Applications of deep learning and statistical methods for a systems understanding of convergence in immune repertoires

Citation: Moghimi, Pejvak Abbas Zadeh (2023) Applications of deep learning and statistical methods for a systems understanding of convergence in immune repertoires. [Thesis] (Unpublished)

BIRKBECK, UNIVERSITY OF LONDON

DOCTORAL THESIS

Applications of deep learning and statistical methods for a systems understanding of convergence in immune repertoires.

Author:

Pejvak Abbas Zadeh Moghimi

Supervisors:

Prof. Adrian Shepherd

James Snowden

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy
in the
Shepherd Lab Biological Sciences

March 30, 2022

Declaration

This thesis describes research conducted at the Institute of Structural and Molecular Biology, Birkbeck College and University College London, between September 2017 and March 2022, under the supervision of Prof. Adrian Shepherd and James Snowden. I, Pejvak Abbas Zadeh Moghimi, declare that the research described is original and that any parts of the work that have already appeared in the publication have been suitably cited.

Word count: 34278

Signed:

Date: March 30, 2022

Abstract

Deep learning and adaptive immune receptor repertoire (AIRR) biology are two emerging fields that are highly compatible due to the inherent complexity of the immune systems and the enormous amount of data produced in AIRR-sequencing research combined with the revolutionary success of deep learning technology to make predictions about high-dimensional complex systems/data.

We took steps towards the effective utilisation of and statistical methods in repertoire immunology by undertaking one of the central problems in immunology, i.e. immune repertoire convergence. First, we took part in developing and testing an array of summary statistics for immune repertoires to gain insights into the descriptive features of immune repertoires and grant us the ability to compare repertoires.

We collected the deepest sequencing datasets to address whether the population-wide genomic convergence of immunoglobulin molecules can be predicted. The immunoglobulin molecules were labelled with their "degree of commonality" (DoC), defined as the number of times an immunoglobulin V3J clonotype is observed in a population, where a V3J clonotype is defined by its V and J genes and CDR3 sequence. We developed various bespoke data analytics methods, informed at different stages by the summary statistics we had previously implemented. Importantly, we demonstrated that machine learning (ML) predictions for immune repertoires could lead to misleadingly positive outcomes if data is processed inappropriately due to "data leakage" and addressed this issue by implementing a leak-free data processing pipeline. Here, data leakage refers to immunoglobulin sequences with the same clonotype definition spreading across the train-validation-test splits in the ML task. We designed a multitude of bespoke deep neural network architectures, implemented under various modelling approaches, including a customised squeeze-and-excitation temporal convolutional neural network (SE-TCN) and a Transformer model. Unsurprisingly, given the continuous spectrum of DoCs, regression modelling proved to be the best approach, both in the granularity of predictions and error distribution. Finally, we report that our SE-TCN architecture under the regression modelling framework achieves state-of-the-art performance by achieving an overall mean absolute error (MAE) score of 0.083 and per-DoC error distributions with reasonably small standard deviations.

Acknowledgements

I would like to thank my supervisor, Prof. Adrian Shepherd. I would also like to thank Dr Mark Williams and Dr Dave Houldershaw for their help and support throughout my PhD, and others from across Birkbeck and UCL who have played a role in my intellectual development.

List of Figures

List of Tables

List of Abbreviations

- AI - Artificial Intelligence
- AIRR - Adaptive Immune Receptor Repertoire
- ANN - Artificial Neural Network
- AUC - Area Under the Receiver Operator Characteristic Curve
- BCR - B-cell Receptor
- cDNA - Complementary DNA
- cDoC - Continualised Degree of Commonality
- CDF - Cumulative Distribution Function
- ChIP-Seq - Chromatin Immunoprecipitation-Sequencing
- CDR3 - Complementarity-Determining Region 3
- CM - Confusion Matrix
- CNN - Convolutional Neural Network
- CV - Cross-Validation
- DNN - Deep Neural Network
- DoC - Degree of Commonality
- EDA - Exploratory Data Analysis
- ECDF - Empirical Cumulative Distribution Functions
- FDC - Follicular Dendritic Cells
- FPg - Frequency Polygons
- GAN - Generative Adversarial Network
- GBT - Gradient Boosted Trees
- gDNA - Genomic DNA
- GRAVY - Grand Average of Hydropathy
- HPC - High-Performance Computing
- HT-SELEX - High-Throughput Systematic Evolution of Ligands by Exponential Enrichment
- IMGT - International ImMunoGeneTics Information System
- KDE - Kernel Density Estimation
- KL Divergence - Kullback-Leibler Divergence

- LSTM - Long Short-Term Memory
- MAE - Mean-Absolute Error
- MCC - Matthew's Correlation Coefficient
- ML - Machine Learning
- MPGD - Median of Proportional Geometric Densities
- MSE - Mean Squared Error
- NGS - Next-Generation Sequencing
- NLP - Natural Language Processing
- OAS - Observed Antibody Space
- PDF - Probability Distribution Functions
- PMF - Probability Mass Function
- RAM - Random-Access Memory
- RNN - Recurrent Neural Network
- ReLu - Rectified Linear Units
- RIIM - Relative Immunoglobulin Incidence Measure
- RDD - Resilient Distributed Dataset
- SE-TCN - Squeeze-and-Excitation Temporal Convolutional Neural Network
- SGD - Stochastic Gradient Decent
- SHM - Somatic Hypermutation
- SPGD - Summed Proportional Geometric Densities
- SVM - Support Vector Machine
- TCN - Temporal Convolutional Neural Network
- TFHC - T Follicular Helper Cells
- t-SNE - T-Distributed Stochastic Neighbor Embedding
- UMI - Unique Molecular Identifiers
- V3J - Identical V and J Gene Segments Usage and CDR3 Sequence

I would like to dedicate this thesis to Mona and Becky.
To my mum, Mona, whose motherly love and infinite
unconditional sacrifices are the reasons that I am where
I am today. Nothing would have been possible without
you, and everything I do is with you in my mind.
To Becky, for all the love and beauty you bring to my
world, and thank you for your immeasurable patience
with me throughout writing this thesis.
It is with you that these milestones
In life, have worth and meaning.

1 Introduction

1.1 Fundamentals of Immunoglobulins structure, function, and repertoires

Immunoglobulin molecules are an extremely diverse class of peptides produced by cells of the B lymphocyte lineage in vertebrate animals that respond to foreign bodies known as antigens, granting animals protection against pathogens[1]. Immunoglobulins are divided into two forms, namely a membrane-bound form alternatively known as a B-cell receptor (BCR) and a secreted form otherwise known as an antibody, the latter being produced by mature B cells, known as plasma cells[1,2]. In humans and most other animals, Immunoglobulins are composed of two chains, known as the heavy chain and light chain, each consisting of two main structural elements, namely the constant (C) and variable (V) regions[1,2].

