteria. A number of the genes were found in non-*Salmonella* species, many of which were commensals (e.g. *E. coli*).

Table 12 Table of the peptides predicted for pldA and whether they are unique to Salmonella or are found in other species.

gene	unique to *Salmonella*
WNRLYTRLMAENGNW	No
EVKFQLSLAFPLWRG	No
LKIGYHLGEAVLSAK	No, but only matches are in pathogenic species
KIGYHLGEAVLSAKG	Yes
DEVKFQLSLAFPLWR	No
IGYHLGEAVLSAKGQ	Yes
SWNRLYTRLMAENGN	No
NRLYTRLMAENGNWL	No

As shown in Table 12 most of the peptides predicted in pldA were found in other species. However, 2 were unique to *Salmonella* and 1 was only found in *Shigella*, a pathovar of *E. coli.* These 3 peptides still could be involved in conferring immunity to *Salmonella* Typhi*.*

The sadA autotransporter is known to illicit an antibody mediated response via IgG[312]. Currently a T-cell mediated response has not been demonstrated. Linear B-cell epitope prediction using Bepipred did show that there were a large number of linear epitopes on sadA; with only 15 amino acids not being predicted to be part of a linear epitope, suggesting it is highly immunogenic.

ttgI is a gene encoded for a toluene efflux protein, it is involved in solvent tolerance in Gram negative bacteria. There is no external evidence to suggest that it is immunogenic. It is found in a number of Gram negative bacteria. As Table 13 shows none of these peptides are unique to *Salmonella.*

Table 13 Table of peptides predicted for ttgI and whether they are unique to Salmonella or are found in other species.

gene	unique to *Salmonella*
LMAFLQQDALHLSDL	No
ESSLSSIDAAKAAFY	No
VTARIGAVKAREAEQ	No
MAFLQQDALHLSDLF	No
IESSLSSIDAAKAAF	No
TARIGAVKAREAEQE	No
NLMAFLQQDALHLSD	No

yehB is a protein inferred from homology, based on similarity to a protein found in *E. coli* (https://www.uniprot.org/uniprot/P33341). Due to having no functional characterisation, being found in non-pathogenic bacteria species and having no literature surrounding immunogenicity it was not considered further.

yscJ and yscU are both involved in the formation of the Ysc-Yop T3SS, allowing Yop effector proteins to be injected into the host cytosol where they proceed to interfere with host innate immune response[313]. There appears to be no direct characterisation of this system in *Salmonella* nor has there been any discussion on the immunogenicity of its components.

11.2.4.1 Verification of known targets

There are a number of known genes that have epitopes that cause an immune response in response to *S.* Typhi infection. To this end the genes HlyE[314], AhpC[315], EutC[315] and ompC[316] have all been shown to illicit a CD4 response. They did not appear in the final list of peptides, so in order to gain further insights into RECIPIENT's strengths and weaknesses they were searched for in order to try and gain a rationalisation for how to best channel the output information.

OmpC was filtered out because it is undergoing positive selection (D = -0.2). It has a relatively low number of predicted MHC class II antigens at 4, but they are all 100% conserved across the pangenome. The ompC gene itself was found in 100% of the genome and it was successfully predicted to code for a protein that would localise to the outer membrane. All these facets are positives that would suggest ompC would make a good immune target. The fact that ompC is not under strong purifying selection but still has conserved MHC II epitopes gives cause for a sliding window Tajima's D to be incorporated and for preservation of each peptide to be calculated and used as a filtering step with preference over Tajima's D, however this step was computationally expensive.

A similar reason led to the exclusion of ahpC and eutC, both were annotated as cytoplasmic proteins which are less likely to illicit an immune response, however as discussed, while not as strongly immunogenic as the outer membrane proteins, they are still capable of inducing an immune response.

11.3 *SARS*-CoV-2 Dataset

11.3.1 Pangenome identification

For the SARS-CoV-2 dataset, 46 targets were identified, with most consisting of gene families. Of these 46 targets, only 8 were conserved enough to be considered "shell" genes in the pangenome. It has previously been noted that Prokka was first used as a bacterial annotation tool. Whilst Prokka is still performant on bacteria it is outperformed by other virus specific annotation softwares, for example Vgas[317] and VAPiD[318].

11.3.2 Tajima's D and immune prediction

Only Tajima's D, MHC I and II binding and BCR epitopes were predicted as there is no tool for the prediction of viral subcellular location. Two genes were undergoing strong negative selection according to their Tajima's D; gene 1a, the protease of SARS-CoV-2 and gene 9b, corresponding to ORF1ab.

Table 14 the 2 identified conserved genes with strong negative selection

Gene	D	BCR epitopes	MHC I good binders	MHC II good binders
9b	7.092023	0	7	0
1a	2.228861	98	109	108

Interestingly, gene 1a has no annotation in Prokka, but the DIAMOND search step revealed that there was a 100% match with the protein structure with PDB ID *5RE7*, identified as the SARS-CoV-2 main protease. The paper remains unpublished at the time of writing. The Prokka databases may not have been updated at the time of analysis or alternatively Prokka simply did not annotate the sequence correctly. Neither was determined to be essential for survival or a virulence factor when scanning against the DEG and VFDB databases.

Gene 9b had no linear epitopes of at least 5 amino acids in length and had no predicted good MHC class II binders. It did however have 7 MHC class I binders. However, none of them were well conserved across the pangenome.

All of 1a's B-cell epitopes and MHC binders were well conserved. However, there are no known immune targets corresponding to ORF1ab in the literature that match those predicted by RECIPIENT. A number of B-cell epitopes appear to have been predicted in another *in silico* analysis, however none of these sequences have been made available to be verified[319].

Gene 1a has been discussed previously as a drug target for treatment of COVID-19[320], but very little exists in terms of immune targeting. The protein is dissimilar to human proteases, suggesting that it could possibly be a good immune target[321]. It has been detected to have localised in the nucleus, ER and cytosol of host cells[322], suggesting that peptide fragments may be presented.

11.3.3 Structural analysis

The *5RE7* structure was analysed with Biostructmap. The hydrophobicity and surface availability of the residues were calculated in Biostructmap. As Figure 45 shows, there is a large pocket of highly accessible and low hydrophobicity on the protease (shown at the top from the reader's view) centred around residue 83. High availability of this region could make it more available to access by antibodies. The low hydrophobicity of the area also means that it may remain accessible in solution. The region of availability did not match any of the predicted epitopes.

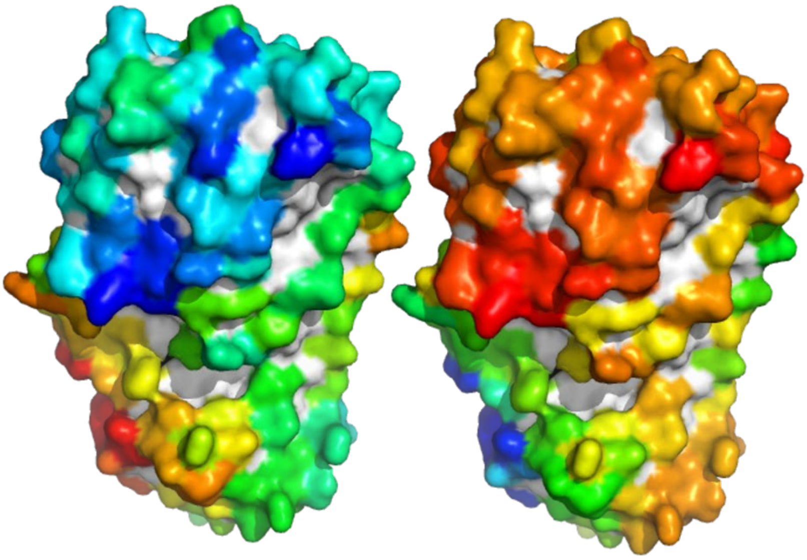


Figure 45 The hydrophobicity (left) and availability (right) of residues in the 5RE7 PDB file. The colours are on a spectrum such that red is high and blue/violet are low.

11.4 Hepatitis *B* Dataset

11.4.1 Pangenome annotation

The Hepatitis B analysis revealed 14 conserved gene families. Of these 14 genes, 2 were deemed "soft core" (> 95% coverage but < 99%) genes with others falling into the shell category. The summary of the presence of each gene in the pangenome can be found in Figure 46. The gene annotation for every gene was marked as hypothetical in Roary. To overcome this, a BLASTx search was performed using the pan genome reference sequence for each of the 14 gene families.

The pangenome calculation appears to have missed gene annotations in certain genomes. As discussed earlier, Prokka was originally designed with bacteria in mind. Due to this, all groups were considered in order to avoid removing potentially interesting genes.

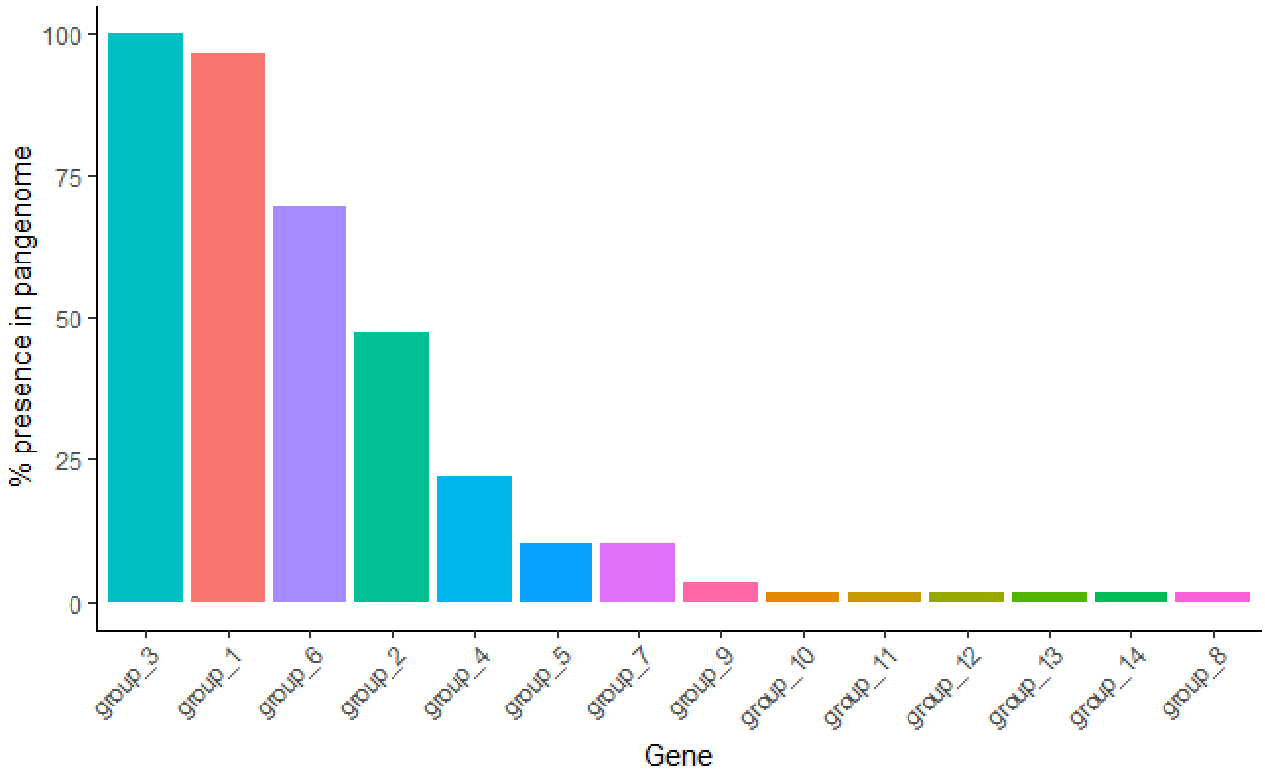


Figure 46 Percentage presence in the pangenome for each gene group identified by Roary.

Like the example in 11.3, all the results returned were not annotated as genes, but rather as ORFs. Through manual searching, it was found that Group_3 and group_5 are both part of the polymerase of the Hepatitis B genome, which has already been targeted in anti-viral contexts[323] but not by the immune system. Group_1 appears to be HBcAg, the Hepatitis B core antigen which is a well-described target for vaccines [324,325]. HBcAg is the capsid protein of the HBV virus. Group_2 was annotated as the X protein.

11.4.2 Tajima's D and immune prediction

A total of 7 of the 14 pangenome groups were undergoing strong negative selection. Immunogenicity predictions for the proteins showed that there was a stronger emphasis on BCR epitopes being the mode of recognition for these proteins. The number of MHC binders (both class I and II) were low for all groups. Group_3, the polymerase gene was the best predicted MHC class I and II target with 11 and 7 predicted epitopes, respectively.

Table 15 the 7 strongly selected Hepatitis B gene families, with their Tajima's D score, the number of BCR epitopes and the total number of predicted good MHC I and II binders. The number of BCR epitopes here is the total number of peptides longer than 5 amino acids with predicted B-cell receptor binding.

Gene	D	BCR epitopes	MHC I good binders	MHC II good binders
group_6	14.1548	1	0	0
group_1	13.53904	5	3	0
group_5	10.80408	5	2	0
group_3	10.46658	8	11	7
group_2	10.32454	2	0	0
group_7	9.398233	2	3	0

Groups 1, 3 and 5 had the highest amount of BCR epitopes. Group 1 had 5 individual epitopes comprising of a total number of 99 amino acids, group 3 had 8 epitopes totalling 173 amino acids and finally group 5 had 5 epitopes consisting of 72 amino acids. The conservation of each peptide was calculated across the pangenome is described in Table 16. Table 16 shows that group 1 had 4 peptides that were well-conserved (here deemed to be any percentage > 80%), group 3 had 1 peptide and group 5 had 3.

Group 1 peptide IDPYKEFGATVE was seen to be a substring of known HBV epitopes based on a BLAST search with >90% similarity in the IEDB, however this was only to a T-cell epitope, not the predicted BCR epitope. However regions of peptide NTNMGLKFRQLLWF had previously been described as an epitope[326]; as did TPPAYRPPNAPILSTLPETTVVRRRGRSPRRRTPSPRRRRSQSPRRRRSQSR[327]. Neither peptide showed a similarity to peptides in other species.

Table 16 Each peptide predicted to be a good BCR epitope, the gene corresponding to it and the percentage at which it is present in the pangenomic sequences.

gene	peptide	% conservation
group_1	DIDPYKEFGATVE	87.7193
group_1	RDALESPEHCSPH	3.508772
group_1	VNLEDPAS	14.03509
group_1	NTNMGLKFRQLLWF	82.45614
group_1	TPPAYRPPNAPILSTLPETTVVRRRGRSPRRRTPSPRRRRSQSPRRRRSQSR	80.70175
group_3	KFAVPNLQSLTNLLSSNLSWLSLDVSAAFYHLPLHPAAMPHLLVGSS	88.13559
group_3	SNSRILNHQHGTMQN	3.389831
group_3	EHIIQK	5.084746
group_3	KECFRKLPVNRPIDWKV	93.22034
group_3	QAFTFSPT	94.91525
group_3	LNLYPVARQRP	52.54237
group_3	RMRGTFLAPLP	62.71186

group_3	VYVPSALNPADDPSRGRLGLSRPLLRLPFRPTTGRTSLYADSPSVPSHLPDRVHFASP	67.79661
group_5	DCEFWPRHTVVPPRKLREVHHESR	100
group_5	NKKNPACNTRRGPRNPD	16.66667
group_5	LEKIDDKGEAV	16.66667
group_5	FTVPELEPPAGKYRPLTLGS	66.66667

11.4.3 Database searching

None of the samples in Table 15 had any association for genes annotated for essentiality and virulence in the DEG and VFDB. However, group_1 did have a high similarity (99.5%) match with the PDB *6HTX.* Unlike in the SARS-CoV-2 example, there was very few pockets of highly available residues, with only one residue at position 146 being highly available. This can be seen in the red at the top of Figure 47. This did not link to any predicted B cell epitopes or predicted MHC presented peptides. The predicted epitope site DIDPYKEFGATVE was found starting at residue 2 in the 6HTX PDB file and did not have a high availability.

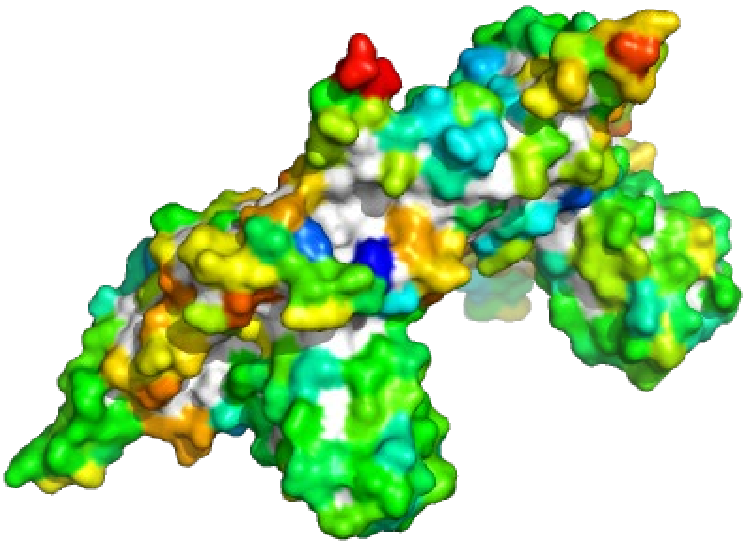


Figure 47 Accessibility of the potential HBV vaccine target predicted by RECIPIENT. Red indicates high availability.