Concerning genomic diversity and antigen-binding ability, the variable region is the region of primary importance. This region is formed by the recombination of three gene segments, i.e. the variable (V), Diversity (D) and Joining (J) segments, each of which, is found in many different copies with each copy often found in many allelic forms[1,2] (*Figure 1-1, Table 1-1*). *Whilst b*oth the heavy and light chains contain variable regions, only the heavy-chains variable region contains all three segments, while the light-chain's variable region is only formed by the V and J segments. Important to the binding ability and diversity, the variable region of each chain contains 3 hypervariable regions known as the complementarity determining regions (CDRs), with each CDR, separated from the next by a framework (fr) region (*Figure 1-1*). CDRs *als*o have the largest amount of contact with antigen molecules, with the complementarity-determining region 3 (CDR3) region eliciting the highest level of variability across the entire molecule, playing the central role in defining antibody binding and affinity out of all the regions (*Figure 1-2*).

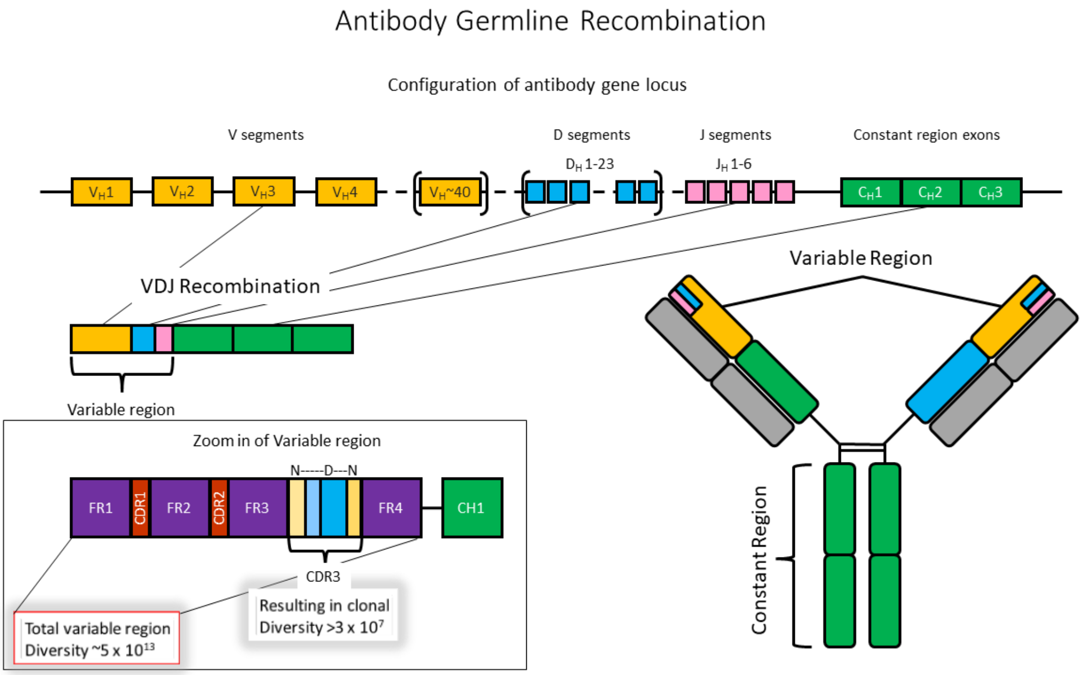


Figure 1-1 Antibody Heavy-Chain Germline Recombination. *The three gene-segment types, namely the Variability (V), Diversity (D) and Joining (J) segments, which code for the variable region of the heavy chain have many different copies. These copies are recombined in a probabilistic process, whereby a single copy of each type is selected and joined to form the variable region. The variable region can be divided into two types of regions, i.e., the complementarity determining region (CDR) and the framework region. The CDR regions are the largest contributors to variable region diversity and have the largest contact surface area with antigens. The same details are true for the light chain, except that only V and J gene segments are used for coding for the light chain variable domain, and that, there are two sets of light chains, i.e. the Kappa and Lambda chains.* ***(This figure is inspired by a figure in the textbook Janeway's Immunology***[2]***)****.*

Number of functional gene segments in human immunoglobulin loci

Segment	Light Chains		Heavy Chain
	κ	λ	H
Variable (V)	34-38	29-33	38-46
Diversity (D)	0	0	23
Joining (J)	5	4-5	6
Constant (C)	1	4-5	9

Table 1-1 The gene-segment statistics of antibody chains. *Each of the four gene-segment types which code for the variable and constant domains of the heavy chain, and the three types coding for the light chain, exist at varying levels of diversity. This results in a very large combinatorial space of possible immunoglobulin permutations.* ***(This figure is inspired by a figure in the textbook Janeway's Immunology[2], which may be outdated by the time of reading, as more allelic variants are continuously found in populations).***