11.5 *Performance*

Each of the four datasets analysed were done so using HPC resources at Advanced Research Computing Cardiff University (ARCCA). MASH and Roary were run on 40 cores as they required a large input: every source FASTA file for MASH and every annotated GFF file for Roary. The rest of the tasks discussed were ran on one core. The run times in CPU hours are shown in Figure 48.

As shown below in Figure 48, performance scales with the size of the pangenome dataset. There is an initial bottleneck in processing and annotating the genome in Prokka and then the creation of the pangenome in Roary; as well as the optional steps of k-mer sketching and k-mer classification using Mash and Kraken2 respectively. These processes scale with the number of input FASTA files.

The majority of the steps outside of that are scaled on the size of the genome as from there on in each process is acted upon an individual gene using the pangenome reference. Processes limited in this way include the MHC class I and II binding, BCR epitope prediction, the subcellular localisation prediction and database searches. Tajima’s D, sequence entropy and calculating the conservation of peptides is dependent on both the number of genomes and the size of the genome in terms of genes found.

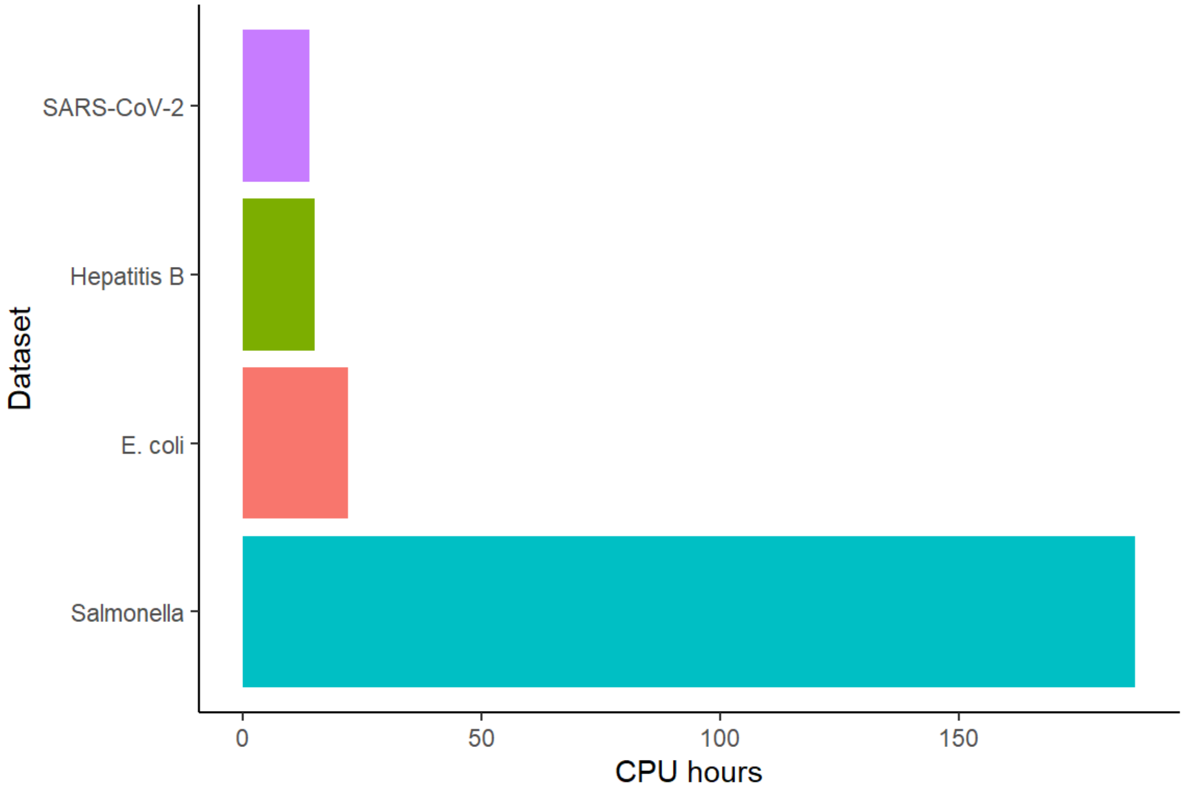


Figure 48 runtime in CPU hours for each of the 4 datasets. Performance roughly scales to the size of the pangenome analysed.

12 DISCUSSION

12.1 IMPLICATION TO VACCINE DESIGN

Vaccine design is a constantly evolving field, requiring vast amounts of technological and computational resources as well as biological understanding on the target being vaccinated against, the immune response of the host and molecular mechanisms underpinning this response. This response is of course complicated by the fact that the target of the vaccine (e.g., a virus, bacteria, or cancer) are in a dynamic interplay with the host response. This interplay happens on both the individual and population level.

12.1.1 RECIPIENT and reverse vaccinology

On the whole protein or gene level RV has truly become a viable option for vaccine design efforts with the explosion of NGS technologies. In recent years, the cost has dropped significantly and continues to do so (Figure 49).

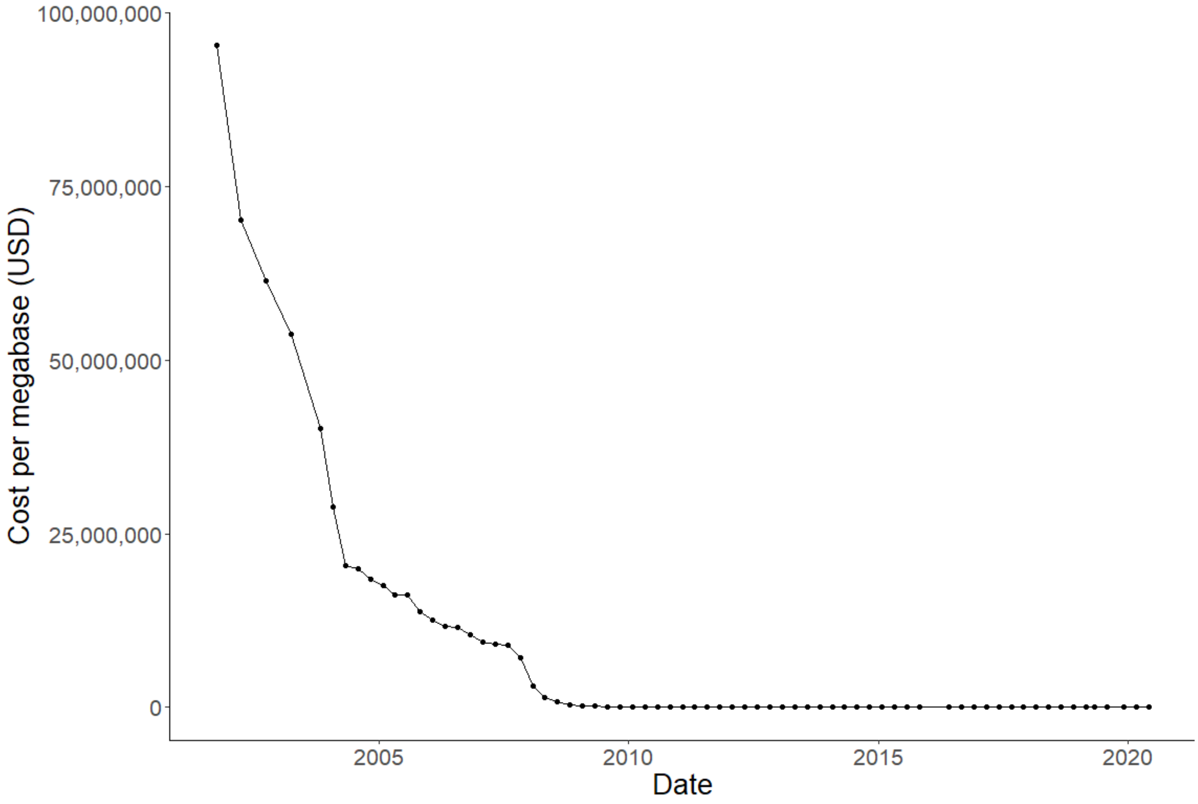


Figure 49 The cost to sequence a megabase of DNA, adapted from US government statistics (https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data)

The abundance of raw data is complemented by a growing stable of different methods for understanding pangenomes[252,328,329] and population dynamics of pathogens. Specific to computational immunology, there is also a growing pool of predictive tools available for estimating immune response, be it TCR binding, MHC I and II binding, proteasomal cleavage and processing and linear or non-linear B-cell epitope prediction. This is supplemented by databases understanding the effectiveness, essentiality, and virulence of genes in pathogens[261,262].

All of these technological advances mean that RV provides a genuine methodology to predict vaccine targets. RECIPIENT contributes to this field by adding several novel features not previously considered in the literature, including estimation of selection pressure by means of Tajima's D and Shannon entropy; structural homology scanning and considering the pangenome. Furthermore, this novelty is supported by being designed in a framework that prioritises ease of use through pipeline management and containerization.

Pipeline management and containerisation workflows (discussed in 10.3) are an incredibly important facet of bioinformatics project. It allows for the bundling of software packages into single purpose environments which are easily shared. It improves tool versioning, documentation and the ability in installing software[330]; as well as improving the overall reproducibility of results[331].

The biological interpretation of the test data used in RECIPIENT has also garnered some interesting vaccine targets, as well as revealing avenues for further improvements to the tool.

In the prediction of potential *Salmonella* Typhi vaccination targets a number of targets were predicted that could be of interest, however further investigation is needed, either because that gene is not well functionally phenotyped or has not been described in *Salmonella* Typhi (or similar model systems, such as *Salmonella* Typhimurium).

The *Salmonella* Typhi pangenome showed a strong amount of polarisation, with a large number of genes either being in the core genome (45.2%) or the cloud genome (47.2%) meaning that a large number of genes were instantly discarded. After filtering genes on the following conditions:

1. Presence in the pangenome > 90%
2. Tajima's D > 2
3. Not being annotated as a hypothetical gene/gene product
4. Subcellular localisation predicted to be "extracellular", "outer membrane", "flagellar" "T3SS", "host associated" or "fimbriael"

this left 15 potential gene candidates, 12 of which had at least one MHC class II predicted epitopes considered a "good binder" by netMHCIIpan. In these 12 genes, every peptide was conserved in >80% of samples and 67.9% of these peptides were present in >99% of samples, showing that the RECIPIENT method found a number of potentially immunogenic, conserved peptides.

The efforts to characterise the genes for which these peptides were predicted demonstrates one important caveat of RV approaches, a number of these genes were not well-characterised in *Salmonella* Typhi. This is a two-edged sword, one the one hand it means that this predicted target is novel and could be an interesting vaccine target. However, on the other hand it means that without experimental validation these targets remain predictions.

Some of these predicted peptides warrant further investigation. bamD, caf1A, kdgM, lptE, PldA all had little information about their role in immune response to *Salmonella*. BamD is well characterised in *Salmonella*[332], but the remaining genes were not.

FimD is well-described in *Salmonella*[301]. Interestingly, other RV methods have also predicted fimD as a possible target in *Acinetobacter baumannii*[302] and *Acinetobacter nosocomialis*[303] but there are no functional studies. OmpF has been shown to be immunogenic in *Salmonella*[309] and is also well-described indicating that both could be investigated further.

The results on the EPEC pangenome were also interesting. Firstly, the fact that many known genes were not annotated successfully in Prokka. Especially in other less understood systems, this could lead to potential genes of interest being missed. One way around this would be allowing the pipeline to accept fastq reads of the data, along with a reference and annotation (e.g., a GenBank file or GTF file) and map directly to this. This would limit RECIPIENT to systems that are well described and have a reference genome of sufficient quality which could hinder usability.

This reference-based method however could lead to enhanced results. This would require further testing as Roary is designed to be ran on the output of Prokka, however in principle the output should be the same. Alternatively, a tool designed to be ran on with a reference genome could be used, such as NGSPanPipe[333].

A well-annotated reference genome, or an option to supply annotation (for example in through a genbank file) would also aid in the annotation of plasmids, which was shown to be an issue in 11.1.1. This option would be beneficial for well-described genomes. If a sample does not have a sufficiently annotated genome, then the pipeline is dependent on being supplied assemblies that were plasmid aware. I believe that performing genome assembly as part of the pipeline is too complex to do without prior knowledge of the sample.

7 genes of interest were shown to have a good immune predicted immune response, either by MHC I or II mediated presentation, or by containing linear B-cell epitopes. They were all well conserved (present in >80% of samples). However, none have been seen in the IEDB. It remains unclear whether this is due to the inaccuracy of the epitope prediction software or if they have not been validated yet.

The HBV and SARS-CoV-2 datasets both demonstrated difficulty in finding good vaccination targets for the two viruses of interest. Partially this could be argued that the RV approach works well for bacteria due to having a higher number of genes. As will be discussed in 12.2, there are several possible avenues to be explored that would increase the effectiveness of RECIPIENT in analysing viral targets.

I believe that the SARS-CoV-2 dataset in particular demonstrates the limitations of RECIPIENT. Thanks to a large-scale global effort the number of genomes available of SARS-CoV-2 is unparalleled (>2 million genomes in the United Kingdom alone) and there is a well-described pipeline for aligning, annotating and tracking mutations of these genomes (for example Datapipe and Phylopipe in the United Kingdom[334]) whose computational resources outstretch those of a single user running RECIPIENT. The pangenome generation is very computationally expensive, scaling with the size of the genome being studied and the number of genomes; meaning that constructing a pangenome of the vast number of sequences of SARS-CoV-2 is not feasible.

Moreso, given the small and well-conserved nature of the SARS-CoV-2 genome, the pangenome approach may be redundant as one could simply extract all of the gene sequences made publically available through these sequencing efforts and proceed with downstream analysis directly.

12.1.2 GPU-accelerated CPL scanning

CPL scans have an important role in vaccine design. The most obvious being the identification of high quality potential epitopes through database driven scans[335]. However there is more depth to the role of CPL scans, the naturally wide-searching technique means that it can be also used to estimate degeneracy of a T-cell[26] and how a high affinity TCR might interact with the self proteome[336]. All methods can be improved in terms of computational speed by GP-GPU, as shown in chapter 8.

GP-GPU is going from strength to strength. As of November 2019, 6 of the top 10 supercomputers in the world based on the TOP500 (https://www.top500.org) rating have a significant influence from GPU acceleration[337] and while this was traditionally the domain of chemistry, physics and machine learning[338] there is an ever-expanding pool of software in genomics, structural biology, biochemistry and systems biology that implements GP-GPU[131].

Here I have provided a foundation for applying GP-GPU to CPL library scanning, and more broadly have introduced another GP-GPU method for analyzing biological sequence data. This is important for two reasons. The first is that it accelerated the CPL scanning software significantly. The increase in performance means that larger databases or more samples can be scanned without a worry of computational limitations. The second is that the CUDA implementation of PICPL has shown more broadly that CPLs and protein sequence datasets are conducive to GP-GPU programming. This opens up further avenues for the application of GP-GPU to similarly framed problems.

12.1.3 STACEI and structural profiling of the TCR-pMHC complex

Structural understanding of the TCR-pMHC complex is essential for truly understanding immune recognition and in turn designing vaccines. These efforts could be in generalized frameworks, e.g. using the binding “rules” generated by all known structures to inform decisions on the repertoire level as in GLIPH[76]; or they could be more direct methods to gain the exact mechanism of a certain vaccine, for example demonstrating that modification of the anchor residues of the peptide confer recognition[67,339] and the role of hotspots or conserved binding sites in recognition of antigen[257]. There are also methods for designing and inferring structure from sequence[75,340] to aid these steps. However, none of these various efforts provide a systematic way analyze the binding modes of TCR-pMHC complexes, unlike STACEI. This means that the work in chapter 3-5 complements more "black box" methods for vaccine design that use machine learning techniques, for example in defining the how therapeutic an antibody is[341] and using modelling approaches to infer the function of a TCR indirectly[75,340].

At the time of writing there are still no methods for calculating the number of contacts for a given TCR-pMHC structure, quantify the type of bonds made by these structures and present them in a human readable fashion. Likewise, no methods measure BSA, ASA in a higher resolution then on a chain-by-chain basis. Also, no other tool generates publication quality tools to the same degree as STACEI. Measures that are replicated, such as IMGT numbering (found in STCRDAB: http://opig.stats.ox.ac.uk/webapps/stcrdab/) , SC and crossing angle (TCR3D: https://tcr3d.ibbr.umd.edu/) are not available to be ran locally or on a server with a PDB file of the user's choice, meaning that analysis is limited to already publicly available TCR-pMHC complexes.

Similar to RECIPIENT, it is version controlled, it exists as part of a Python package and is able to be ran in a container, meaning that the tool should be able to be executed in a reproducible manner well into the future.

The findings of the review of αβ TCR-pMHC complexes found in the PDB also shed some biological interpretation of the overall binding mechanisms of these complexes, the overall number of contacts between TCR and peptide is relatively well conserved, whilst MHC class II restricted peptides have a larger frequency of contacts to its constituent MHC, due to the flatter binding of the peptide with each end flanking out the ends of the MHC.

The relationship between SC and number of contacts made shows a weak correlation. This demonstrates the inherent need for both measures. SC provides a good overview of "overall" binding and interaction between TCR and pMHC; but does not portray the whole story. SC does not necessarily constitute good overall affinity, whereas number of contacts made by the TCR to the peptide does appear to be moreso[342].

Calculation of BSA, ASA and availability in STACEI was successful in being able to indicate which peptide residues were buried into the MHC binding groove, constituting anchor residues and which were able to be accessed by the TCR. This is important in the rational design of peptide vaccine targets as one would want to preserve the stability of the anchor residues whilst making the remaining residues able to be engaged sufficiently by the TCR.

Finally, this overall picture of broad TCR-pMHC binding could become more important as computationally aided rational design of TCRs becomes more prevalent. Similar to Thera-SAbDab[343] in the antibody design space the collective output of STACEI could be used to define "standard" behaviour of a TCR-pMHC complex in order to help predict what designs are biologically feasible. However, this is some way off being feasible in terms of TCR-pMHCs: There are > 3,600 antibody structures in the PDB compared to 520 TCRs at the time of writing.

12.2 Future work

In the future, I would like to expand these tools. STACEI currently exists as a tool specifically for analyzing αβ TCRs in complex with pMHC. The decision to specifically target αβ TCR-pMHCs was made early on in the development of the tool, meaning that a number of analyses and error checking steps explicitly make assumptions specific to αβ complexes (e.g. that the antigen is a peptide).

As an ongoing project, collaborators and I are making STACEI more generalized and integrating options to analyze "free" structures of TCRs not in complex with pMHC and for TCRs engaging with non-peptide antigens and non-classical MHC-like molecules, such as CD1 and MHC-related protein 1 (MR1).

Some of the changes needed to be made are relatively simple, for example generalizing STACEI to detect and pair γδ TCRs and detect non-classical MHC only requires expansion of the BLAST database used in chain pairing. However, detecting non-peptide ligands effectively with no *a priori* knowledge remains a challenge. STACEI assumes that all molecules involved in the TCR recognition are independent chains in the PDB file and this is often not the case for non-peptide antigens. This means that a search of the *HETATOM* (discussed briefly in 3.1.3.1) section of the PDB file. While finding what atoms contact the

TCR in the *HETATOM* section is trivial, deciding which is the antigen is not, as there are often a number of cofactors, solvents and ions contacting the TCR, too.

In chapter 8 I demonstrated that GP-GPU has a role to play in the efficient scanning of CPL databases. In this chapter I briefly touched on assessing the similarity of peptide sets by means of CUDA sped-up alignment, I would like to explore this further, and see if PAM30 alignment matrices can infer peptides which are cross-reactive mimics of self-peptides. In order to do this, I would collate known self-peptides and run them against the human genome.

The CUDA driven database scan is also currently ran on one system using one GPU. In future the software could be expanded to leverage CUDA's support for multiple GPUs by using the *cudaSetDevice()* and *cudaDeviceSynchronize()* functionalties. This should generalize well, certainly if still only using a low number of GPUs as each step can still be performed independently of the other steps until the final sort is performed. At the time of writing this is performed using a bash script that generates multiple instances of the same program and then collates the results. While this does work, it is not optimal.

RECIPIENT, described in chapters 9, 10 and 11 provides a framework for the design of potential vaccine candidates. At the moment, a number of the key steps, such as annotation and subcellular localization prediction only work for bacteria and fungi (although not explored in this thesis). In future, I would first like to address these steps for viral genomes. While gene annotation is partially dependent on the literature surrounding a given pathogen, there are tools aimed at virus specific annotation such as VAPiD[318] and Vgas[317].

Failing this, I would like to consider an input option that allows users to upload a DIAMOND search database of genes of their interest to perform bespoke annotation. An alternative option would be to allow the pipeline to be ran from FASTQ reads and mapping the genome to a reference and annotation file. As discussed above, this would somewhat hinder RECIPIENT in only being able to be ran on well-understood systems, it would give more accurate results.

The second point I would like to address is subcellular localization of viral proteins. This is of course, a slightly different framing to most bacterial proteins, and aims to predict instead where in the host cell these proteins aggregate. Most tools (e.g. PSORT and LocTree, discussed earlier) are aimed at prokaryotic and eukaryotic localization prediction. However, tools such as MSLVP[344] are geared towards predicting where in human cells the virus localizes. I would like to integrate this into the pipeline to address the problem of localization.

I also would like to explore a more predictive method in assessing these vaccine targets. To do this I would need to find known vaccine targets and run them through RECIPEINT. An approach similar to the *Therapeutic Antibody Profiler*[341] could be applied wherein new targets are compared to how similar their properties are to known vaccine targets.

12.3 Combining the tools

There is a great potential to develop these immunoinformatics tools further by utilising their strengths in epitope discovery and vaccine design. In particular, the role of structural analysis could be particularly fruitful, as it has such a fundamental focus on the bottom-up mechanism of TCR-pMHC binding and recognition. For example, the structural analysis efforts of STACEI could be applied to the GPU accelerated epitope design. The top scoring peptides could be modelled using existing tools[69,75,340] to create template TCR-pMHC structures to address the difficulty in experimental validation. This could help expediate analysis as structure determination is often a difficult bottleneck in the analysis process. STACEI could then be used to analyze these structures to find shared commonalities of the binding event, as well as find potential areas for improvement not captured by the GPU scanning. Likewise, this could also be done in the case of RECIPIENT, incorporating TCR modelling and structural analysis to the top epitopes found, as well as modelling the whole proteins for antibody targets. Although computationally very expensive, the existing framework of RECIPIENT would allow users to filter out incompatible targets.

Another interesting facet for analysis would be the role of TCR cross-reactivity in vaccine design. The degeneracy of the TCR is both a positive and a negative in vaccine design as it allows for a potentially pan-pathogen response[345], but also can cause targeting of self-antigens[346]. One must consider the potential off-target effects of TCR based vaccines. STACEI allows for a mechanistic insight into the underpinning of cross-reactivity. There are multiple instances of the same TCR being engaged with different pMHC. The mechanism of binding could be combined with estimates of diversity from CPL scans to predict whether more cross-reactive TCRs behave so through a conserved mechanism. This could then steer vaccine design away from overly cross-reactive T-cells.

The GPU accelerated CPL scanning could also compliment the RECIPIENT pipeline. In another project not discussed in this thesis, a “mock” CPL scan file was generated based on the known antigen of a T-cell clone and was used to predict potential viral mimicry in the clearance of melanoma[347]. This same approach could be used to help predict vaccine targets in the RECIPIENT pipeline by allowing users to direct their search using CPL scans of T-cells of interest.

13 BIBLIOGRAPHY

1. Turvey, S. E. & Broide, D. H. Innate immunity. *J. Allergy Clin. Immunol.* **125**, S24–S32 (2010).

2. Anthony, R. M., Rutitzky, L. I., Urban Jr, J. F., Stadecker, M. J. & Gause, W. C. Protective immune mechanisms in helminth infection. *Nat. Rev. Immunol.* **7**, 975–987 (2007).

3. Boots, M. & Bowers, R. G. Three mechanisms of host resistance to microparasites-avoidance, recovery and tolerance-show different evolutionary dynamics. *J. Theor. Biol.* **201**, 13–23 (1999).

4. Ladner, R. C. Mapping the Epitopes of Antibodies. *Biotechnol. Genet. Eng. Rev.* **24**, 1–30 (2007).

5. Doherty, D. G., Melo, A. M., Moreno-Olivera, A. & Solomos, A. C. Activation and Regulation of B Cell Responses by Invariant Natural Killer T Cells . *Frontiers in Immunology* vol. 9 1360 (2018).

6. Townsend, C. L. *et al.* Significant Differences in Physicochemical Properties of Human Immunoglobulin Kappa and Lambda CDR3 Regions. *Front. Immunol.* **7**, 388 (2016).

7. Green, A. E. *et al.* Recognition of nonpeptide antigens by human V gamma 9V delta 2 T cells requires contact with cells of human origin. *Clin. Exp. Immunol.* **136**, 472–482 (2004).

8. Karunakaran, M. M., Göbel, T. W., Starick, L., Walter, L. & Herrmann, T. Vγ9 and Vδ2 T cell antigen receptor genes and butyrophilin 3 (BTN3) emerged with placental mammals and are concomitantly preserved in selected species like alpaca (Vicugna pacos). *Immunogenetics* **66**, 243–254 (2014).

9. Halary, F. *et al.* Shared reactivity of V{delta}2(neg) {gamma}{delta} T cells against cytomegalovirus-infected cells and tumor intestinal epithelial cells. *J. Exp. Med.* **201**, 1567–1578 (2005).

10. Neefjes, J., Jongsma, M. L. M., Paul, P. & Bakke, O. Towards a systems understanding of MHC class I and MHC class II antigen presentation. *Nat. Rev. Immunol.* **11**, 823 (2011).

11. Rudolph, M. G., Stanfield, R. L. & Wilson, I. A. How Tcrs Bind Mhcs, Peptides, and Coreceptors. *Annu. Rev. Immunol.* **24**, 419–466 (2006).

12. Lythe, G., Callard, R. E., Hoare, R. L. & Molina-París, C. How many TCR clonotypes does a body maintain? *J. Theor. Biol.* **389**, 214–224 (2016).

13. Davis, M. M. *et al.* LIGAND RECOGNITION BY αβ T CELL RECEPTORS. *Annu. Rev. Immunol.* **16**, 523–544 (1998).

14. Robinson, J. *et al.* The IPD and IMGT/HLA database: allele variant databases. *Nucleic Acids Res.* **43**, D423–D431 (2015).

15. McAdam, S. N. *et al.* A uniquely high level of recombination at the HLA-B locus. *Proc. Natl. Acad. Sci. U. S. A.* **91**, 5893–5897 (1994).

16. Chothia, C., Boswell, D. R. & Lesk, A. M. The outline structure of the T-cell alpha beta receptor. *EMBO J.* **7**, 3745–3755 (1988).

17. Ma, Y., Pannicke, U., Schwarz, K. & Lieber, M. R. Hairpin Opening and Overhang Processing by an Artemis/DNA-Dependent Protein Kinase Complex in Nonhomologous End Joining and V(D)J Recombination. *Cell* **108**, 781–794 (2002).

18. Hülsmeyer, M. *et al.* A Major Histocompatibility Complex·Peptide-restricted Antibody and T Cell Receptor Molecules Recognize Their Target by Distinct Binding Modes: CRYSTAL STRUCTURE OF HUMAN LEUKOCYTE ANTIGEN (HLA)-A1·MAGE-A1 IN COMPLEX WITH FAB-HYB3 . *J. Biol. Chem.* **280**, 2972–2980 (2005).

19. Rossjohn, J. *et al.* T cell antigen receptor recognition of antigen-presenting molecules. *Annu. Rev. Immunol.* **33**, 169–200 (2015).

20. Tynan, F. E. *et al.* A T cell receptor flattens a bulged antigenic peptide presented by a major histocompatibility complex class I molecule. *Nat. Immunol.* **8**, 268–276 (2007).

21. Borbulevych, O. Y. *et al.* T cell receptor cross-reactivity directed by antigen-dependent tuning of peptide-MHC molecular flexibility. *Immunity* **31**, 885–896

(2009).

22. Burrows, S. R. *et al.* Hard wiring of T cell receptor specificity for the major histocompatibility complex is underpinned by TCR adaptability. *Proc. Natl. Acad. Sci. U. S. A.* **107**, 10608–10613 (2010).

23. Mason, D. A very high level of crossreactivity is an essential feature of the T-cell receptor. *Immunol. Today* **19**, 395–404 (1998).

24. Borbulevych, O. Y., Santhanagopolan, S. M., Hossain, M. & Baker, B. M. TCRs used in cancer gene therapy cross-react with MART-1/Melan-A tumor antigens via distinct mechanisms. *J. Immunol.* **187**, 2453–2463 (2011).

25. Birnbaum, M. E. *et al.* Deconstructing the peptide-MHC specificity of T cell recognition. *Cell* **157**, 1073–1087 (2014).

26. Wooldridge, L. *et al.* A Single Autoimmune T Cell Receptor Recognizes More Than a Million Different Peptides * □. *J. Biol. Chem.* **287**, 1168–1177 (2012).

27. Kreslavsky, T., Gleimer, M., Garbe, A. I. & von Boehmer, H. αβ versus γδ fate choice: counting the T-cell lineages at the branch point. *Immunol. Rev.* **238**, 169–181 (2010).

28. Hassan, C. *et al.* Naturally processed non-canonical HLA-A*02:01 presented peptides. *J. Biol. Chem.* **290**, 2593–2603 (2015).

29. Holland, C. J., Cole, D. K. & Godkin, A. Re-Directing CD4(+) T Cell Responses with the Flanking Residues of MHC Class II-Bound Peptides: The Core is Not Enough. *Front. Immunol.* **4**, 172 (2013).

30. Li, Q.-J. *et al.* CD4 enhances T cell sensitivity to antigen by coordinating Lck accumulation at the immunological synapse. *Nat. Immunol.* **5**, 791–799 (2004).

31. Kern, P., Hussey, R. E., Spoerl, R., Reinherz, E. L. & Chang, H. C. Expression, purification, and functional analysis of murine ectodomain fragments of CD8alphaalpha and CD8alphabeta dimers. *J. Biol. Chem.* **274**, 27237–27243 (1999).

32. Gangadharan, D. & Cheroutre, H. The CD8 isoform CD8αα is not a functional homologue of the TCR co-receptor CD8αβ. *Curr. Opin. Immunol.* **16**, 264–270 (2004).

33. Jones, E. Y., Fugger, L., Strominger, J. L. & Siebold, C. MHC class II proteins and disease: a structural perspective. *Nat. Rev. Immunol.* **6**, 271–282 (2006).

34. Harndahl, M. *et al.* Peptide-MHC class I stability is a better predictor than peptide affinity of CTL immunogenicity. *Eur. J. Immunol.* **42**, 1405–1416 (2012).

35. Vigneron, N. & Van den Eynde, B. J. Proteasome subtypes and regulators in the processing of antigenic peptides presented by class I molecules of the major histocompatibility complex. *Biomolecules* **4**, 994–1025 (2014).

36. York, I. A., Brehm, M. A., Zendzian, S., Towne, C. F. & Rock, K. L. Endoplasmic reticulum aminopeptidase 1 (ERAP1) trims MHC class I-presented peptides in vivo and plays an important role in immunodominance. *Proc. Natl. Acad. Sci.* **103**, 9202 LP – 9207 (2006).

37. Harding, C. V & Unanue, E. R. Quantitation of antigen-presenting cell MHC class II/peptide complexes necessary for T-cell stimulation. *Nature* **346**, 574–576 (1990).

38. Mantegazza, A. R., Magalhaes, J. G., Amigorena, S. & Marks, M. S. Presentation of phagocytosed antigens by MHC class I and II. *Traffic* **14**, 135–152 (2013).

39. Neefjes, J. CIIV, MIIC and other compartments for MHC class II loading. *Eur. J. Immunol.* **29**, 1421–1425 (1999).

40. Colbert, J. D., Cruz, F. M. & Rock, K. L. Cross-presentation of exogenous antigens on MHC I molecules. *Curr. Opin. Immunol.* **64**, 1–8 (2020).

41. Sigal, L. J., Crotty, S., Andino, R. & Rock, K. L. Cytotoxic T-cell immunity to virus-infected non-haematopoietic cells requires presentation of exogenous antigen. *Nature* **398**, 77–80 (1999).

42. Huang, A. Y. C. *et al.* Role of bone marrow-derived cells in presenting MHC class I-restricted tumor antigens. *Science (80-.).* **264**, 961–965 (1994).

43. Smith, C. M. *et al.* Cognate CD4(+) T cell licensing of dendritic cells in CD8(+) T cell immunity. *Nat. Immunol.* **5**, 1143–1148 (2004).

44. Muraille, E., Leo, O. & Moser, M. Th1/Th2 Paradigm Extended: Macrophage Polarization as an Unappreciated Pathogen-Driven Escape Mechanism? . *Frontiers in*

Immunology vol. 5 603 (2014).

45. Strutt, T. M. *et al.* Memory CD4+ T cells induce innate responses independently of pathogen. *Nat. Med.* **16**, 558–64, 1p following 564 (2010).

46. Walker, J. A. & McKenzie, A. N. J. TH2 cell development and function. *Nat. Rev. Immunol.* **18**, 121–133 (2018).

47. Tesmer, L. A., Lundy, S. K., Sarkar, S. & Fox, D. A. Th17 cells in human disease. *Immunol. Rev.* **223**, 87–113 (2008).

48. Crotty, S. T follicular helper cell differentiation, function, and roles in disease. *Immunity* **41**, 529–542 (2014).

49. Leem, J., de Oliveira, S. H. P., Krawczyk, K. & Deane, C. M. STCRDab: the structural T-cell receptor database. *Nucleic Acids Res.* **46**, D406–D412 (2018).