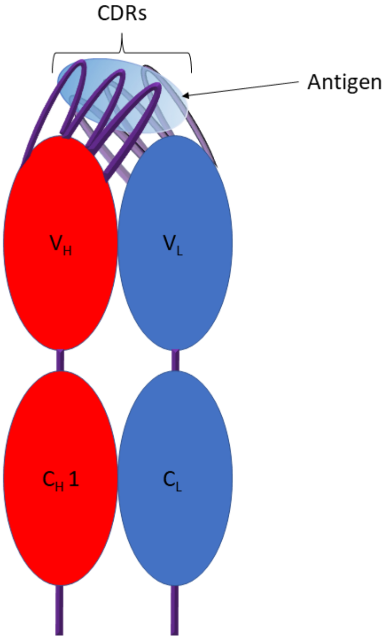


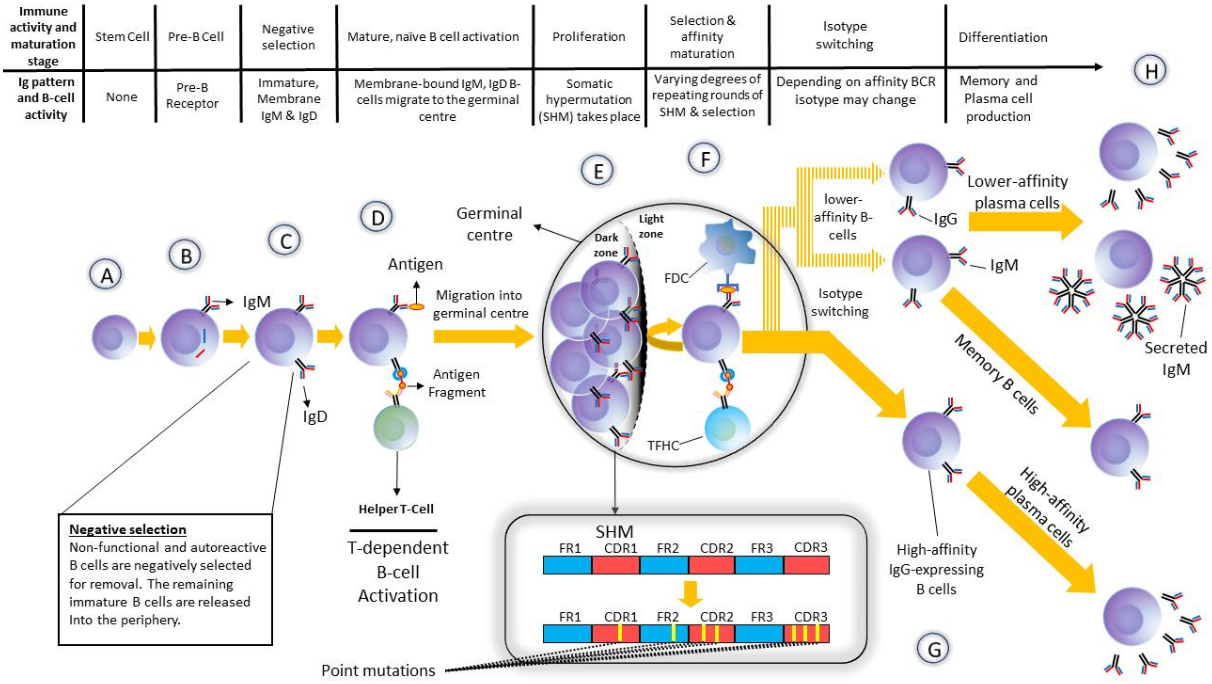
Figure 1-2 Cartoon structure of the overall antibody-antigen complex. *Broadly, an antibody is comprised of two chains where each chain consists of two domains, namely a constant domain, which is not directly involved in antigen binding, and a variable domain which is central to binding, with the CDR regions having the largest contact surface-area with the antigen.*

The B cell response represents a form of protection that evolves to meet the need of facing unseen immunogenic challenges, as well as maintain a memory of previously encountered challenges[1–4]. Pre-B cells terminally differentiate into naïve-matured B cells in the primary lymph nodes, such as bone marrow[1,2], and undergo development at the same site[1,2] (*Figure 1-3*). These *ce*lls then migrate to secondary lymph nodes via efferent lymphatic vessels, where they are either met by an antigen that binds to one, or more, BCR(s) on a B cell, which otherwise migrates into blood vessels where they may encounter antigens[1,2] (*Figure 1-3*). Befor*e t*his migration of naïve immature B cells, non-functional and/or autoreactive B cells are selected for removal[1,2] (*Figure 1-3*). B cells, which are activated through binding an antigen, form (or enter) a special zone in the lymph node called a 'germinal centre', where they undergo massive proliferation and somatic hypermutation (SHM)[1,2] (*Figure 1-3*). This *pro*cess is followed by a 'Darwinian selection' process, which ensures an evolutionary response that counteracts the evolution of pathogens and enables animals to mount an immune response to unseen antigens[1–4]. Even the selected and surviving B cells have a limited lifespan, which is affected by whether it encounters an antigen it can bind, but also its isotype[1,2]. In addition to the genomic and structural diversity of the variable region, Immunoglobulins can also diversify into many isotypes and isotype subtypes through modulations of their constant region. An isotype, typically low-affinity immunoglobulin-Ms/immunoglobulin-Ds, can be converted into another, where each isotype operates in different physiological environments with different effector functions and lifespans suited to the tissue it protects.

B cell development and maturation is a complex and complicated process[1,2]. Each BCR is composed of a homodimer, with each part composed of a heavy chain and a light chain[1,2,5]. During the development of B cells from Pre-B cells, the genomic region corresponding to encoding these chains undergoes a major recombination event which results in a single gene-segment copy, one from each of the three V, D and J heavy-chain gene segments for the heavy-chain (limited to an independent set of only V and J light-chain-segments for the light chain) to be ligated to each other, forming a template region for the transcription of the BCR gene[1,2]. These genes form the variable region of the antibody, which incorporates the antigen-binding CDR regions of the expressed immunoglobulin[1,2,5]

(*Figure 1-1 and 1-2*). This recombination event is a probabilistic process, which together with the heavy-light-chains pairing, introduces a great deal of diversity given the two-fold large combinatorial space[1–3,6]. Further to the combinatorial diversity, the recombination event results in indels and point mutations, causing an additional "junctional diversity". This potential diversity is further expanded by the process of SHM during the affinity maturation process - after a B cell is activated by an antigen and/or Helper T cells and has migrated to a germinal centre[1–3,6]. Once a naïve B cell undergoes SHM, the random mutations are primarily introduced to the CDRs (also known as hypervariable regions), though, these mutations are not exclusive to CDRs and can happen within the FR regions as well[1–3,6] (*Figure 1-3*). B cells that undergo SHM in the dark zone of the germinal centre then migrate to the light zone, where they are inspected by Follicular Dendritic cells (FDC) and T follicular helper cells (TFHC), where the B cells which bind the antigens presented by FDCs, ingest the antigen, and then present a fragment to the TFHCs[1,2] (*Figure 1-3*). The outcoming B cells with affinity to an antigen are selected to survive and re-enter the dark zone for further rounds of SHM, where this dynamic may be repeated many times as part of the affinity maturation process, while those with no affinity die (*Figure 1-3*). It should be noted that once a B cell is activated and undergoes proliferation, all progenies of a clonotype (a B cell with a unique set of V and J gene segments and CDR3 sequence) undergo this repetitive process of affinity maturation to diversify, but each to varying degrees depending on their affinity. One can appreciate how this explosive and diversifying expansion together with the previously noted combinatorial process can lead to a "B-cell repertoire".

Cells exit the affinity maturation stage at varying degrees of affinity and go on to have different fates (*Figure 1-3*). Some cells will exit this process at an earlier stage, at relatively lower affinities, to differentiate into memory B cells (*Figure 1-3*). Some cells may undergo isotype switching, e.g. from immunoglobulin-M to immunoglobulin-G, after an early exit, and then differentiate to relatively lower affinity antibody-secreting cells, known as plasma cells (*Figure 1-3*). Conversely, some cells will achieve very high levels of affinity, before exiting and isotype-switching, and then differentiate into high-affinity plasma cells (*Figure 1-3*). It is also possible for high, and relatively-lower, affinity immunoglobulin-Ms to differentiate into plasma cells (*Figure 1-3*). It is important to note that, while overall the immune system imposes a selection pressure for the creation of high-affinity tissue-specific antigen isotypes acting as targeted strikes against a particular antigen, there is also an underlying range of responses, which allows the immune system to explore the "solution space", which is a testament to the complexity and intelligent behaviour of the adaptive immune system.



Immune activity and maturation stage	Stem Cell	Pre-B Cell	Negative selection	Mature, naïve B cell activation	Proliferation	Selection & affinity maturation	Isotype switching	Differentiation
Ig pattern and B-cell activity	None	Pre-B Receptor	Immature, Membrane IgM & IgD	Membrane-bound IgM, IgD B-cells migrate to the germinal centre	Somatic hypermutation (SHM) takes place	Varying degrees of repeating rounds of SHM & selection	Depending on affinity BCR isotype may change	Memory and Plasma cell production

Negative selection
Non-functional and autoreactive B cells are negatively selected for removal. The remaining immature B cells are released Into the periphery.