50. Shugay, M. *et al.* VDJdb: a curated database of T-cell receptor sequences with known antigen specificity. *Nucleic Acids Res.* **46**, D419–D427 (2018).

51. Rosati, E. *et al.* Overview of methodologies for T-cell receptor repertoire analysis. *BMC Biotechnol.* **17**, 61 (2017).

52. Padovan, E. *et al.* Expression of two T cell receptor alpha chains: dual receptor T cells. *Science (80-.).* **262**, 422 LP – 424 (1993).

53. Brown, S. D., Raeburn, L. A. & Holt, R. A. Profiling tissue-resident T cell repertoires by RNA sequencing. *Genome Med.* **7**, 125 (2015).

54. Giudicelli, V., Chaume, D. & Lefranc, M.-P. IMGT/GENE-DB: a comprehensive database for human and mouse immunoglobulin and T cell receptor genes. *Nucleic Acids Res.* **33**, D256–D261 (2005).

55. Altschul, S. F., Gish, W., Miller, W., Myers, E. W. & Lipman, D. J. Basic local alignment search tool. *J. Mol. Biol.* **215**, 403–410 (1990).

56. Ye, J., Ma, N., Madden, T. L. & Ostell, J. M. IgBLAST: an immunoglobulin variable domain sequence analysis tool. *Nucleic Acids Res.* **41**, W34–W40 (2013).

57. Zhang, W. *et al.* IMonitor: A Robust Pipeline for TCR and BCR Repertoire Analysis.

Genetics **201**, 459–472 (2015).

58. Langmead, B. & Salzberg, S. L. Fast gapped-read alignment with Bowtie 2. *Nat. Methods* **9**, 357–359 (2012).

59. Gerritsen, B., Pandit, A., Andeweg, A. C. & de Boer, R. J. RTCR: a pipeline for complete and accurate recovery of T cell repertoires from high throughput sequencing data. *Bioinformatics* **32**, 3098–3106 (2016).

60. Thomas, N., Heather, J., Ndifon, W., Shawe-Taylor, J. & Chain, B. Decombinator: a tool for fast, efficient gene assignment in T-cell receptor sequences using a finite state machine. *Bioinformatics* **29**, 542–550 (2013).

61. Bolotin, D. A. *et al.* MiTCR: software for T-cell receptor sequencing data analysis. *Nat. Methods* **10**, 813–814 (2013).

62. Bolotin, D. A. *et al.* MiXCR: software for comprehensive adaptive immunity profiling. *Nat. Methods* **12**, 380–381 (2015).

63. Morris, E. K. *et al.* Choosing and using diversity indices: insights for ecological applications from the German Biodiversity Exploratories. *Ecol. Evol.* **4**, 3514–3524 (2014).

64. Sethna, Z., Elhanati, Y., Callan Jr, C. G., Walczak, A. M. & Mora, T. OLGA: fast computation of generation probabilities of B- and T-cell receptor amino acid sequences and motifs. *Bioinformatics* **35**, 2974–2981 (2019).

65. Bagaev, D. V *et al.* VDJviz: a versatile browser for immunogenomics data. *BMC Genomics* **17**, 453 (2016).

66. Cole, D. K. *et al.* Dual Molecular Mechanisms Govern Escape at Immunodominant HLA A2-Restricted HIV Epitope . *Frontiers in Immunology* vol. 8 1503 (2017).

67. Madura, F. *et al.* TCR-induced alteration of primary MHC peptide anchor residue. *Eur. J. Immunol.* **49**, 1052–1066 (2019).

68. Cole, D. K. *et al.* Hotspot autoimmune T cell receptor binding underlies pathogen and insulin peptide cross-reactivity. *J. Clin. Invest.* **126**, 2191–2204 (2016).

69. Chaudhury, S., Lyskov, S. & Gray, J. J. PyRosetta: a script-based interface for implementing molecular modeling algorithms using Rosetta. *Bioinformatics* **26**, 689–691 (2010).

70. Haidar, J. N. *et al.* Structure-based design of a T-cell receptor leads to nearly 100-fold improvement in binding affinity for pepMHC. *Proteins* **74**, 948–960 (2009).

71. Nussinov, B. M. and R. Trp/Met/Phe Hot Spots in Protein-Protein Interactions: Potential Targets in Drug Design. *Current Topics in Medicinal Chemistry* vol. 7 999–1005 (2007).

72. Zoete, V., Irving, M., Ferber, M., Cuendet, M. A. & Michielin, O. Structure-Based, Rational Design of T Cell Receptors. *Front. Immunol.* **4**, 268 (2013).

73. Sivasubramanian, A., Sircar, A., Chaudhury, S. & Gray, J. J. Toward high-resolution homology modeling of antibody Fv regions and application to antibody-antigen docking. *Proteins* **74**, 497–514 (2009).

74. Choi, Y. & Deane, C. M. FREAD revisited: Accurate loop structure prediction using a database search algorithm. *Proteins Struct. Funct. Bioinforma.* **78**, 1431–1440 (2010).

75. Gowthaman, R. & Pierce, B. G. TCRmodel: high resolution modeling of T cell receptors from sequence. *Nucleic Acids Res.* **46**, W396–W401 (2018).

76. Glanville, J. *et al.* Identifying specificity groups in the T cell receptor repertoire. *Nature* **547**, 94–98 (2017).

77. Jurtz, V. I. *et al.* NetTCR: sequence-based prediction of TCR binding to peptide-MHC complexes using convolutional neural networks. *bioRxiv* 433706 (2018) doi:10.1101/433706.

78. Fischer, D. S., Wu, Y., Schubert, B. & Theis, F. J. Predicting antigen-specificity of single T-cells based on TCR CDR3 regions. *bioRxiv* 734053 (2019) doi:10.1101/734053.

79. Jokinen, E., Heinonen, M., Huuhtanen, J., Mustjoki, S. & Lähdesmäki, H. TCRGP: Determining epitope specificity of T cell receptors. *bioRxiv* 542332 (2019) doi:10.1101/542332.

80. Gielis, S. *et al.* TCRex: a webtool for the prediction of T-cell receptor sequence

epitope specificity. *bioRxiv* 373472 (2018) doi:10.1101/373472.

81. Dash, P. *et al.* Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature* **547**, 89–93 (2017).

82. Pogorelyy, M. V *et al.* Detecting T cell receptors involved in immune responses from single repertoire snapshots. *PLOS Biol.* **17**, e3000314 (2019).

83. Laugel, B., Price, D. A., Milicic, A. & Sewell, A. K. CD8 exerts differential effects on the deployment of cytotoxic T lymphocyte effector functions. *Eur. J. Immunol.* **37**, 905–913 (2007).

84. Price, D. A. *et al.* Antigen specific release of chemokines by anti-HIV-1 cytotoxic T lymphocytes. *Curr. Biol.* **8**, 355–358 (1998).

85. Szomolay, B. *et al.* Identification of human viral protein-derived ligands recognized by individual MHCI-restricted T-cell receptors. *Immunol. Cell Biol.* **94**, 573–582 (2016).

86. Abelin, J. G. *et al.* Defining HLA-II Ligand Processing and Binding Rules with Mass Spectrometry Enhances Cancer Epitope Prediction. *Immunity* **51**, 766-779.e17 (2019).

87. Bennek, E. *et al.* Subcellular antigen localization in commensal E. coli is critical for T cell activation and induction of specific tolerance. *Mucosal Immunol.* **12**, 97–107 (2019).

88. Gregg, B. *et al.* Subcellular Antigen Location Influences T-Cell Activation during Acute Infection with Toxoplasma gondii. *PLoS One* **6**, e22936 (2011).

89. Sayle, R. A. & Milner-White, E. J. RASMOL: biomolecular graphics for all. *Trends Biochem. Sci.* **20**, 374–376 (1995).

90. McNicholas, S., Potterton, E., Wilson, K. S. & Noble, M. E. M. Presenting your structures: the CCP4mg molecular-graphics software. *Acta Crystallogr. D. Biol. Crystallogr.* **67**, 386–394 (2011).

91. Schrodinger LLC. The PyMOL Molecular Graphics System, Version 1.8. (2015).

92. Cock, P. J. A. *et al.* Biopython: freely available Python tools for computational molecular biology and bioinformatics. *Bioinformatics* **25**, 1422–1423 (2009).

93. Ireland, S. M. & Martin, A. C. R. atomium—a Python structure parser. *Bioinformatics* (2020) doi:10.1093/bioinformatics/btaa072.

94. Grant, B. J., Rodrigues, A. P. C., ElSawy, K. M., McCammon, J. A. & Caves, L. S. D. Bio3d: an R package for the comparative analysis of protein structures. *Bioinformatics* **22**, 2695–2696 (2006).

95. Winn, M. D. *et al.* research papers Overview of the CCP 4 suite and current developments research papers. *Acta Crystallogr. Sect. D* **4449**, 235–242 (2011).

96. Chakravarty, D., Guharoy, M., Robert, C. H., Chakrabarti, P. & Janin, J. Reassessing buried surface areas in protein-protein complexes. *Protein Sci.* **22**, 1453–1457 (2013).

97. Cole, D. K. *et al.* Structural Mechanism Underpinning Cross-reactivity of a CD8+ T-cell Clone That Recognizes a Peptide Derived from Human Telomerase Reverse Transcriptase. *J. Biol. Chem.* **292**, 802–813 (2017).

98. Fan, S. *et al.* Structural and biochemical analyses of swine MHC class I complexes and prediction of the epitope map in important influenza A strains. *J. Virol.* **90**, JVI.00119-16 (2016).

99. Krissinel, E. & Henrick, K. Inference of Macromolecular Assemblies from Crystalline State. *J. Mol. Biol.* **372**, 774–797 (2007).

100. Pettersen, E. F. *et al.* UCSF Chimera—A visualization system for exploratory research and analysis. *J. Comput. Chem.* **25**, 1605–1612 (2004).

101. Lawrence, M. C. & Colman, P. M. Shape Complementarity at Protein/Protein Interfaces. *J. Mol. Biol.* **234**, 946–950 (1993).

102. Garcia, K. C. *et al.* Structural Basis of Plasticity in T Cell Receptor Recognition of a Self Peptide-MHC Antigen. *Science (80-.).* **279**, 1166 LP – 1172 (1998).

103. Ysern, X., Li, H. & Mariuzza, R. Imperfect interfaces. *Nat. Struct. Biol.* **5**, 412–414 (1998).

104. Winn, M. D. *et al.* Overview of the CCP4 suite and current developments. *Acta Crystallogr. D. Biol. Crystallogr.* **67**, 235–242 (2011).

105. Berman, H. M. *et al.* The Protein Data Bank. *Nucleic Acids Res.* **28**, 235–42 (2000).

106. Dunbar, J. & Deane, C. M. ANARCI: antigen receptor numbering and receptor classification. *Bioinformatics* **32**, btv552 (2015).

107. Eddy, S. R. Accelerated Profile HMM Searches. *PLOS Comput. Biol.* **7**, e1002195 (2011).

108. Lee, B. & Richards, F. M. The interpretation of protein structures: Estimation of static accessibility. *J. Mol. Biol.* **55**, 379-IN4 (1971).

109. Krissinel, E. & Henrick, K. Inference of Macromolecular Assemblies from Crystalline State. *J. Mol. Biol.* **372**, 774–797 (2007).

110. Lawrence, M. C. & Colman, P. M. Shape Complementarity at Protein/Protein Interfaces. *J. Mol. Biol.* **234**, 946–950 (1993).

111. Lefranc, M. *et al.* IMGT, the international ImMunoGeneTics information system 25 years on. *Nucleic Acids Res.* **43**, 413–422 (2015).

112. L., M. J. *et al.* Recommendations for the presentation of NMR structures of proteins and nucleic acids. *Eur. J. Biochem.* **256**, 1–15 (2001).

113. Adams, J. J. *et al.* T cell receptor signaling is limited by docking geometry to peptide-Major Histocompatibility Complex. *Immunity* **35**, 681–693 (2011).

114. Beringer, D. X. *et al.* T cell receptor reversed polarity recognition of a self-antigen major histocompatibility complex. *Nat. Immunol.* **16**, 1153 (2015).

115. Gerard, D. Data-based RNA-seq simulations by binomial thinning. *BMC Bioinformatics* **21**, 206 (2020).

116. Frazee, A. C., Jaffe, A. E., Langmead, B. & Leek, J. T. Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics* **31**, 2778–2784 (2015).

117. Johnson, S., Trost, B., Long, J. R., Pittet, V. & Kusalik, A. A better sequence-read simulator program for metagenomics. *BMC Bioinformatics* **15**, S14 (2014).

118. Angly, F. E., Willner, D., Rohwer, F., Hugenholtz, P. & Tyson, G. W. Grinder: a versatile amplicon and shotgun sequence simulator. *Nucleic Acids Res.* **40**, e94–e94 (2012).

119. Weber, C. R. *et al.* immuneSIM: tunable multi-feature simulation of B- and T-cell receptor repertoires for immunoinformatics benchmarking. *Bioinformatics* **36**, 3594–3596 (2020).

120. Leem, J., de Oliveira, S. H. P., Krawczyk, K. & Deane, C. M. STCRDab: the structural T-cell receptor database. *Nucleic Acids Res.* **46**, D406–D412 (2018).

121. Gowthaman, R. & Pierce, B. G. TCR3d: The T cell receptor structural repertoire database. *Bioinformatics* **35**, 5323–5325 (2019).

122. Hahn, M., Nicholson, M. J., Pyrdol, J. & Wucherpfennig, K. W. Unconventional topology of self peptide–major histocompatibility complex binding by a human autoimmune T cell receptor. *Nat. Immunol.* **6**, 490–496 (2005).

123. Gray, B. P. & Brown, K. C. Combinatorial peptide libraries: mining for cell-binding peptides. *Chem. Rev.* **114**, 1020–1081 (2014).

124. Miles, J. J. *et al.* Peptide mimic for influenza vaccination using nonnatural combinatorial chemistry. *J. Clin. Invest.* **128**, 1569–1580 (2018).

125. Dagum, L. & Menon, R. OpenMP: An Industry-Standard API for Shared-Memory Programming. *IEEE Comput. Sci. Eng.* **5**, 46–55 (1998).

126. Forum, M. P. *MPI: A Message-Passing Interface Standard*. (1994).

127. Manocha, D. General-purpose computations using graphics processors. *Computer (Long. Beach. Calif).* **38**, 85–88 (2005).

128. Stone, J. E., Gohara, D. & Shi, G. OpenCL: A Parallel Programming Standard for Heterogeneous Computing Systems. *Comput. Sci. Eng.* **12**, 66–73 (2010).

129. Karimi, K., Dickson, N. G. & Hamze, F. A Performance Comparison of CUDA and OpenCL. (2010).

130. Corporation, N. *NVIDIA CUDA C Programming Guide*. (2010).

131. Nobile, M. S., Cazzaniga, P., Tangherloni, A. & Besozzi, D. Graphics processing units in bioinformatics, computational biology and systems biology. *Brief. Bioinform.* **18**, 870–885 (2016).

132. Blom, J. *et al.* Exact and complete short-read alignment to microbial genomes using Graphics Processing Unit programming. *Bioinformatics* **27**, 1351–1358 (2011).

133. Klus, P. *et al.* BarraCUDA - a fast short read sequence aligner using graphics processing units. *BMC Res. Notes* **5**, 27 (2012).

134. Liu, W., Schmidt, B. & Muller-Wittig, W. CUDA-BLASTP: Accelerating BLASTP on CUDA-Enabled Graphics Hardware. *IEEE/ACM Trans. Comput. Biol. Bioinforma.* **8**, 1678–1684 (2011).

135. Cheng, L. & Butler, G. Accelerating Search of Protein Sequence Databases using CUDA-Enabled GPU. in *DASFAA* (2015).

136. Jiang, H. & Ganesan, N. CUDAMPF: a multi-tiered parallel framework for accelerating protein sequence search in HMMER on CUDA-enabled GPU. *BMC Bioinformatics* **17**, 106 (2016).

137. Stehle, E. & Jacobsen, H.-A. A Memory Bandwidth-Efficient Hybrid Radix Sort on GPUs. in *Proceedings of the 2017 ACM International Conference on Management of Data* 417–432 (Association for Computing Machinery, 2017). doi:10.1145/3035918.3064043.

138. Tanizawa, Y., Fujisawa, T. & Nakamura, Y. DFAST: a flexible prokaryotic genome annotation pipeline for faster genome publication. *Bioinformatics* **34**, 1037–1039 (2017).

139. Wood, D. E., Lu, J. & Langmead, B. Improved metagenomic analysis with Kraken 2. *bioRxiv* 762302 (2019) doi:10.1101/762302.

140. Ondov, B. D. *et al.* Mash: fast genome and metagenome distance estimation using MinHash. *Genome Biol.* **17**, 132 (2016).

141. Baker, D. N. & Langmead, B. Dashing: fast and accurate genomic distances with HyperLogLog. *Genome Biol.* **20**, 265 (2019).

142. He, Y., Xiang, Z. & Mobley, H. L. T. Vaxign: the first web-based vaccine design program for reverse vaccinology and applications for vaccine development. *J. Biomed. Biotechnol.* **2010**, 297505 (2010).