Figure 1-3 B-cell developmental landscape. *Hematopoietic stem cells differentiate into Pre-B cells, which express a BCR pre-cursor**(A)**. These cells then develop into immature B cells, which can express immunoglobulin-M or immunoglobulin-D BCRs and undergo Thymic (or equivalent) selection **(B&C)**, whereby autoreactive and non-functional B cells are selected for removal. The remaining matured B cells are activated by binding an antigen to ingest the antigen and present a peptide fragment to Helper T cells, which induce the B cell to migrate into a germinal centre **(D)**. The migrated B cells start proliferating in the dark zone of the germinal centre, during which they SHM **(E)**. SHM introduces point mutations across the variable domain of B cells, with a higher concentration of mutations at the CDR sites. The mutated B cells then migrate into the light zone of the germinal centre, where they are presented with antigens by Follicular Dendritic Cells (FDC) **(F)**. If B cells are capable of binding to antigen, they ingest it and present it to T Follicular Helper cells at the site, which signal to these cells to migrate back to the dark zone for further affinity maturation, a process which can repeat many times as long as the B cells display affinity to their target antigen **(F)**. B cells can exit the affinity maturation process after varying repetitions of the process. Some cells will exit early, when their affinity remains relatively low, to differentiate into memory B cells*

***(G&H)**. Some cells will undergo isotype-switching, whereby an immunoglobulin-M or immunoglobulin-D-expressing B cell can be converted, by replacing part of their constant domain, into a B cell which can express other kinds of immunoglobulin, e.g., immunoglobulin-G **(G&H)**. The class-switched, with high or relatively low affinities, can differentiate into antibody-secreting plasma cells **(G&H)**. This is also possible for B cells, which have not undergone class-switching.*

Together, these processes result in a level of theoretical BCR diversity that is astronomically high[7–11], although published approximations of the diversity in a single repertoire range between 10^6-10^20[1,2,9]. There are a great number of biological parameters and constraints (such as the preferential use of certain germline gene segments) to consider when these calculations are made, some of which are poorly understood, which explains why estimates often differ widely[8,9]. This huge diversity is crucial to the adaptive immune system's ability to counteract pathogens, which means that at any one time, there is a large repertoire of diverse immunoglobulins circulating in the body. In addition to genomic and structural diversity, inclusive of genetic biases such as gene-segment usage and VDJ recombination profiles, there are more dimensions to diversity among immunoglobulins, such as physicochemical characteristics, such as those associated with the physicochemical properties of an immunoglobulin's constituent amino acids.

A recurring characteristic of adaptive complex biological systems is the robustness and redundancy of these systems to prevent total system failure/collapse in the event of local failures within the system, often resulting in convergent behaviour within and among systems[12,13]. Immunoglobulin repertoires are no exception to this phenomenon, and convergent behaviour can be expected to happen within a system, i.e. multiple immunoglobulins with differing genomic, structural or other characteristics responding to the same antigen, or even epitope (the local region of an antigen to which an immunoglobulin binds)[14,15]. A different kind of convergent behaviour can also be expected and indeed is likely common, where immune systems of different individuals in a population converge on a "solution" to a particular antigen by immunoglobulins that are similar in sequence and/or structure[16–18]. Moreover, given the prevalence of degeneracy in biological systems, it is also likely that immunoglobulins of differing sequences and structures could converge on the same function in terms of binding the same epitope[19]. Studying the mechanics and dynamics of these constantly changing and evolving complex systems requires a different set of approaches, such as machine learning and other statistical methods, as well as bespoke “wet-lab” techniques, which can help decipher the complexity of such systems. It is for this reason that the emerging field of immune repertoire biology, going forward, will play a crucial role in understanding both health and disease and basic immunology.

1.2 AIRR-Seq: High-Throughput Sequencing of Immune Repertoires

The advent of next-generation sequencing (NGS) has revolutionised our understanding of disease and diversity in many areas of the biological sciences[20–24]. With the significant decrease in the cost of NGS, access to this technology has become widespread [22,23,25–27], together with other emerging technologies, enabling the scientific community to increasingly adopt systems biology approaches to addressing biological problems[22,27–30]. AIRR research has benefitted from these advances, resulting in the ongoing development of the emerging field of repertoire immunology[4,31–35]. Currently, Immune repertoire sequencing, increasingly referred to as AIRR-seq, involves both bulk and single-cell sequencing of B and T cells, although single-cell sequencing has only recently taken off as a viable option for sequencing a sufficient number of samples appropriate for capturing the diversity of repertoires at reasonable depth. In the case of B cells, approximately 10 billion cells are circulating in a human individual. Species richness analyses of ultra-deep sequencing studies[16,17] recently have estimated clonotype diversity of $\sim 1\times10^7$ - $\sim 2\times10^9$ and sequence diversity of $\sim 1\times10^8$ - $\sim 2\times10^9$. Until recently, single-cell sequencing technologies were able to only sequence in the range of 100s of thousands to a few million sequences, which, given the diversity within receptor repertoires, is far from enough for a thorough systems analysis of immunoglobulin repertoires. Conversely, bulk sequencing approaches have achieved results of up to billions of sequences per study[16,17]. Nonetheless, the advantages of single-cell sequencing over bulk sequencing (discussed below) are overwhelmingly important to the AIRR field, and with the improvements being made to its methodology the sample sizes should soon reach practical levels.

So far, the primary sources of library preparation in AIRR-seq studies have been complementary DNA (cDNA; reverse transcribed mRNA) and genomic DNA (gDNA)[36,37], due to the advantages of DNA sequencing over RNA-seq (i.e. direct sequencing of mRNA). However, RNA-seq is not the most practical method, due to its inability of capturing lower-abundance CDR3 sequences[36], it has also been used relatively infrequently[36,38]. The decision over the usage of cDNA or gDNA is based on the objectives of the research, and both have their advantages. For instance, gDNA is more stable than mRNAs and as a result easier to acquire and maintain[35], does not require reverse transcription and is more reflective of the number of cells. However, sequencing gDNA requires a relatively higher concentration of sequencing templates and is more likely to result in primer-annealing than cDNA. On the other hand, mRNAs are found in much greater abundance in every cell, resulting in reduced interference by the non-coding loci in sequencing, and providing the full-length CDR3 region more readily. However, mRNAs need to be reverse transcribed to cDNA, which could result in error[39], and due to RNA-instability imposes significantly greater overall costs compared to gDNA sequencing.

Another important issue for AIRR-seq is the choice of amplification method used for library preparation, where the choice is between multiplexed-PCR and 5’RACE-PCR. While multiplexed-PCR can be used for both gDNA and cDNA amplifications, it can produce amplification bias due to varying cross-reactivity and efficiency across different primers[36,40]. Though 5’RACE-PCR minimises this particular bias in primers, it is limited to mRNA amplification and can be biased towards shorter sequences[40]. However, the use of unique molecular identifiers (UMI) has reduced the amplification biases and allowed more accurate estimations of relative clonotype abundance. Additionally, UMIs allow backtracking of sequencing information to the cells of origin, which can be particularly useful for single-cell sequencing and acquiring sequencing data which preserves the heavy and light chains pairing information[35,41–44].

In addition to amplification biases, there are also sequencing errors to consider. Illumina sequencing offers multiple advantages over other sequencing platforms, one of which, is a much greater sequencing depth enabled by the much greater number of shorter reads. However, this strength results in a disadvantage by making the Illumina platform prone to a higher number of mismatched sequences, which is also inherently susceptible to substitution errors. Nonetheless, all other platforms have pitfalls, but, currently not as many strengths. Clustering algorithms, together with UMIs, allow the correction of sequencing errors[22,36]. Ultimately, despite all the strengths of increasingly common ultra-deep bulk sequencing, the crucial heavy and light chains' pairing information, crucial to more precise analyses of repertoires in many different respects, e.g., studying function, is lost. Naturally, the way forward will be further development of single-cell methods to achieve higher throughput. Technologies such as the Oxford Nanopore sequencing platform could also be quite exciting going forward, as not only do they allow sequencing extremely long reads of single cells on the fly[45], but they can also be integrated with display technologies, which enables us to combine sequencing with antigen-binding information[46].

1.3 Systems immunology of BCR repertoire

Systems and computational analysis of receptor repertoires start from a variety of datasets including the immunogenomic dataset acquired from AIRR-seq, which is the focus here. Computational analysis of immunogenomic features of the receptor repertoire typically starts from the post-processing phase of AIRR-seq data[47–49]. This includes processing the AIRR-seq data and creating FASTA files of individual BCR chains generated using the sequencing protocols[47–49].

This is followed by the annotation of individual sequences based on a set of pre-determined germline gene-segments reference sets compiled and stored by the international ImMunoGeneTics information system (IMGT) as the community standard[11,50,51]. Following the gene-segments annotation a series of other annotations, crucial to repertoire analysis, become possible, such as the constant, framework and CDR3 regions of the sequences and their boundaries[50,52]. An exhaustive, yet actively maturing, schema of all annotations, agreed upon by the community, is standardised and compiled by the AIRR community[53].

These annotations are used in further downstream bioinformatics analysis, such as determining the pairwise distance between sequences, the distribution of CDR3 lengths (spectratyping), gene usage frequency, hydrophobicity and Grand Average of Hydropathy (GRAVY) distribution [50,52,54]. Where multiple samples from a single individual at different timepoints are available, the temporal dynamics of clonal evolution and isotype switching may be investigated.

The results of these analyses, and the sequences themselves, can be used in a number of different ways for systemically analysing the repertoire. Generally, systems immunogenomic analysis of receptor repertoires can be divided into four different categories: repertoire diversity, repertoire similarity network architecture, evolutionary analysis and population-level convergence of repertoires[7,18,55]. Correct annotation of the repertoire allows one to investigate the diversity through recombination statistics, such as gene-usage frequency and clonotype diversity.

1.4 Network analysis

The architecture of the repertoire can be determined through network biology[56]. The pairwise distances between the sequences in a repertoire, using Levenshtein distance, can be used to build a similarity network, given a threshold of similarity[56–59]. These thresholds then can be used to deconvolute the repertoire into distinct layers of Levenshtein distance similarity, which in turn allows us to access a layer of information, useful for comparative analysis of repertoire, otherwise hidden without the use of network biology[55–60]. One way to characterise a repertoire is through the diameter of such network, e.g. how diverse the repertoire is in terms of the number of distant clones, or through the degree of connectivity in local regions of the repertoire[55,56]. The latter helps to evaluate the degree of convergence among separate clones, i.e. the higher the degree of connectivity, the smaller the span of the repertoire, in the sequence space, and likely, the narrower range of specificity to the variety of antigens. We can use network analysis of such distance metrics across and within repertoires to elucidate the degree of convergence between the clones within a repertoire and between the repertoires[55,56].

Miho et al showed that naïve repertoires display a rather even distribution in the degree of connectivity, whereas challenged repertoires follow a power law, in terms of the degree distribution[56]. Some other local network features that can be used to summarise the topology of networks at a clonal level include PageRank, authority, closeness and betweenness[56]. Similarly, there are other ways for making global evaluations among repertoires, besides the degree of connectivity through similarity, such as clustering coefficient, diameter and assortaitivity[56].

There is multiple software for network visualisation and analysis, such as igraph[61], Gephi[62], Networkx[63] and cytoscape[64]. However, given the growing sample size of the BCR data in the rep-seq studies, the visualisation is exceedingly uninformative, and we need to look into more quantifiable ways to analyse network data in repertoire studies. The coefficients described above represent an approach for such network analysis. Though, it remains to be seen if we can compare repertoire networks without reducing the data into low-resolution metrics and rather, carry out holistic evaluation. Furthermore, is it possible to frame such data-driven comparative analysis into a mathematical framework?

1.5 Machine Learning Applications

One of the aims of machine learning approaches in BCR repertoire analysis is identifying convergence across repertoires, at the clonotype and sequence motif level. One such research would aim to investigate the convergence as a result of an immunogenic challenge and convergence that exists intrinsically[18,55]. For the analysis of the clonotype convergence, pairwise distance metrics, with statistical frameworks which would take normalisation of the clonal convergence with respect to the repertoire sizes, would suffice[55]. However, given the complex, and consequently high-dimensional, nature of the BCR repertoire sequence space, investigating the sequence-level convergence by distance-based methods is insufficient. This is where machine learning (ML) algorithms will be very powerful at identifying sequence signatures associated with the convergence of repertoires of the same class, i.e. sequences exposed to the same challenge and sharing binding motifs, while possibly having large molecular distances.

There have been few reported uses of ML in such analysis, for instance, it was shown that the TCRβ repertoire of mice immunised with ovalbumin could be discerned from one of the mice immunised without it by ~80% accuracy[60]. In another study, it was shown that individuals share a set of sequences with shared motifs that can be thought of as a public repertoire, and furthermore, most of the BCR sequences in different individuals are private and share no common sequence signature[18]. This study used k-mer decomposition to identify such sequence associations, however, this has been expanded by the addition of physiochemical information (or other metrics such as the ones previously discussed above) of the sequences to complement such search[18].