143. Doytchinova, I. A. & Flower, D. R. VaxiJen: a server for prediction of protective antigens, tumour antigens and subunit vaccines. *BMC Bioinformatics* **8**, 4 (2007).

144. Goodswen, S. J., Kennedy, P. J. & Ellis, J. T. Vacceed: a high-throughput in silico vaccine candidate discovery pipeline for eukaryotic pathogens based on reverse vaccinology. *Bioinformatics* **30**, 2381–2383 (2014).

145. Rizwan, M. *et al.* VacSol: a high throughput in silico pipeline to predict potential therapeutic targets in prokaryotic pathogens using subtractive reverse vaccinology. *BMC Bioinformatics* **18**, 106 (2017).

146. Fay, J. C., Wyckoff, G. J. & Wu, C.-I. Positive and Negative Selection on the Human Genome. *Genetics* **158**, 1227 LP – 1234 (2001).

147. Tajima, F. Statistical Method for Testing the Neutral Mutation Hypothesis by DNA Polymorphism. *Genetics* **123**, 585–595 (1989).

148. Fu, Y. X. & Li, W. H. Statistical tests of neutrality of mutations. *Genetics* **133**, 693–709 (1993).

149. Fay, J. C. & Wu, C. I. Hitchhiking under positive Darwinian selection. *Genetics* **155**, 1405–1413 (2000).

150. Strait, B. J. & Dewey, T. G. The Shannon information entropy of protein sequences. *Biophys. J.* **71**, 148–155 (1996).

151. Vita, R. *et al.* The Immune Epitope Database (IEDB): 2018 update. *Nucleic Acids Res.* **47**, D339–D343 (2018).

152. Jespersen, M. C., Peters, B., Nielsen, M. & Marcatili, P. BepiPred-2.0: improving sequence-based B-cell epitope prediction using conformational epitopes. *Nucleic Acids Res.* **45**, W24–W29 (2017).

153. Parry, C. M., Hien, T. T., Dougan, G., White, N. J. & Farrar, J. J. Typhoid fever. *N. Engl. J. Med.* **347**, 1770–1782 (2002).

154. Levine, M. M., Tapia, M. D. & Zaidi, A. K. M. CHAPTER 16 - Typhoid and Paratyphoid (Enteric) Fever. in (eds. Guerrant, R. L., Walker, D. H. & Weller Pathogens and Practice (Third Edition), P. F. B. T.-T. I. D. P.) 121–127 (W.B. Saunders, 2011).

doi:https://doi.org/10.1016/B978-0-7020-3935-5.00016-1.

155. Stanaway, J. D. *et al.* The global burden of typhoid and paratyphoid fevers: a systematic analysis for the Global Burden of Disease Study 2017. *Lancet Infect. Dis.* **19**, 369–381 (2019).

156. Engels, E. A., Falagas, M. E., Lau, J. & Bennish, M. L. Typhoid fever vaccines: a meta-analysis of studies on efficacy and toxicity. *BMJ* **316**, 110 LP – 116 (1998).

157. Amicizia, D., Arata, L., Zangrillo, F., Panatto, D. & Gasparini, R. Overview of the impact of Typhoid and Paratyphoid fever. Utility of Ty21a vaccine (Vivotif®). *J. Prev. Med. Hyg.* **58**, E1–E8 (2017).

158. Wierzba, T. F. & Shetty, A. K. Vi-TT&#x2014;a typhoid conjugate vaccine for infants and young children. *Lancet Glob. Heal.* **9**, e1483–e1484 (2021).

159. Lin, F. Y. *et al.* The efficacy of a Salmonella typhi Vi conjugate vaccine in two-to-five-year-old children. *N. Engl. J. Med.* **344**, 1263–1269 (2001).

160. Firoz Mian, M., Pek, E. A., Chenoweth, M. J. & Ashkar, A. A. Humanized mice are susceptible to Salmonella typhi infection. *Cell. Mol. Immunol.* **8**, 83–87 (2011).

161. Santos, R. L. *et al.* Animal models of Salmonella infections: enteritis versus typhoid fever. *Microbes Infect.* **3**, 1335–1344 (2001).

162. House, D., Bishop, A., Parry, C., Dougan, G. & Wain, J. Typhoid fever: pathogenesis and disease. *Curr. Opin. Infect. Dis.* **14**, 573–578 (2001).

163. Kraehenbuhl, J. P. & Neutra, M. R. Epithelial M cells: differentiation and function. *Annu. Rev. Cell Dev. Biol.* **16**, 301–332 (2000).

164. Jones, B. D., Ghori, N. & Falkow, S. Salmonella typhimurium initiates murine infection by penetrating and destroying the specialized epithelial M cells of the Peyer's patches. *J. Exp. Med.* **180**, 15–23 (1994).

165. Hameleers, D. M. H., van der Ende, M., Biewenga, J. & Sminia, T. An immunohistochemical study on the postnatal development of rat nasal-associated lymphoid tissue (NALT). *Cell Tissue Res.* **256**, 431–438 (1989).

166. Pham, O. H. & McSorley, S. J. Protective host immune responses to Salmonella infection. *Future Microbiol.* **10**, 101–110 (2015).

167. McSorley, S. J., Asch, S., Costalonga, M., Reinhardt, R. L. & Jenkins, M. K. Tracking Salmonella-Specific CD4 T Cells In Vivo Reveals a Local Mucosal Response to a Disseminated Infection. *Immunity* **16**, 365–377 (2002).

168. Voedisch, S. *et al.* Mesenteric lymph nodes confine dendritic cell-mediated dissemination of Salmonella enterica serovar Typhimurium and limit systemic disease in mice. *Infect. Immun.* **77**, 3170–3180 (2009).

169. Tam, M. A., Rydström, A., Sundquist, M. & Wick, M. J. Early cellular responses to Salmonella infection: dendritic cells, monocytes, and more. *Immunol. Rev.* **225**, 140–162 (2008).

170. Andrés, V.-T. *et al.* Salmonella Pathogenicity Island 2-Dependent Evasion of the Phagocyte NADPH Oxidase. *Science (80-.).* **287**, 1655–1658 (2000).

171. Conlan, J. W. Neutrophils prevent extracellular colonization of the liver microvasculature by Salmonella typhimurium. *Infect. Immun.* **64**, 1043–1047 (1996).

172. Klose, C. S. N. *et al.* A T-bet gradient controls the fate and function of CCR6-RORγt+ innate lymphoid cells. *Nature* **494**, 261–265 (2013).

173. Rydström, A. & Wick, M. J. Monocyte Recruitment, Activation, and Function in the Gut-Associated Lymphoid Tissue during Oral Salmonella Infection. *J. Immunol.* **178**, 5789 LP – 5801 (2007).

174. Kupz, A. *et al.* NLRC4 inflammasomes in dendritic cells regulate noncognate effector function by memory CD8^{+} T cells. *Nat. Immunol.* **13**, 162–169 (2012).

175. Sierro, F. *et al.* Flagellin stimulation of intestinal epithelial cells triggers CCL20-mediated migration of dendritic cells. *Proc. Natl. Acad. Sci.* **98**, 13722 LP – 13727 (2001).

176. Chen, Z. M. & Jenkins, M. K. Clonal expansion of antigen-specific CD4 T cells following infection with Salmonella typhimurium is similar in susceptible (Itys) and resistant (Ityr) BALB/c mice. *Infect. Immun.* **67**, 2025–2029 (1999).

177. Hess, J., Ladel, C., Miko, D. & Kaufmann, S. H. Salmonella typhimurium aroA- infection in gene-targeted immunodeficient mice: major role of CD4+ TCR-alpha beta cells and IFN-gamma in bacterial clearance independent of intracellular location. *J. Immunol.* **156**, 3321 LP – 3326 (1996).

178. Lee, S.-J., Dunmire, S. & McSorley, S. J. MHC class-I-restricted CD8 T cells play a protective role during primary Salmonella infection. *Immunol. Lett.* **148**, 138–143 (2012).

179. Mastroeni, P., Villarreal-Ramos, B. & Hormaeche, C. E. Adoptive transfer of immunity to oral challenge with virulent salmonellae in innately susceptible BALB/c mice requires both immune serum and T cells. *Infect. Immun.* **61**, 3981–3984 (1993).

180. Barr, T. A., Brown, S., Mastroeni, P. & Gray, D. TLR and B cell receptor signals to B cells differentially program primary and memory Th1 responses to Salmonella enterica. *J. Immunol.* **185**, 2783–2789 (2010).

181. C., K. A., Malin, S. & Jo, W. M. In Vivo Compartmentalization of Functionally Distinct, Rapidly Responsive Antigen-Specific T-Cell Populations in DNA-Immunized or Salmonella enterica Serovar Typhimurium-Infected Mice. *Infect. Immun.* **72**, 6390–6400 (2004).

182. Nelson, R. W., McLachlan, J. B., Kurtz, J. R. & Jenkins, M. K. CD4+ T cell persistence and function after infection are maintained by low-level peptide:MHC class II presentation. *J. Immunol.* **190**, 2828–2834 (2013).

183. Lee, S.-J. *et al.* Temporal Expression of Bacterial Proteins Instructs Host CD4 T Cell Expansion and Th17 Development. *PLOS Pathog.* **8**, e1002499 (2012).

184. Berg, R. E., Cordes, C. J. & Forman, J. Contribution of CD8+ T cells to innate immunity: IFN-γ secretion induced by IL-12 and IL-18. *Eur. J. Immunol.* **32**, 2807–2816 (2002).

185. Broz, P. & Monack, D. M. Molecular mechanisms of inflammasome activation during microbial infections. *Immunol. Rev.* **243**, 174–190 (2011).

186. Barat, S. *et al.* Immunity to Intracellular Salmonella Depends on Surface-associated Antigens. *PLOS Pathog.* **8**, e1002966 (2012).

187. Gomes, T. A. T. *et al.* Diarrheagenic Escherichia coli. *Brazilian J. Microbiol.* **47**, 3–30 (2016).

188. Deborah Chen, H. & Frankel, G. Enteropathogenic Escherichia coli: unravelling pathogenesis. *FEMS Microbiol. Rev.* **29**, 83–98 (2005).

189. Ingle, D. J. *et al.* In silico serotyping of E. coli from short read data identifies limited novel O-loci but extensive diversity of O:H serotype combinations within and between pathogenic lineages. *Microb. genomics* **2**, e000064–e000064 (2016).

190. Nataro, J. P. & Kaper, J. B. Diarrheagenic Escherichia coli. *Clin. Microbiol. Rev.* **11**, 142–201 (1998).

191. Dean, P. & Kenny, B. The effector repertoire of enteropathogenic E. coli: ganging up on the host cell. *Curr. Opin. Microbiol.* **12**, 101–109 (2009).

192. Batchelor, M. *et al.* Structural basis for recognition of the translocated intimin receptor (Tir) by intimin from enteropathogenic Escherichia coli. *EMBO J.* **19**, 2452–2464 (2000).

193. Berger, C. N. *et al.* EspZ of enteropathogenic and enterohemorrhagic Escherichia coli regulates type III secretion system protein translocation. *MBio* **3**, (2012).

194. Trabulsi, L. R., Keller, R. & Tardelli Gomes, T. A. Typical and atypical enteropathogenic Escherichia coli. *Emerg. Infect. Dis.* **8**, 508–513 (2002).

195. Girón, J. A., Ho, A. S. & Schoolnik, G. K. An inducible bundle-forming pilus of enteropathogenic Escherichia coli. *Science* **254**, 710–713 (1991).

196. Bieber, D. *et al.* Type IV pili, transient bacterial aggregates, and virulence of enteropathogenic Escherichia coli. *Science* **280**, 2114–2118 (1998).

197. Moreira, C. G. *et al.* Bundle-forming pili and EspA are involved in biofilm formation by enteropathogenic Escherichia coli. *J. Bacteriol.* **188**, 3952–3961 (2006).

198. Rasko, D. A., Myers, G. S. A. & Ravel, J. Visualization of comparative genomic analyses by BLAST score ratio. *BMC Bioinformatics* **6**, 2 (2005).

199. Sahl, J. W., Caporaso, J. G., Rasko, D. A. & Keim, P. The large-scale blast score ratio

(LS-BSR) pipeline: a method to rapidly compare genetic content between bacterial genomes. *PeerJ* **2**, e332 (2014).

200. Hazen, T. H. *et al.* Genomic diversity of EPEC associated with clinical presentations of differing severity. *Nat. Microbiol.* **1**, 15014 (2016).

201. Pham, T. H., Gao, X., Singh, G. & Hardwidge, P. R. Escherichia coli virulence protein NleH1 interaction with the v-Crk sarcoma virus CT10 oncogene-like protein (CRKL) governs NleH1 inhibition of the ribosomal protein S3 (RPS3)/nuclear factor κB (NF-κB) pathway. *J. Biol. Chem.* **288**, 34567–34574 (2013).

202. El-Aouar Filho, R. A. *et al.* Heterogeneous Family of Cyclomodulins: Smart Weapons That Allow Bacteria to Hijack the Eukaryotic Cell Cycle and Promote Infections . *Frontiers in Cellular and Infection Microbiology* vol. 7 208 (2017).

203. Marchès, O. *et al.* EspJ of enteropathogenic and enterohaemorrhagic Escherichia coli inhibits opsono-phagocytosis. *Cell. Microbiol.* **10**, 1104–1115 (2008).

204. Valleau, D. *et al.* Functional diversification of the NleG effector family in enterohemorrhagic Escherichia coli; *Proc. Natl. Acad. Sci.* **115**, 10004 LP – 10009 (2018).

205. Gao, X. *et al.* NleB, a bacterial effector with glycosyltransferase activity, targets GAPDH function to inhibit NF-κB activation. *Cell Host Microbe* **13**, 87–99 (2013).

206. Hodgson, A. *et al.* Metalloprotease NleC Suppresses Host NF-κB/Inflammatory Responses by Cleaving p65 and Interfering with the p65/RPS3 Interaction. *PLOS Pathog.* **11**, e1004705 (2015).

207. Kristina, C. *et al.* The Type III Effector NleD from Enteropathogenic Escherichia coli Differentiates between Host Substrates p38 and JNK. *Infect. Immun.* **85**, e00620-16 (2021).

208. Kralicek, S. E., Nguyen, M., Rhee, K.-J., Tapia, R. & Hecht, G. EPEC NleH1 is significantly more effective in reversing colitis and reducing mortality than NleH2 via differential effects on host signaling pathways. *Lab. Invest.* **98**, 477–488 (2018).

209. Blasche, S. *et al.* The E. coli Effector Protein NleF Is a Caspase Inhibitor. *PLoS One* **8**,

e58937 (2013).

210. Chatterjee, S. *et al.* The type III secretion system effector EspO of enterohaemorrhagic Escherichia coli inhibits apoptosis through an interaction with HAX-1. *Cell. Microbiol.* **23**, e13366 (2021).

211. Loureiro, I. *et al.* Human colostrum contains IgA antibodies reactive to enteropathogenic Escherichia coli virulence-associated proteins: intimin, BfpA, EspA, and EspB. *J. Pediatr. Gastroenterol. Nutr.* **27**, 166–171 (1998).

212. Parissi-Crivelli, A., Parissi-Crivelli, J. M. & Girón, J. A. Recognition of enteropathogenic Escherichia coli virulence determinants by human colostrum and serum antibodies. *J. Clin. Microbiol.* **38**, 2696–2700 (2000).

213. Rojas-Lopez, M., Monterio, R., Pizza, M., Desvaux, M. & Rosini, R. Intestinal Pathogenic Escherichia coli: Insights for Vaccine Development. *Front. Microbiol.* **9**, 440 (2018).

214. Basnayake, S. K. & Easterbrook, P. J. Wide variation in estimates of global prevalence and burden of chronic hepatitis B and C infection cited in published literature. *J. Viral Hepat.* **23**, 545–559 (2016).

215. Lozano, R. *et al.* Global and regional mortality from 235 causes of death for 20 age groups in 1990 and 2010: a systematic analysis for the Global Burden of Disease Study 2010. *Lancet (London, England)* **380**, 2095–2128 (2012).

216. Liang, T. J. Hepatitis B: The virus and disease. *Hepatology* **49**, S13–S21 (2009).

217. Milich, D. & Liang, T. J. Exploring the biological basis of hepatitis B e antigen in hepatitis B virus infection. *Hepatology* **38**, 1075–1086 (2003).

218. Bertoletti, A., Maini, M. K. & Ferrari, C. The host-pathogen interaction during HBV infection: immunological controversies. *Antivir. Ther.* **15 Suppl 3**, 15–24 (2010).

219. Dunn, C. *et al.* Temporal analysis of early immune responses in patients with acute hepatitis B virus infection. *Gastroenterology* **137**, 1289–1300 (2009).

220. Wieland, S. F. & Chisari, F. V. Stealth and cunning: hepatitis B and hepatitis C viruses. *J. Virol.* **79**, 9369–9380 (2005).

221. Durantel, D. & Zoulim, F. Innate response to hepatitis B virus infection: observations challenging the concept of a stealth virus. *Hepatology (Baltimore, Md.)* vol. 50 1692–1695 (2009).