Some of the other notable uses of machine learning in BCR repertoire research concern the identification of the disease-related status of repertoire. *Greiff et al* showed, by ~80% accuracy, that even by using lower dimensional, sequence-independent data, such as clonal frequency, one could differentiate between healthy and diseased repertoires[7]. They used hierarchical clustering and support vector machine (SVM) to discriminate between repertoires of different states by partitioning through the degree of sharing of what one could describe as sub-repertoire. Somewhat similarly, *Yokota et al* showed that by creating a dissimilarity profile and dividing a repertoire into sub-repertoires, using T-Distributed Stochastic Neighbor Embedding (t-SNE), which is a popular method of dimensionality reduction in immunoinformatics, as well hierarchical clustering, one could identify the inter-sample hierarchical structure and the most contributing sequences (or motifs) in this hierarchical structure[65].

Another important question in this field is the prediction of antigen-specificity based on the sequence of a BCR. The idea is to identify sequence signatures that can be associated with specific antigens, or a biased sequence similarity among BCR sequences, which would bind the same antigen, as opposed to sequences that do not. Some methods have been developed to address this problem, however, these methods are either based on clustering of similar sequences (either by shared motifs or global similarity) or by taking into account various characteristics of sequences[66,67].

To the best of my knowledge, a fully ML-based approach, that could potentially use repertoire-characterising-metrics, is yet to be developed. One outstanding problem to solve in this field arises from the lack of sufficient data. Current powerful ML techniques require large amounts of data for training (in order to generalise well across different datasets), and the issue is not just quantitative, but also qualitative, in that, one needs to account for the diversity of samples and sources of data. In the case of immunogenomic data analysis, for instance, one needs to take into account the background of the sample donors from different aspects, e.g. ethnicity, age, pathology, life history, etc. It is important to generalise observations through ML by using, not only large but diverse datasets. Though, these decisions are very important to be made on a case-by-case basis and intelligently, as this probably poses to be one of the most important aspects of data /ML research applied to biology.

The difficulty in acquiring patient data, has led to concentrated effort on simulation of repertoire data and development of tools, such as Partis[52]. Simulation of data requires a great deal of understanding of the data and its complexities, and as such, one can see the huge importance of ML methods in allowing the generalisation of repertoires by high and low dimensional features. In a later chapter I will specifically discuss the use of deep neural networks (DNN) in biological sciences and how we can apply this powerful method to immunogenomics.

1.6 Deep Learning

Perhaps the first thing to address when discussing artificial intelligence (AI) and deep learning is the matter of terminology. This issue partially arises from the popular attention to application of machine learning. The term AI is often used when discussing machines that mimic human intelligence, but, omits the type of algorithm involved in carrying out such operations. The recent revolution in AI, however, is owed particularly to the advances in the field of deep learning. Though, the term deep learning poses its own issues. It is another umbrella term, for a type of ML, that describes the focus on creating deep architectures of plethora of artificial neural network (ANN) algorithms, which mostly, have existed for a long time. Due to various advances, both algorithmic and in hardware development, these ANNs are created in exceedingly deeper architectures, and hence the term deep learning[68–73]. The theory behind AI is over 6 decades old, and ANNs have been applied to data analysis for nearly as long[74]. These recent advances are the result of number of things discussed below. Furthermore, I will give brief overview of important components in deep learning.

The simplest and first type of an ANN is a single layer feed forward perceptron, where the single layer is just the output layer (input layers are typically ignored when counting number of layers)[72,75–77]. This type of ANN is simply a linear prediction function for binary classification, where the information from the input only moves in one direction towards the output through a set of weights and biases[72,75–77]. In supervised learning, these weights can be adjusted through various learning methods, over many epochs of training the network, which calculates the error of the network based on some divergence metric between the predicted class output and the class label of the input[68,70,78–82]. In contrast, unsupervised learning approximates a function through identification of features from unlabelled data[31,70,78,83–87]. There are various approaches for unsupervised learning, which is beyond the scope of this review, however, some of these methods will be briefly discussed when applied to a biological problem.

Failure of single layer perceptron in learning linearly inseparable patterns sparked increasing the number of layers, which also enabled these deeper networks to classify more than two categories[88]. However, several issues stagnated the use of deep networks. Perhaps, one of the tightest bottlenecks was the lack of sufficient computing power[80]. Even until recently, despite the parallelisation of the neural computation over many CPU cores, the speed bottleneck was a significant issue. The concentrated efforts in writing numerical analysis and deep learning programming libraries, which utilised parallelisation of neural computation over GPU cores resulted in significant speed up in deep learning[80,89]. Another significant phenomenon was the entering the age of “big data”. The premise of machine learning is learning of patterns through many examples over many iterations, and the availability of huge amount of data, that is available to us today, plays a significant part in deep learning[72,75,80]. Obviously, biosciences have been at the forefront of data generation in this age[68,69,71,90].

Beside the importance of hardware and the big data for deep learning, there have been significant algorithms and architectures of networks that have played an important role in the recent machine learning revolution.

Activation function is a mathematical function that draws a threshold for activation of a neuron, meaning that if the value of Y (see *Equation 1-1*) is below the threshold, the neuron will be inactivated, or remain inactive[78]. Initially step activation functions were used for binary classification, but non-linear functions, such as sigmoid and Tanh became popular for multi-classification problems. Most recently, rectified linear units (ReLu), which despite the name is not a linear function, have become popular[91,92]. This is due to the sparsity of neural activation when ReLu is used as oppose to Sigmoid and Tanh functions. This dramatically reduces the computational burden of having too many active neurons. Furthermore, ReLu is less computationally expensive to calculate when compared to Tanh and Sigmoid functions. However, one problem is the learning gradient of zero that results from using this function, which is compensated by using different permutations of ReLu, such as leaky ReLu, which has a non-zero gradient[91,92].

$$Y = \sum (weight * input) + bias \tag{1-1}$$

Learning algorithm calculates the cost function, which is the networks overall output error[09,122–124]. The gradient of this cost function is calculated by **backpropagation**, which results in the slight changes in the weights of neurons[69,70,80,81,83,93–98]. Some of the learning algorithms include gradient decent, stochastic gradient descent (SGD), Adam and etc. Gradient decent was very effective for a long time, but with the increase in the size of data it is not computationally feasible to fit all the training data in memory for calculation of the total gradient at once. Recently, the competition has been between using dynamic learning algorithms, such as Adam, and mini-batch SGD. mini-batch

SGD divides the dataset into multiple mini batches and then calculates the gradient separately for each batch[99–101]. The independent calculation of gradient for each batch introduces a stochasticity that helps with scaping local minima[99–101], and therefore overcoming a common problem with optimisation algorithms. **Vanishing gradient** is a phenomenon arising in the learning process in deep networks, and is exacerbated by the increase in the depth of a network, where the learning gradient becomes smaller the deeper the network gets[69,86,102].

Convolutional neural network (CNN) is particular type of ANN used for motif detection, and naturally heavily used in computer vision[72,80,103–106]. The architecture of CNNs is composed of convolutional layers, activation layers, pooling layers, followed by a typical multilayer fully connected network and the output layer[72,80,103–106]. The convolutional layer is comprised of multiple convolutional filters (kernels), which scanning over the input data and carrying out matrix multiplication between the kernel values and the local values in the data[72,80,103–106]. The number of channels in the data is conserved, however, by this operation, the dimensionality of the data is increased[72,80,103–106]. The pooling units typically maintain the maximum convolution maps or average these maps, resulting in dimensionality reduction, whereby the fully connected network usually receives a one dimensional input[72,80,103–106]. CNN, and its applications, are discussed in greater detail below.

Recurrent neural network (RNN) is another type of widely used deep learning algorithm, which is powerful in analysis of data concerning time or position[107–113]. These networks function by learning signal with stationary features over time. Generally, RNNs are limited in depth, due to vanishing gradient problem, but also, slow training due to the sequential nature of their computation[107–113]. However, a particular kind of RNN called long short-term memory (LSTM) network overcomes the vanishing gradient problem by introducing additional gating components to the typical RNN gating mechanism[110,111,114–116]. This alternative gating mechanism effectively converts an ordinary RNN into a memory unit able to decide what, and when, to remember or forget[110,111,114–116]. This essentially turns an LSTM unit into a mini neural network, in its own right, where each unit is typically composed of four stacked up layers[110,111,114–116]. Furthermore, recent implementations of LSTMs and RNNs allows parallelisation, though, limited to a particular configuration[117].

Generative adversarial network (GAN) is a state of the art deep learning algorithm with underlying game theoretical dynamics. In this approach, two ANNs (each could be any kind of network, e.g. CNN and RNN, or hybrid) take part in an adversarial relationship, where one acts as the discriminator and the other as generator[86,96,118]. The ground truth data is randomised into noise and fed into the generator, which is supposed to transform the noise into data resembling the real data[86,96,118]. The discriminator is fed with the intact training dataset, as well the generated data, and is supposed to differentiate between the real data and the simulated data[86,96,118]. The cost function calculated by the discriminator is backpropagated in both the discriminator and the generator networks, which means that both networks improve overtime, like training any other ANN, at approximating a function [86,96,118]. Eventually, the generator achieves such high accuracy in mimicking the real dataset that the discriminator cannot possibly tell the difference between what is real and what is not[86,96,118]. This eventuality is guaranteed by the game theoretical framework that governs this dynamic[86,96,118]. An important emergent property of such learning mechanism is that for generator to simulate the real data in an undistinguishable way from reality, or the discriminator to distinguish simulated data, both networks must accumulate deep understanding of the complexity in the data. In other words, these networks are very powerful at extracting features from complex dataset in order to carry out their functions.

1.7 Natural Language Processing

Important to the application of deep learning to genomics data are the revolutionary deep learning advances that have been made in the field of natural language processing (NLP), which is the application of computational (and particularly machine learning) techniques to problems in the field of linguistics[119]. NLP field itself has several large subfields, which are active areas of research and growing very rapidly, especially in the past 10 years[117,120]. Some of the most heavily researched of these subfields include text classification, neural machine translation, language modelling, speech recognition, and neural language understanding and generation[117,120]. In all these subfields, the machine learning model receives sequential input data, which often exhibit complex relationships with long-distance dependencies among its textual building blocks. Suppose a machine learning model, which attempts to comprehend and summarise the writing in this thesis. Such model must be able to overcome many challenges, beyond the writing and communication style of a scientist! One such ability is to identify, and ultimately make connections among parts of the text, which may be disjoint in sequence (perhaps due to the narrative), but conceptually and/or semantically related with important consequences to the holistic understanding of the text when identified as such. All of these properties closely resemble those of genomics sequence data, which often exhibit non-linear relationships that are not limited to the local relationships within and between genomic loci, nor even the larger context of a whole chromatin complex, with all the potential regulatory functions associated with chromatin folding. The complex relationship within genomics sequences imposes further complex relationships within downstream products, i.e. how the expressed polypeptide folds into a particular 3D conformation, how it interacts with other molecules, and so on. If anything, one could argue that techniques used in NLP are likely to be insufficient for "solving" genomics tasks, and will likely need to go further in term of sophistication and/or complexity in order to achieve similar levels of success. Indeed, it seems reasonable to assume that a simple vertical expansions of model capacity (i.e. model parameters and/or complexity) will be insufficient on its own, although such a trend has, so far, been successful in NLP. This point has been shown to be true with the design of Alphafold 2, which beside utilising state-of-the-art NLP architecture and techniques, integrates several bespoke engineering features into the model crucial to handling of the intrinsic complexity of the task[121–123].

A key advance in tackling NLP tasks was made by Mikolov *et al*[124], who developed a technique for learning *n*-dimensional numerical vector representations of words from the input data, an approach commonly known as word embeddings. These vector representations effectively learn the geometric relationships (typically Euclidean) among the words' semantics with respect to the chosen objective function, which is commonly the prediction of a word given the words that surround it, or vice versa[113,124]. Such pre-trained embeddings can then be used in a range of downstream NLP tasks, such as text classification, where a general understanding of language semantics is beneficial. The initial embedding technique, known as Word2Vec, has since been improved and extended by many different techniques, such as fastText developed by Facebook, which (as well as the words themselves) incorporates sub-words (such as prefixes and suffixes) into the learning process[125,126].

With the return of neural networks into popular use, NLP scientists shifted towards utilising various neural network architecture in their work. Until recently, RNNs, and in particular the LSTM variant, arguably remained the most effective and popular architecture for tacking sequential data, including language tasks[110,111,113,115]. since their invention, embeddings have been used as a fundamental building block of neural network models, where they are used as the first layer of a model, i.e. the one that directly receives the inputs[127,128]. One such case is the usage of embeddings in LSTM-based neural network models with the ability to capture semantically relevant long-distance features[128].

A strong competitor to these models has been the temporal convolutional neural network (TCN) architecture, where a special variation of convolution operation is used[129,130]. This variation uses a dilation parameter, which defines the number of features to skip between every convolved feature at every stride of the convolution kernel. This allows convolutional operations to go beyond learning only local relationships in data, and as more of such layers are stacked up, without using a pooling operation in between layers, the span of long-distance features captured in a convolution operation quickly increases to large numbers. The advantages of using TCNs over RNNs are generally two-fold and relate to aspect of RNNs that are often problematic: the vanishing gradient problem, and challenge of implementing efficient computational parallelism[130].

Despite the great accomplishments of deep neural networks in NLP since early 2010s, the invention of attention mechanism and Transformer architecture nevertheless proved revolutionary[131]. The attention mechanism was born out of the neural machine translation subfield, where the encoder module, which encodes the input from the query language, is connected by an attention mechanism to the decoder module, which decodes the latent representation of the encoder into a translation of the query[117]. This kind of attention mechanism simply enables the model to learn what parts of the input to focus on in an autoregressive manner. Whilst these kinds of encoder-decoder models were typically constructed from CNN or RNN building blocks, the transformer architecture took the utilisation of the attention mechanism to a different level, by only using the attention blocks and entirely dispensing with convolution and recurrent operations[132]. The attention mechanism was later extended to the self-attention mechanism, where the attention is calculated between the features of the input data[133,134]. This was the next step in the evolution of the transformer architecture, which extended its applications to tasks such as text classification where (unlike machine neural translation) a target sequence does not exist[133]. This was particularly important for language model pre-training approaches in NLP research, where the embedding layer would not only learn the semantic relationships as usual, but also, the occurrence patterns of words[120,133]. Finally, the embeddings used in transformer architectures were extended to learn positional information about input features [134]. Together these innovations extend the ability of Transformers in learning non-linear semantic and contextual information from the data to unprecedented levels, deserving of the adjective revolutionary.