222. Alberti, A., Diana, S., Sculard, G. H., Eddleston, A. L. & Williams, R. Detection of a new antibody system reacting with Dane particles in hepatitis B virus infection. *Br. Med. J.* **2**, 1056–1058 (1978).

223. Webster, G. J. *et al.* Incubation phase of acute hepatitis B in man: dynamic of cellular immune mechanisms. *Hepatology* **32**, 1117–1124 (2000).

224. Tan, A., Koh, S. & Bertoletti, A. Immune Response in Hepatitis B Virus Infection. *Cold Spring Harb. Perspect. Med.* **5**, a021428–a021428 (2015).

225. Guidotti, L. G. & Chisari, F. V. Immunobiology and pathogenesis of viral hepatitis. *Annu. Rev. Pathol.* **1**, 23–61 (2006).

226. Iannacone, M., Sitia, G., Ruggeri, Z. M. & Guidotti, L. G. HBV pathogenesis in animal models: recent advances on the role of platelets. *J. Hepatol.* **46**, 719–726 (2007).

227. Wherry, E. J. & Ahmed, R. Memory CD8 T-cell differentiation during viral infection. *J. Virol.* **78**, 5535–5545 (2004).

228. Pallett, L. J. *et al.* Metabolic regulation of hepatitis B immunopathology by myeloid-derived suppressor cells. *Nat. Med.* **21**, 591–600 (2015).

229. Sandalova, E. *et al.* Increased levels of arginase in patients with acute hepatitis B suppress antiviral T cells. *Gastroenterology* **143**, 78-87.e3 (2012).

230. Ye, B. *et al.* T-cell exhaustion in chronic hepatitis B infection: current knowledge and clinical significance. *Cell Death Dis.* **6**, e1694–e1694 (2015).

231. Kennedy, P. T. F. *et al.* Preserved T-cell function in children and young adults with immune-tolerant chronic hepatitis B. *Gastroenterology* **143**, 637–645 (2012).

232. Granowitz, E. V *et al.* Intravenous endotoxin suppresses the cytokine response of peripheral blood mononuclear cells of healthy humans. *J. Immunol.* **151**, 1637–1645 (1993).

233. Gehring, A. J. *et al.* Mobilizing monocytes to cross-present circulating viral antigen in chronic infection. *J. Clin. Invest.* **123**, 3766–3776 (2013).

234. Kosinska, A. D. *et al.* Combination of DNA prime--adenovirus boost immunization with entecavir elicits sustained control of chronic hepatitis B in the woodchuck model. *PLoS Pathog.* **9**, e1003391 (2013).

235. Shu, Y. & McCauley, J. GISAID: Global initiative on sharing all influenza data - from vision to reality. *Euro Surveill.* **22**, 30494 (2017).

236. Lim, Y. X., Ng, Y. L., Tam, J. P. & Liu, D. X. Human Coronaviruses: A Review of Virus-Host Interactions. *Dis. (Basel, Switzerland)* **4**, 26 (2016).

237. Jungreis, I., Sealfon, R. & Kellis, M. SARS-CoV-2 gene content and COVID-19 mutation impact by comparing 44 Sarbecovirus genomes. *Nat. Commun.* **12**, 2642 (2021).

238. Wrapp, D. *et al.* Cryo-EM structure of the 2019-nCoV spike in the prefusion conformation. *Science* **367**, 1260–1263 (2020).

239. de Wit, E., van Doremalen, N., Falzarano, D. & Munster, V. J. SARS and MERS: recent insights into emerging coronaviruses. *Nat. Rev. Microbiol.* **14**, 523–534 (2016).

240. Dogan, M. *et al.* SARS-CoV-2 specific antibody and neutralization assays reveal the wide range of the humoral immune response to virus. *Commun. Biol.* **4**, 129 (2021).

241. Jiang, H. *et al.* SARS-CoV-2 proteome microarray for global profiling of COVID-19 specific IgG and IgM responses. *Nat. Commun.* **11**, 3581 (2020).

242. Padoan, A. *et al.* IgA-Ab response to spike glycoprotein of SARS-CoV-2 in patients with COVID-19: A longitudinal study. *Clin. Chim. Acta.* **507**, 164–166 (2020).

243. Ibarrondo, F. J. *et al.* Rapid Decay of Anti–SARS-CoV-2 Antibodies in Persons with Mild Covid-19. *N. Engl. J. Med.* **383**, 1085–1087 (2020).

244. Wang, Y. *et al.* Kinetics of viral load and antibody response in relation to COVID-19 severity. *J. Clin. Invest.* **130**, 5235–5244 (2020).

245. Wajnberg, A. *et al.* Robust neutralizing antibodies to SARS-CoV-2 infection persist for months. *Science* **370**, 1227–1230 (2020).

246. Braun, J. *et al.* SARS-CoV-2-reactive T cells in healthy donors and patients with COVID-19. *Nature* **587**, 270–274 (2020).

247. Grifoni, A. *et al.* Targets of T Cell Responses to SARS-CoV-2 Coronavirus in Humans with COVID-19 Disease and Unexposed Individuals. *Cell* **181**, 1489-1501.e15 (2020).

248. Zhou, Z., Alikhan, N.-F., Mohamed, K. & Achtman, M. The user's guide to comparative genomics with EnteroBase. Three case studies: micro-clades within Salmonella enterica serovar Agama, ancient and modern populations of Yersinia pestis;, and core genomic diversity of. *bioRxiv* 613554 (2019) doi:10.1101/613554.

249. Hayer, J. *et al.* HBVdb: a knowledge database for Hepatitis B Virus. *Nucleic Acids Res.* **41**, D566–D570 (2013).

250. Di Tommaso, P. *et al.* Nextflow enables reproducible computational workflows. *Nat. Biotechnol.* **35**, 316–319 (2017).

251. Seemann, T. Prokka: rapid prokaryotic genome annotation. *Bioinformatics* **30**, 2068–2069 (2014).

252. Page, A. J. *et al.* Roary: rapid large-scale prokaryote pan genome analysis. *Bioinformatics* **31**, 3691–3693 (2015).

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

254. Karosiene, E., Lundegaard, C., Lund, O. & Nielsen, M. NetMHCcons: a consensus method for the major histocompatibility complex class I predictions. *Immunogenetics* **64**, (2012).

255. Calis, J. J. A., Reinink, P., Keller, C., Kloetzel, P. M. & Keşmir, C. Role of peptide processing predictions in T cell epitope identification: contribution of different prediction programs. *Immunogenetics* **67**, 85–93 (2015).

256. Goldberg, T. *et al.* LocTree3 prediction of localization. *Nucleic Acids Res.* **42**, W350–W355 (2014).

257. Bianchi, V. *et al.* A Molecular Switch Abrogates Glycoprotein 100 (gp100) T-cell

Receptor (TCR) Targeting of a Human Melanoma Antigen. *J. Biol. Chem.* **291**, 8951–8959 (2016).

258. Osorio, D., Rondon-Villarreal, P. & Torres, R. Peptides: A Package for Data Mining of Antimicrobial Peptides. *R J.* **7**, 4–14 (2015).

259. Guy, A. J., Irani, V., Richards, J. S. & Ramsland, P. A. BioStructMap: a Python tool for integration of protein structure and sequence-based features. *Bioinformatics* **34**, 3942–3944 (2018).

260. Krogh, A., Larsson, B., von Heijne, G. & Sonnhammer, E. L. L. Predicting transmembrane protein topology with a hidden markov model: application to complete genomes11Edited by F. Cohen. *J. Mol. Biol.* **305**, 567–580 (2001).

261. Zhang, R., Ou, H. & Zhang, C. DEG: a database of essential genes. *Nucleic Acids Res.* **32**, D271–D272 (2004).

262. Chen, L., Xiong, Z., Sun, L., Yang, J. & Jin, Q. VFDB 2012 update: toward the genetic diversity and molecular evolution of bacterial virulence factors. *Nucleic Acids Res.* **40**, D641–D645 (2011).

263. Wood, D. E. & Salzberg, S. L. Kraken: ultrafast metagenomic sequence classification using exact alignments. *Genome Biol.* **15**, R46 (2014).

264. Burkhardt, S. & Kärkkäinen, J. Better Filtering with Gapped q-Grams. *Fundam. Informaticae* **56**, 51–70 (2003).

265. Ma, B., Tromp, J. & Li, M. PatternHunter: faster and more sensitive homology search . *Bioinformatics* **18**, 440–445 (2002).

266. LI, M., MA, B. I. N., KISMAN, D. & TROMP, J. PATTERNHUNTER II: HIGHLY SENSITIVE AND FAST HOMOLOGY SEARCH. *J. Bioinform. Comput. Biol.* **02**, 417–439 (2004).

267. Truong, D. T. *et al.* MetaPhlAn2 for enhanced metagenomic taxonomic profiling. *Nat. Methods* **12**, 902–903 (2015).

268. Nguyen, N., Mirarab, S., Liu, B., Pop, M. & Warnow, T. TIPP: taxonomic identification and phylogenetic profiling. *Bioinformatics* **30**, 3548–3555 (2014).

269. Hyatt, D. *et al.* Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics* **11**, 119 (2010).

270. The UniProt Consortium. UniProt: the universal protein knowledgebase. *Nucleic Acids Res.* **45**, D158–D169 (2017).

271. O'Leary, N. A. *et al.* Reference sequence (RefSeq) database at NCBI: current status, taxonomic expansion, and functional annotation. *Nucleic Acids Res.* **44**, D733–D745 (2016).

272. Finn, R. D. *et al.* Pfam: the protein families database. *Nucleic Acids Res.* **42**, D222–D230 (2014).

273. Haft, D. H., Selengut, J. D. & White, O. The TIGRFAMs database of protein families. *Nucleic Acids Res.* **31**, 371–373 (2003).

274. Petersen, T. N., Brunak, S., von Heijne, G. & Nielsen, H. SignalP 4.0: discriminating signal peptides from transmembrane regions. *Nat. Methods* **8**, 785–786 (2011).

275. Li, W. & Godzik, A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* **22**, 1658–1659 (2006).

276. Enright, A. J., Van Dongen, S. & Ouzounis, C. A. An efficient algorithm for large-scale detection of protein families. *Nucleic Acids Res.* **30**, 1575–1584 (2002).

277. Chaudhari, N. M., Gupta, V. K. & Dutta, C. BPGA- an ultra-fast pan-genome analysis pipeline. *Sci. Rep.* **6**, 24373 (2016).

278. Löytynoja, A. Phylogeny-aware alignment with PRANK BT - Multiple Sequence Alignment Methods. in (ed. Russell, D. J.) 155–170 (Humana Press, 2014). doi:10.1007/978-1-62703-646-7_10.

279. Yu, C.-S., Lin, C.-J. & Hwang, J.-K. Predicting subcellular localization of proteins for Gram-negative bacteria by support vector machines based on n-peptide compositions. *Protein Sci.* **13**, 1402–1406 (2004).

280. Yu, N. Y. *et al.* PSORTb 3.0: improved protein subcellular localization prediction with refined localization subcategories and predictive capabilities for all prokaryotes. *Bioinformatics* **26**, 1608–1615 (2010).

281. Horton, P. *et al.* WoLF PSORT: protein localization predictor. *Nucleic Acids Res.* **35**, W585–W587 (2007).

282. Briesemeister, S., Rahnenführer, J. & Kohlbacher, O. YLoc--an interpretable web server for predicting subcellular localization. *Nucleic Acids Res.* **38**, W497–W502 (2010).

283. Sukumaran, J. & Holder, M. T. DendroPy: a Python library for phylogenetic computing. *Bioinformatics* **26**, 1569–1571 (2010).

284. Soares, I., Moleirinho, A., Oliveira, G. N. P. & Amorim, A. DivStat: A User-Friendly Tool for Single Nucleotide Polymorphism Analysis of Genomic Diversity. *PLoS One* **10**, e0119851 (2015).

285. Kumar, S., Stecher, G., Li, M., Knyaz, C. & Tamura, K. MEGA X: Molecular Evolutionary Genetics Analysis across Computing Platforms. *Mol. Biol. Evol.* **35**, 1547–1549 (2018).

286. Librado, P. & Rozas, J. DnaSP v5: a software for comprehensive analysis of DNA polymorphism data. *Bioinformatics* **25**, 1451–1452 (2009).

287. Korneliussen, T. S., Moltke, I., Albrechtsen, A. & Nielsen, R. Calculation of Tajima's D and other neutrality test statistics from low depth next-generation sequencing data. *BMC Bioinformatics* **14**, 289 (2013).

288. Vinga, S. Information theory applications for biological sequence analysis. *Brief. Bioinform.* **15**, 376–389 (2013).

289. Gilpin, W. PyPDB: a Python API for the Protein Data Bank. *Bioinformatics* **32**, 159–160 (2015).

290. Buchfink, B., Xie, C. & Huson, D. H. Fast and sensitive protein alignment using DIAMOND. *Nat. Methods* **12**, 59–60 (2015).

291. Naz, A. *et al.* Identification of putative vaccine candidates against Helicobacter pylori exploiting exoproteome and secretome: A reverse vaccinology based approach. *Infect. Genet. Evol.* **32**, 280–291 (2015).

292. Jensen, K. K. *et al.* Improved methods for predicting peptide binding affinity to MHC class II molecules. *Immunology* **154**, 394–406 (2018).

293. Yang, L.-A., Chang, Y.-J., Chen, S.-H., Lin, C.-Y. & Ho, J.-M. SQUAT: a Sequencing Quality Assessment Tool for data quality assessments of genome assemblies. *BMC Genomics* **19**, 238 (2019).

294. Bankevich, A. *et al.* SPAdes: a new genome assembly algorithm and its applications to single-cell sequencing. *J. Comput. Biol.* **19**, 455–477 (2012).

295. Antipov, D. *et al.* plasmidSPAdes: assembling plasmids from whole genome sequencing data. *Bioinformatics* **32**, 3380–3387 (2016).

296. Corbishley, A. *et al.* Identification of epitopes recognised by mucosal CD4(+) T-cell populations from cattle experimentally colonised with Escherichia coli O157:H7. *Vet. Res.* **47**, 90 (2016).

297. Soltan, M. A. *et al.* In Silico Prediction of a Multitope Vaccine against Moraxella catarrhalis: Reverse Vaccinology and Immunoinformatics. *Vaccines* vol. 9 (2021).

298. Awanye, A. M. *et al.* Immunogenicity profiling of protein antigens from capsular group B Neisseria meningitidis. *Sci. Rep.* **9**, 6843 (2019).

299. Al-Jawdah, A. D. *et al.* Induction of the immunoprotective coat of Yersinia pestis at body temperature is mediated by the Caf1R transcription factor. *BMC Microbiol.* **19**, 68 (2019).

300. Cao, L. *et al.* Vulnerabilities in Yersinia pestis caf operon are unveiled by a Salmonella vector. *PLoS One* **7**, e36283–e36283 (2012).

301. Kolenda, R., Ugorski, M. & Grzymajlo, K. Everything You Always Wanted to Know About Salmonella Type 1 Fimbriae, but Were Afraid to Ask . *Frontiers in Microbiology* vol. 10 1017 (2019).

302. Ahmad, S., Ranaghan, K. E. & Azam, S. S. Combating tigecycline resistant Acinetobacter baumannii: A leap forward towards multi-epitope based vaccine discovery. *Eur. J. Pharm. Sci. Off. J. Eur. Fed. Pharm. Sci.* **132**, 1–17 (2019).

303. Sajjad, R., Ahmad, S. & Azam, S. S. In silico screening of antigenic B-cell derived T-cell epitopes and designing of a multi-epitope peptide vaccine for Acinetobacter nosocomialis. *J. Mol. Graph. Model.* **94**, 107477 (2020).

304. Blot, N., Berrier, C., Hugouvieux-Cotte-Pattat, N., Ghazi, A. & Condemine, G. The oligogalacturonate-specific porin KdgM of Erwinia chrysanthemi belongs to a new porin family. *J. Biol. Chem.* **277**, 7936–7944 (2002).

305. Li, X., Gu, Y., Dong, H., Wang, W. & Dong, C. Trapped lipopolysaccharide and LptD intermediates reveal lipopolysaccharide translocation steps across the Escherichia coli outer membrane. *Sci. Rep.* **5**, 11883 (2015).

306. Zielke, R. A. *et al.* Proteomics-driven Antigen Discovery for Development of Vaccines Against Gonorrhea. *Mol. Cell. Proteomics* **15**, 2338–2355 (2016).

307. Mukherjee, S., Gangopadhay, K. & Mukherjee, S. B. Identification of potential new vaccine candidates in Salmonella typhi; using reverse vaccinology and subtractive genomics-based approach. *bioRxiv* 521518 (2019) doi:10.1101/521518.

308. Zha, Z., Li, C., Li, W., Ye, Z. & Pan, J. LptD is a promising vaccine antigen and potential immunotherapeutic target for protection against Vibrio species infection. *Sci. Rep.* **6**, 38577 (2016).

309. Pérez-Toledo, M. *et al.* Salmonella Typhi Porins OmpC and OmpF Are Potent Adjuvants for T-Dependent and T-Independent Antigens . *Frontiers in Immunology* vol. 8 230 (2017).

310. Secundino, I. *et al.* Salmonella porins induce a sustained, lifelong specific bactericidal antibody memory response. *Immunology* **117**, 59–70 (2006).

311. L., M. K., J., S. T., Susan, G., Russell, B. & Stephen, T. M. The Escherichia coli Phospholipase PldA Regulates Outer Membrane Homeostasis via Lipid Signaling. *MBio* **9**, e00379-18 (2021).

312. Raghunathan, D. *et al.* SadA, a trimeric autotransporter from Salmonella enterica serovar Typhimurium, can promote biofilm formation and provides limited protection against infection. *Infect. Immun.* **79**, 4342–4352 (2011).

313. Foultier, B., Troisfontaines, P., Müller, S., Opperdoes, F. R. & Cornelis, G. R. Characterization of the ysa pathogenicity locus in the chromosome of Yersinia enterocolitica and phylogeny analysis of type III secretion systems. *J. Mol. Evol.* **55**,

37–51 (2002).

314. Chin, C. F. *et al.* Delineation of B-cell Epitopes of Salmonella enterica serovar Typhi Hemolysin E: Potential antibody therapeutic target. *Sci. Rep.* **7**, 2176 (2017).

315. Maybeno, M. *et al.* Polyfunctional CD4+ T Cell Responses to Immunodominant Epitopes Correlate with Disease Activity of Virulent Salmonella. *PLoS One* **7**, e43481 (2012).

316. Singh, S. P., Williams, Y. U., Klebba, P. E., Macchia, P. & Miller, S. Immune recognition of porin and lipopolysaccharide epitopes of Salmonella typhimurium in mice. *Microb. Pathog.* **28**, 157–167 (2000).

317. Zhang, K.-Y. *et al.* Vgas: A Viral Genome Annotation System. *Frontiers in microbiology* vol. 10 184 (2019).

318. Shean, R. C., Makhsous, N., Stoddard, G. D., Lin, M. J. & Greninger, A. L. VAPiD: a lightweight cross-platform viral annotation pipeline and identification tool to facilitate virus genome submissions to NCBI GenBank. *BMC Bioinformatics* **20**, 48 (2019).

319. Crooke, S. N., Ovsyannikova, I. G., Kennedy, R. B. & Poland, G. A. Immunoinformatic identification of B cell and T cell epitopes in the SARS-CoV-2 proteome. *Sci. Rep.* **10**, 14179 (2020).

320. Mengist, H. M., Dilnessa, T. & Jin, T. Structural Basis of Potential Inhibitors Targeting SARS-CoV-2 Main Protease . *Frontiers in Chemistry* vol. 9 (2021).

321. Ullrich, S. & Nitsche, C. The SARS-CoV-2 main protease as drug target. *Bioorg. Med. Chem. Lett.* **30**, 127377 (2020).

322. Scott, B. M., Lacasse, V., Blom, D. G., Tonner, P. D. & Blom, N. S. Predicted coronavirus Nsp5 protease cleavage sites in the human proteome. *BMC Genomic Data* **23**, 25 (2022).

323. Clark, D. N. & Hu, J. Unveiling the roles of HBV polymerase for new antiviral strategies. *Future Virol.* **10**, 283–295 (2015).

324. Akbar, S. M. F. *et al.* Strong and multi-antigen specific immunity by hepatitis B core

antigen (HBcAg)-based vaccines in a murine model of chronic hepatitis B: HBcAg is a candidate for a therapeutic vaccine against hepatitis B virus. *Antiviral Res.* **96**, 59–64 (2012).

325. Li, J. *et al.* Hepatitis B surface antigen (HBsAg) and core antigen (HBcAg) combine CpG oligodeoxynucletides as a novel therapeutic vaccine for chronic hepatitis B infection. *Vaccine* **33**, 4247–4254 (2015).

326. Bichko, V. *et al.* Epitopes recognized by antibodies to denatured core protein of hepatitis B virus. *Mol. Immunol.* **30**, 221–231 (1993).

327. Sällberg, M., Rudén, U., Wahren, B., Noah, M. & Magnius, L. O. Human and murine B-cells recognize the HBeAg/beta (or HBe2) epitope as a linear determinant. *Mol. Immunol.* **28**, 719–726 (1991).

328. Naz, K. *et al.* PanRV: Pangenome-reverse vaccinology approach for identifications of potential vaccine candidates in microbial pangenome. *BMC Bioinformatics* **20**, 123 (2019).

329. Zhao, Y. *et al.* PanGP: A tool for quickly analyzing bacterial pan-genome profile. *Bioinformatics* **30**, 1297–1299 (2014).

330. Mangul, S. *et al.* Challenges and recommendations to improve the installability and archival stability of omics computational tools. *PLoS Biol.* **17**, e3000333–e3000333 (2019).

331. Grüning, B. *et al.* Practical Computational Reproducibility in the Life Sciences. *Cell Syst.* **6**, 631–635 (2018).

332. Fardini, Y. *et al.* Investigation of the role of the BAM complex and SurA chaperone in outer-membrane protein biogenesis and type III secretion system expression in Salmonella. *Microbiology* **155**, 1613–1622 (2009).

333. Kulsum, U., Kapil, A., Singh, H. & Kaur, P. NGSPanPipe: A Pipeline for Pan-genome Identification in Microbial Strains from Experimental Reads. *Adv. Exp. Med. Biol.* **1052**, 39–49 (2018).

334. Nicholls, S. M. *et al.* CLIMB-COVID: continuous integration supporting decentralised

sequencing for SARS-CoV-2 genomic surveillance. *Genome Biol.* **22**, 196 (2021).

335. Szomolay, B. *et al.* Identification of human viral protein-derived ligands recognized by individual MHCI-restricted T-cell receptors. *Immunol. Cell Biol.* **94**, 573–82 (2016).

336. Bijen, H. M. *et al.* Preclinical Strategies to Identify Off-Target Toxicity of High-Affinity TCRs. *Mol. Ther.* **26**, 1206–1214 (2018).

337. Khalilov, M. & Timoveev, A. Performance analysis of CUDA, OpenACC and OpenMP programming models on TESLA V100 GPU. *J. Phys. Conf. Ser.* **1740**, 12056 (2021).

338. Farsal, W., Anter, S. & Ramdani, M. Deep Learning: An Overview. in *Proceedings of the 12th International Conference on Intelligent Systems: Theories and Applications* (Association for Computing Machinery, 2018). doi:10.1145/3289402.3289538.

339. Cole, D. K. *et al.* Modification of MHC anchor residues generates heteroclitic peptides that alter TCR binding and T cell recognition. *J. Immunol.* **185**, 2600–2610 (2010).

340. Jensen, K. K. *et al.* TCRpMHCmodels: Structural modelling of TCR-pMHC class I complexes. *Sci. Rep.* **9**, 14530 (2019).

341. Raybould, M. I. J. *et al.* Five computational developability guidelines for therapeutic antibody profiling. *Proc. Natl. Acad. Sci.* **116**, 4025 LP – 4030 (2019).

342. Cole, D. Increased Peptide Contacts Govern High Affinity Binding of a Modified TCR Whilst Maintaining a Native pMHC Docking Mode . *Frontiers in Immunology* vol. 4 168 (2013).

343. Raybould, M. I. J. *et al.* Thera-SAbDab: the Therapeutic Structural Antibody Database. *Nucleic Acids Res.* **48**, D383–D388 (2020).

344. Thakur, A., Rajput, A. & Kumar, M. MSLVP: prediction of multiple subcellular localization of viral proteins using a support vector machine. *Mol. Biosyst.* **12**, 2572–2586 (2016).

345. Balz, K., Trassl, L., Härtel, V., Nelson, P. P. & Skevaki, C. Virus-Induced T Cell-Mediated Heterologous Immunity and Vaccine Development. *Front. Immunol.* **11**, 513 (2020).

346. Segal, Y. & Shoenfeld, Y. Vaccine-induced autoimmunity: the role of molecular

mimicry and immune crossreaction. *Cell. Mol. Immunol.* **15**, 586–594 (2018).

347. Chiaro, J. *et al.* Viral Molecular Mimicry Influences the Antitumor Immune Response in Murine and Human Melanoma. *medRxiv* 2020.09.09.20191171 (2020) doi:10.1101/2020.09.09.20191171.

14 SUPPLEMENTARY INFORMATION

Supplementary List S1 List of PDB IDs used in the TCR-pMHC structural review in chapters 3, 4 and 5.

1ao7, 1bd2, 1fyt, 1j8h, 1mi5, 1oga, 1qrn, 1qse, 1qsf, 1ymm, 1zgl, 2ak4, 2bnq, 2bnr, 2esv, 2f53, 2f54, 2gj6, 2iam, 2ian, 2nx5, 2p5e, 2p5w, 2pye, 2vlj, 2vlk, 2vlr, 2wbj, 2ypl, 3d39, 3d3v, 3dxa, 3ffc, 3gsn, 3h9s, 3hg1, 3kpr, 3kps, 3kxf, 3mv7, 3mv8, 3mv9, 3o4l, 3o6f, 3pl6, 3pwp, 3qdg, 3qdj, 3qdm, 3qeq, 3qfj, 3sjv, 3t0e, 3uts, 3utt, 3vxm, 3vxr, 3vxs, 3vxu, 3w0w, 4c56, 4e41, 4eup, 4ftv, 4g8g, 4g9f, 4gg6, 4grl, 4h1l, 4jfd, 4jfe, 4jff, 4jrx, 4jry, 4l3e, 4may, 4mji, 4mnq, 4ozf, 4ozg, 4ozh, 4ozi, 4p4k, 4prh, 4pri, 4prp, 4qok, 4qrp, 4y19, 4y1a, 4z7u, 4z7v, 4z7w, 5brz, 5bs0, 5c07, 5c08, 5c09, 5c0a, 5c0b, 5c0c, 5d2l, 5d2n, 5e6i, 5e9d, 5eu6, 5euo, 5hhm, 5hho, 5hyj, 5isz, 5jhd, 5jzi, 5ks9, 5ksa, 5ksb, 5men, 5nht, 5nme, 5nmf, 5nmg, 5nqk, 5tez, 5w1v, 5w1w, 5wkf, 5wkh, 5xot, 5yxn, 5yxu, 6am5, 6amu, 6avf, 6avg, 6bj2, 6bj8, 6cql, 6cqn, 6cqq, 6cqr, 6d78, 6dfx, 6dkp, 6eqa, 6eqb, 6mtm

Supplementary List S2 Enterobase IDs of S. Typhi genomes used in chapters 9, 10 and 11.

10040_15, 10873, 11780, 11783, 15604, 15619, 160005TY, 160008TY, 160010TY, 160011TY, 203377, 203378, 203379, 205588, 206713, 207584, 208724, 208725, 208761, 209994, 211143, 211756, 212757, 212761, 214578, 214579, 214714, 216315, 216339, 216994, 218472, 220459, 220513, 221700, 221701, 222161, 224130, 22420_1_100_Pak8770_2017, 22420_1_10_Pak60006_2016, 22420_1_11_Pak60086_2016, 22420_1_12_Pak60092_2016, 22420_1_13_Pak60306_2016, 22420_1_14_Pak60352_2016, 22420_1_15_Pak0083_2017, 22420_1_16_Pak6802_2017, 22420_1_17_Pak0416_2017, 22420_1_18_Pak0417_2017, 22420_1_19_Pak0364_2017, 22420_1_1_Pak55334_2016, 22420_1_20_Pak0706_2017, 22420_1_21_Pak0696_2017, 22420_1_22_Pak0582_2017, 22420_1_23_Pak1020_2017, 22420_1_24_Pak1172_2017, 22420_1_25_Pak1502_2017, 22420_1_26_Pak1421_2017, 22420_1_27_Pak1672_2017, 22420_1_28_Pak4421_2017, 22420_1_29_Pak2591_2017, 22420_1_2_Pak53977_2016, 22420_1_30_Pak1783_2017, 22420_1_31_Pak3349_2017, 22420_1_32_Pak2749_2017, 22420_1_33_Pak3748_2017, 22420_1_34_Pak4006_2017, 22420_1_35_Pak1986_2017, 22420_1_36_Pak1908_2017, 22420_1_37_Pak3963_2017, 22420_1_38_Pak250_2017, 22420_1_39_Pak1352_2017, 22420_1_3_Pak55719_2016, 22420_1_40_Pak4108_2017, 22420_1_41_Pak7563_2017, 22420_1_42_Pak6072_2017, 22420_1_43_Pak4714_2017, 22420_1_44_Pak6578_2017, 22420_1_45_Pak6048_2017, 22420_1_46_Pak6628_2017, 22420_1_47_Pak6547_2017, 22420_1_48_Pak5726_2017, 22420_1_49_Pak0037_2017, 22420_1_4_Pak56360_2016, 22420_1_50_Pak4590_2017, 22420_1_51_Pak5339_2017, 22420_1_52_Pak4016_2017, 22420_1_53_Pak11167_2017, 22420_1_54_Pak11560_2017, 22420_1_55_Pak11570_2017, 22420_1_56_Pak9487_2017, 22420_1_57_Pak0034_2017, 22420_1_58_Pak7882_2017, 22420_1_59_Pak005_2017, 22420_1_5_Pak56690_2016, 22420_1_60_Pak11731_2017, 22420_1_61_Pak9394_2017, 22420_1_62_Pak9695_2017, 22420_1_63_Pak7927_2017, 22420_1_64_Pak12390_2017, 22420_1_65_Pak11726_2017, 22420_1_66_Pak7778_2017, 22420_1_67_Pak7738_2017, 22420_1_68_Pak10804_2017, 22420_1_69_Pak10757_2017, 22420_1_6_Pak59032_2016, 22420_1_70_Pak8215_2017, 22420_1_71_Pak10934_2017, 22420_1_72_Pak12180_2017, 22420_1_73_Pak12510_2017, 22420_1_74_Pak11964_2017, 22420_1_75_Pak9116_2017, 22420_1_76_Pak9267_2017, 22420_1_77_Pak11957_2017, 22420_1_78_Pak9392_2017, 22420_1_79_Pak11476_2017, 22420_1_7_Pak59027_2016, 22420_1_80_Pak8382_2017, 22420_1_81_Pak8291_2017, 22420_1_82_Pak10550_2017, 22420_1_83_Pak8999_2017, 22420_1_84_Pak0016_2017, 22420_1_85_Pak0017_2017, 22420_1_86_Pak0022_2017, 22420_1_87_Pak0024_2017, 22420_1_88_Pak4019_2017, 22420_1_89_Pak56419_2016, 22420_1_8_Pak59711_2016, 22420_1_90_Pak59691_2016, 22420_1_91_Pak6105_2016, 22420_1_92_Pak59655_2016, 22420_1_93_Pak52035_2016, 22420_1_94_Pak60168_2016, 22420_1_95_Pak57387_2016, 22420_1_96_Pak60320_2016, 22420_1_97_Pak59664_2016, 22420_1_98_Pak15186_2017, 22420_1_99_Pak60147_2016, 22420_1_9_Pak59919_2016, 229066, 229163, 229164, 236189, 236191, 236253, 238380, 238381, 238765, 240164, 242364, 243876, 245587, 247223, 247245, 247247, 247255, 248621, 248624, 250722, 250723, 250726, 250728, 250761, 250762, 252398, 253869, 253870, 253871, 253873, 253874, 253875, 254622, 254624, 256122, 257032, 257355, 259265, 259414, 261332, 261906, 264018, 264101, 264102, 265726, 267719, 267722, 267805, 267844, 268671, 271053, 278082, 278161, 278191, 280356, 282412, 282584, 282585, 282589, 285801, 285842, 285844, 285963, 287766, 288895, 289070, 289677, 291713, 291747, 291748, 293002, 293010, 294695, 294969, 296162, 296163, 296207, 296222, 296635, 296820, 296822, 298759, 298761, 298765, 298767, 298897, 298952, 298960, 299395, 299396, 299909, 300657, 300658, 300663, 300759, 301855, 302512, 302602, 302604, 302607, 302872, 302915, 302921, 302924, 304415, 304416, 304424, 305214, 305215, 305258, 306526, 306532, 306588, 307466, 308528, 308534, 308545, 308759, 310264, 310271, 311418, 311422, 311960, 313152, 313840, 313877, 313878, 315157, 316191, 316277, 316347, 316658, 316659, 316660, 318344, 318354, 318355, 318582, 318583, 319724, 319726, 319822, 320181, 320184, 320272, 320994, 320995, 320997, 328583, 329141, 329871, 329879, 329884, 329887, 330438, 330503, 333376, 333912, 335038, 335044, 337070, 337082, 337114, 337622, 337623, 338394, 338438, 339489, 339493, 340181, 340238, 340240, 340241, 340242, 341269, 341274, 341275, 341276, 342544, 342550, 342643, 342648, 342649, 342650, 343717, 343799, 345581, 345582, 346583, 346893, 346899, 347952, 348081, 348734, 351267, 351324, 352822, 353848, 353870, 353877, 353885, 355129, 356097, 356299, 356314, 356321, 356507, 356531, 357750, 357979, 358438, 358440, 360345, 362107, 364177, 365314, 365365, 365366, 366287, 367315, 369054, 369949, 369998, 370898, 370912, 371794, 374594, 378610, 387213, 391124, 423183, 429038, 430040, 458426, 490271, 568310, 603405, 611427, 636302, 643112, 663131, 672572, 672574, 676410, 678025, 680084, 681355, 686749, 690325, 697897, 724583, 7246, 730774, 730973, 735088, 749413, 767624, 768835, 772865, 782094, 783201, 798438, 800602, 801143, 801489, 808871, 812070, 814955, 815563, 816253, 818193, 824393, 824494, 834291, 834848, 871213, 877298, 879304, 879991, 880355, 882861, 884420, 884427, 885154, 885163, 888113, 890414, 892742, 895926, 903519, 903551, 904644, 904656, 905412, 908986, 914495, 917873, 920101, AM-51471, BA1321, BA2820, BA7428, BA7569, BP200, BP2343, BP2397, BP2608, BV145, ERS1545197, ERS1545200, ERS1545201, ERS1545202, ERS1545203, ERS1545204, ERS1545205, ERS1545206, ERS1545207, ERS1545208, ERS1545209, ERS1545211, ERS1545212, ERS1545213, ERS1545214, ERS1545216, ERS1545217, ERS1545218, ERS1545219, ERS1545220, ERS1545221, ERS1545222, ERS1545223, ERS1545622, ERS1545623, ERS1545624, ERS1545625, ERS1545626, ERS1545627, ERS1545628, ERS1545629, ERS1545630, ERS1545631, ERS1545632, ERS1545633, ERS1545634, ERS1545635, ERS1545636, ERS1545637, ERS1545638, ERS1545639, ERS1545640, ERS1545641, ERS1545642, ERS1545643, ERS1545644, ERS1545645, ERS1545647, ERS1545648, ERS1545649, ERS1545650, ERS1545651, ERS1545652, ERS1545653, ERS1545654, ERS1545655, ERS1545656, ERS1545657, ERS1810832, ERS1810833, ERS1810834, ERS1810835, ERS1810836, ERS1810837, ERS1810839, ERS1810840, ERS1810841, ERS1810842, ERS1810843, ERS1810844, ERS1810845, ERS1810846, ERS1810863, ERS1867173, ERS1867174, ERS1867175, ERS1867177, ERS1867178, ERS1867179, ERS1867180, ERS1867181, ERS1867182, ERS1867183, ERS1867184, ERS1867185, ERS1867186, ERS1867187, ERS1867188, ERS1867189, ERS1867190, ERS1867191, ERS1867192, ERS1867193, ERS1867194, ERS1867195, ERS1867196, ERS1867197, ERS1867198, ERS1867199, ERS1867200, ERS1867201, ERS1867203, ERS1867207, ERS1867208, ERS1867209, ERS1867210, ERS1867211, ERS1867212, ERS1867214, ERS1867215, ERS1867216, ERS1867217, ERS1867218, ERS1867219, ERS1867220, ERS1867221, ERS1867222, ERS1867223, ERS1867224, ERS1867225, ERS1867226, ERS1867227, ERS1867228, ERS1867229, ERS1867230, ERS1867232, ERS1867233, ERS1867234, ERS1867235, ERS1867237, ERS1867238, ERS1867239, ERS1867240, ERS1867241, ERS1867242, ERS1867243, ERS1867244, ERS1867245, ERS1867246, ERS1867247, ERS1867248, ERS1867249, ERS1867250, ERS1867251, ERS1867252, ERS1867254, ERS1867256, ERS1867257, ERS1867258, ERS1867259, ERS1867260, ERS1867261, ERS1867262, ERS1867263, ERS1867264, ERS1867265, ERS1867266, ERS1867267, ERS1867269, ERS1867270, ERS1867271, ERS1867272, ERS1867273, ERS1867274, ERS1867290, ERS1867291, ERS1867304, ERS3348176, ERS3399763, ERS3399771, ERS3399792, ERS3399796, ERS3399841, ERS3399859, ERS3399879, ERS3399894, ERS3399897, ERS3399915, ERS3399941, ERS3399943, ERS3399951, ERS3399970, ERS3400005, ERS3400019, ERS3400047, ERS3400067, ERS3400090, ERS3400110, ERS3400115, ERS3400116, ERS3400121, ERS3400124, ERS3400125, ERS3400126, ERS3400142, ERS3400144, ERS3400152, ERS3400183, ERS3400196, ERS3400204, ERS3400208, ERS3400210, ERS3400212, ERS3400226, ERS3400240, ERS3400249, ERS3400250, ERS3400256, ERS3400262, ERS3400263, ERS3400271, ERS3400277, ERS3400280, ERS3400283, ERS3400296, ERS3400318, ERS3400323, ERS3400328, ERS3400330, ERS3400333, ERS3400334, ERS3400335, ERS3400337, ERS3400342, ERS3400343, ERS3400344, ERS3400346, ERS3400356, ERS3400376, ERS3400419, ERS3400430, ERS3400437, ERS3400444, ERS3400467, ERS3400483, ERS3400488, ERS3400492, ERS3400495, ERS3400520, ERS3400541, ERS3400556, ERS3400558, ERS3400593, ERS3400615, ERS3400652, ERS3400681, ERS3400731, ERS5447091, Gurgaon01, Gurgaon02, Iso21-29_05_2016, Iso22-04_06_2016, Iso23-11_06_2016, Iso24-30_06_2016, Iso25-10_07_2016, Iso27-12_07_2016, Iso32-23_10_2015, Iso33-26_10_2015, Iso34-30_10_2015, Iso35-11_11_2015, Iso36-16_11_2015, Iso37-23_11_2015, Iso39-

15_12_2015, Iso41-21_12_2015, Iso42-22_12_2015, Iso43-31_12_2015, Iso44-18_01_2016, Iso45-27_01_2016, Iso47-02_03_2016, Iso48-11_03_2016, Iso51-22_03_2016, Iso52-02_05_2016, Iso53-10_05_2016, Iso54-12_05_2016, Iso55-19_05_2016, Iso58-28_01_2016, Iso59-25_05_2016, Iso60-26_05_2016, Iso61-03_06_2016, Iso62-10_06_2016, Iso63-06_07_2016, Iso64-14_07_2016, Iso65-15_07_2016, Iso66-22_07_2016, Iso67-28_07_2016, Iso69-01_08_2016, Iso70-04_08_2016, Iso72-09_09_2016, Iso82-28_02_2016, KRSAL17-2203, KRSAL17-2204, KRSAL17-2205, KRSAL17-2206, KRSAL17-2207, KRSAL17-2216, KRSAL17-2217, KRSAL17-2218, RR051, SalBs19, SLT0096, SLT0291, SLT0469, SLT0596, SLT0626, SLT0892, SLT10, SLT1105, SLT1131, SLT11, SLT1, SLT2, SLT3, SLT4, SLT8, XDR10, XDR11, XDR13, XDR15, XDR16, XDR17, XDR18, XDR19, XDR1, XDR21, XDR22, XDR23, XDR24, XDR25, XDR26, XDR27, XDR28, XDR2, XDR30, XDR31, XDR35, XDR3, XDR4, XDR5, XDR6, XDR9,

Supplementary List S3 Enterobase IDs of EPEC genomes used in chapters 9, 10 and 11.

ESC_BA2094AA_AS, ESC_BA4381AA_AS, ESC_BA6287AA_AS, ESC_BB9042AA_AS, ESC_HA9212AA_AS, ESC_HA9215AA_AS, ESC_HA9220AA_AS, ESC_HA9222AA_AS, ESC_HA9224AA_AS, ESC_HA9227AA_AS, ESC_HA9229AA_AS, ESC_HA9233AA_AS, ESC_HA9237AA_AS, ESC_HA9238AA_AS, ESC_HA9257AA_AS, ESC_HA9264AA_AS, ESC_HA9273AA_AS, ESC_HA9286AA_AS, ESC_HB7205AA_AS, ESC_QA4139AA_AS, ESC_WB2499AA_AS, ESC_WB2500AA_AS, ESC_WB2501AA_AS, ESC_WB2502AA_AS, ESC_WB2503AA_AS, ESC_WB2504AA_AS, ESC_WB2505AA_AS, ESC_WB2506AA_AS, ESC_WB2507AA_AS, ESC_WB2508AA_AS, ESC_WB2509AA_AS, ESC_WB2510AA_AS, ESC_WB2511AA_AS, ESC_WB2512AA_AS, ESC_WB2561AA_AS, ESC_WB2563AA_AS, ESC_WB2570AA_AS, ESC_WB2653AA_AS, ESC_WB2654AA_AS, ESC_WB2655AA_AS, ESC_WB2656AA_AS, ESC_WB2664AA_AS, ESC_WB2665AA_AS, ESC_WB2666AA_AS, ESC_WB2680AA_AS, ESC_WB2681AA_AS, ESC_WB2684AA_AS, ESC_WB2685AA_AS, ESC_WB2692AA_AS, ESC_WB2693AA_AS, ESC_WB2703AA_AS, ESC_WB2704AA_AS, ESC_WB2705AA_AS, ESC_WB2707AA_AS, ESC_WB2709AA_AS, ESC_WB2710AA_AS, ESC_WB2711AA_AS

Supplementary List S4 HBVDB IDs of HBV genomes used in chapters 9, 10 and 11.

ESC_GA3262AA_AS, ESC_IA2720AA_AS, ESC_KA6005AA_AS, ESC_OA1637AA_AS, ESC_GA3268AA_AS, ESC_IA8359AA_AS, ESC_KA6008AA_AS, ESC_OA1638AA_AS, ESC_HA0247AA_AS, ESC_KA2130AA_AS, ESC_KA6010AA_AS, ESC_OA1643AA_AS, ESC_HA0256AA_AS, ESC_KA2808AA_AS, ESC_KA6014AA_AS, ESC_PA1362AA_AS, ESC_HA0258AA_AS, ESC_KA2810AA_AS, ESC_KA6016AA_AS, ESC_PA1363AA_AS, ESC_HA0259AA_AS, ESC_KA2811AA_AS, ESC_KA6018AA_AS, ESC_PA1365AA_AS, ESC_HA0261AA_AS, ESC_KA5968AA_AS, ESC_KA6019AA_AS, ESC_PA1367AA_AS, ESC_HA0262AA_AS, ESC_KA5971AA_AS, ESC_KA6023AA_AS, ESC_PA1377AA_AS, ESC_HA2835AA_AS, ESC_KA5975AA_AS, ESC_KA6025AA_AS, ESC_RA1210AA_AS, ESC_HA9215AA_AS, ESC_KA5981AA_AS, ESC_NA8556AA_AS, ESC_RA4033AA_AS, ESC_HA9222AA_AS, ESC_KA5991AA_AS, ESC_NA9304AA_AS, ESC_RA8968AA_AS, ESC_HA9224AA_AS, ESC_KA5992AA_AS, ESC_NA9306AA_AS, ESC_TA0085AA_AS, ESC_HA9229AA_AS, ESC_KA5994AA_AS, ESC_NA9307AA_AS, ESC_TA2097AA_AS, ESC_HA9237AA_AS, ESC_KA5999AA_AS, ESC_NA9310AA_AS, ESC_TA2098AA_AS, ESC_HA9238AA_AS, ESC_KA6000AA_AS, ESC_NA9311AA_AS, ESC_HA9286AA_AS, ESC_KA6004AA_AS, ESC_NA9312AA_AS

Supplementary List S5 GISAID IDs of SARS-CoV-2 genomes used in chapters 9, 10 and 11.

EPI_ISL_402119, EPI_ISL_404227, EPI_ISL_406799, EPI_ISL_408010, EPI_ISL_408669, EPI_ISL_414366, EPI_ISL_422219, EPI_ISL_402120, EPI_ISL_404228, EPI_ISL_406800, EPI_ISL_408068, EPI_ISL_408670, EPI_ISL_415655, EPI_ISL_422223, EPI_ISL_402121, EPI_ISL_404253, EPI_ISL_406801, EPI_ISL_408430, EPI_ISL_408975, EPI_ISL_416442, EPI_ISL_423308, EPI_ISL_402123, EPI_ISL_404895, EPI_ISL_406844, EPI_ISL_408431, EPI_ISL_408976, EPI_ISL_416673, EPI_ISL_423623, EPI_ISL_402124, EPI_ISL_405839, EPI_ISL_406862, EPI_ISL_408478, EPI_ISL_408977, EPI_ISL_417307, EPI_ISL_424105, EPI_ISL_402125, EPI_ISL_406030, EPI_ISL_406959, EPI_ISL_408479, EPI_ISL_408978, EPI_ISL_417317, EPI_ISL_424853, EPI_ISL_402126, EPI_ISL_406031, EPI_ISL_406960, EPI_ISL_408480, EPI_ISL_408994, EPI_ISL_417455, EPI_ISL_424898, EPI_ISL_402127, EPI_ISL_406034, EPI_ISL_406970, EPI_ISL_408481, EPI_ISL_408995, EPI_ISL_417460, EPI_ISL_425613, EPI_ISL_402128, EPI_ISL_406036, EPI_ISL_406973, EPI_ISL_408482, EPI_ISL_408996, EPI_ISL_417505, EPI_ISL_426704, EPI_ISL_402130, EPI_ISL_406531, EPI_ISL_407073, EPI_ISL_408484, EPI_ISL_408998, EPI_ISL_41773, EPI_ISL_427044, EPI_ISL_402131, EPI_ISL_406533, EPI_ISL_407079, EPI_ISL_408485, EPI_ISL_408999, EPI_ISL_417935, EPI_ISL_427142, EPI_ISL_402132, EPI_ISL_406534, EPI_ISL_407084, EPI_ISL_408486, EPI_ISL_409000, EPI_ISL_418159, EPI_ISL_427312, EPI_ISL_403928, EPI_ISL_406535, EPI_ISL_407193, EPI_ISL_408487, EPI_ISL_409001, EPI_ISL_418383, EPI_ISL_427358, EPI_ISL_403929, EPI_ISL_406536, EPI_ISL_407214, EPI_ISL_408488, EPI_ISL_409002, EPI_ISL_418798, EPI_ISL_427388, EPI_ISL_403930, EPI_ISL_406538, EPI_ISL_407215, EPI_ISL_408489, EPI_ISL_409020, EPI_ISL_419482, EPI_ISL_427643, EPI_ISL_403931, EPI_ISL_406592, EPI_ISL_407313, EPI_ISL_408511, EPI_ISL_409022, EPI_ISL_419606, EPI_ISL_428923, EPI_ISL_403932, EPI_ISL_406593, EPI_ISL_407893, EPI_ISL_408512, EPI_ISL_409023, EPI_ISL_419670, EPI_ISL_428950, EPI_ISL_403933, EPI_ISL_406594, EPI_ISL_407894, EPI_ISL_408513, EPI_ISL_409024, EPI_ISL_420169, EPI_ISL_429021, EPI_ISL_403934, EPI_ISL_406595, EPI_ISL_407896, EPI_ISL_408514, EPI_ISL_409025, EPI_ISL_420303, EPI_ISL_429219, EPI_ISL_403935, EPI_ISL_406596, EPI_ISL_407976, EPI_ISL_408515, EPI_ISL_409026, EPI_ISL_420477, EPI_ISL_429311, EPI_ISL_403936, EPI_ISL_406597, EPI_ISL_407987, EPI_ISL_408665, EPI_ISL_409027, EPI_ISL_420644, EPI_ISL_429697, EPI_ISL_403937, EPI_ISL_406716, EPI_ISL_407988, EPI_ISL_408666, EPI_ISL_409067, EPI_ISL_421662, EPI_ISL_403962, EPI_ISL_406717, EPI_ISL_408008, EPI_ISL_408667, EPI_ISL_413888, EPI_ISL_421768, EPI_ISL_403963, EPI_ISL_406798, EPI_ISL_408009, EPI_ISL_408668, EPI_ISL_414026, EPI_ISL_422118

Supplementary List S6 GISAID IDs of EPEC genomes used in chapter 11.1 to verify the annotation of the genomes.

JHQV00000000, JHQW00000000, JHQX00000000, JHQY00000000, JHQZ00000000, JHRA00000000, JHRB00000000, JHRC00000000, JHRD00000000, JHRE00000000, JHRF00000000, JHRG00000000, JHRH00000000, JHRI00000000, JHRJ00000000, JHRK00000000, JHRL00000000, JHRM00000000, JHRN00000000, JHRO00000000, JHRP00000000, JHRQ00000000, JHRR00000000, JHRS00000000, JHRT00000000, JHRU00000000, JHRV00000000, JHRW00000000, JHRX00000000, JHRY00000000, JHRZ00000000, JHSA00000000, JHSB00000000, JHSC00000000, JHSD00000000, JHSE00000000, JHSF00000000, JHSG00000000, JHSH00000000, JHSI00000000, JHSJ00000000, JHSK00000000, JHSL00000000, JHSM00000000, JHSN00000000, JHSO00000000, JHSP00000000, JHSQ00000000, JHSR00000000, JHSS00000000, JHST00000000, JHSU00000000, JHSV00000000, JHSW00000000, JHSX00000000, JHSY00000000, JHSZ00000000, JHTA00000000, JHTB00000000, JHTC00000000, JHTD00000000, JHTE00000000, JHTF00000000, JHTG00000000, JHTH00000000, JHTI00000000, JHTJ00000000, JHTK00000000, JHTL00000000, JHTM00000000