The application of deep learning in omics biosciences ranges from medical imaging to -omics data. However, there has been a disproportionate usage of deep learning in image-based biomedical data, for reasons related to the fact that data processing is more native in the image format compared to genomics data. Furthermore, an image represents the entirety of an instance of reality, whereas biology is multifaceted and complex and is not feasible to “fully” understand only through one kind of data. For these reasons, I will first outline the use-cases of deep learning in imaging-related studies**.**

Application of deep learning to omics data concerns with acquiring features from the raw data, e.g. sequence, transcriptomics, proteomics data and etc. This process follows a standard pipeline of acquiring the data, cleaning the data and using deep models to extract features and make predictions[104]. In other ML methods the feature extraction step is carried out by an expert with expert knowledge of the domain; this is perhaps one of the most influential differences between deep ML models and the other ML methods, and the likely the most relevant cause of the relative success of DNNs. In genomics, the majority of the work done so far could be classed into three main categories of (through the use DNA and RNA sequence data) predicting function, structure and phenotype[104].

Given that these predictions rely on motifs in data, CNNs have played an important role in genomics. *Alipanahi et al*, creates DeepBind for prediction of DNA and RNA binding proteins, based on the DNA and RNA sequence motifs[135]. DeepBind is composed of convolutional layers, with convolution filters of different sizes, which scan the sequence to find motifs of different sizes[135]. These convolutional layers are followed by activation function layers, ReLu in this case, followed by max and average pooling layers[135]. This model uses experimentally determined binding scores for each sequence, where the score could be a binary value or continuous-value measurements[135]. The data is acquired from protein binding microarrays, RNAcomepete assays, chromatin immunoprecipitation-sequencing (ChIP-Seq) and high-throughput systematic evolution of ligands by exponential enrichment (HT-SELEX)[135]. The result of this work is important for precision medicine, prediction of gene regulation and detection of binding sites. The success of this model stems from convolutional filters, where each identifies a feature of DNA/RNA-protein binding, drop-out regularisation and parallelisation of this model on GPUs.

In a later work, *H. R. Hassanzadeh and M. D. Wang*, developed DeeperBind as an extension of DeepBind algorithm[102]. This model uses the CNN used in DeepBind, however, adds a RNN layer to address the shortcoming of DeepBind where weaker binding motifs are concerned[102]. Weakly-binding motifs can result in high overall affinity of a sequence to a protein. However, since DeepBind ignores the positional dynamics, due to exclusive of use of a CNN model, loci with numerous, but, weaker binding motifs are not identified[102]. Furthermore, even single motifs can have differential binding affinity depending on the position on a sequence, and therefore could often have strong enough binding affinity if present at certain regions of the sequence[102], again, missed by deepbind. These faults are rectified by the use of recurrent neural layers in DeeperBind, where the recurrent layers (composed of LSTM units) take the relative position of long term (motifs) and short term (nucleotides) signals into consideration[102]. This hybrid model resulted in higher accuracy in prediction of protein binding based on target sequence, with only 1% false positive rate[102].

Moving beyond the more traditional deep learning algorithms, in the past ten years, there has been some interesting development in algorithmic design, both in supervised and unsupervised approaches, that have also found their way into biosciences domain[68]. One recent and popular approach is the use of various kinds of autoencoders, for unsupervised learning[136–138]. Though, autoencoders have been discussed since 1980s, various permutations of this neural network have been developed in the past ten years[139]. One defining characteristic of these new architectures is the alternative to backpropagation, since each layer is usually pretrained one layer at the time[139]. This type of neural network is primarily concerned with dimensionality reduction where the "encoder" (hidden layers) maps the input into smaller dimensions and copies this input onto its output[138,139]. By using this method different studies have attempted to understand, and classify, underlying data structure to achieve an unsupervised understanding of complex biological systems[137,140,141].

To predict non-coding RNA binding proteins, *Yi et al* developed RPI-SAN, a deep stacked autoencoder network that achieved 99.33% prediction accuracy and outcompeted all the other available frameworks by at least 3% higher accuracy[142]. This method is very different from the ways we have discussed, so far, in application of ANNs to genomic data. First, they start by creating a k-mer sparse matrix from the RNA sequences. This "hand-engineered" way of encoding features is a typical way of operating in other types of machine learning[18], and largely responsible for lower performance standards compared to deep ANNs. However, this high accuracy is an indicator of the power of deep autoencoder algorithm used here. Perhaps, a hybrid architecture of the stacked autoencoder used here, with a CNN network (which would carry out the motif detection, i.e. k-mer extraction, automatically) could result in even greater results. We have previously seen the power of hybrid architectures in the case of DeepBind and DeeperBind[102,135].

Another innovative deep learning method that was developed by *Goodfellow et al* in 2014 is GAN[118]. The adversarial interaction between discriminator and generator networks enables these networks to acquire a deep understanding of the underlying structure, and features, of data[86,118]. *Ghahramani et al* used a GAN to simulate the single cell gene expression patterns of distinct subpopulations of epidermal stem cells[143]. While being distinct, these subpopulations have the potency to differentiate to any of the different epidermal cell types. In an event of perturbation, however, these cells, ordinarily, give rise to specific cell types[143]. This interesting dynamic is an indicator of an underlying gene regulatory/interactive network that, if decoded, could shone some light on how the behaviour of these cell lines change in different circumstances[143]. They trained the GAN network for 15000 epochs and were able to simulate cell lines with distinct gene expressions[143]. The underlying gene expression features learnt by the network, to allow simulating distinct cell lines, was uncovered by doing t-SNE analysis on the first hidden layer of the network. When they used these features as a discriminator of different cell lines, they successfully divided cells based on their gene expression data[143].

The revolutionary results of the Transformer architecture has motivated a sudden and significant interest in their application to biological challenges, particularly in areas of research where genomics data is used[144,145]. Transformers have proved highly effective, as seen with the performance of Alphafold-2 in protein folding and structure prediction, which has arguably established itself as a pinnacle of deep neural network achievement not only in biology, but in all of science[121–123]. Immune repertoire biology appears a natural area of biology in which to apply such approaches, given the very large quantities of data that could be utilised for transformer pre-training, the exceptional levels of diversity within that data, and the apparent intractability of key tasks using conventional approaches. In fact, several groups have already adopted Transformers in their research, achieving state-of-the-art results in BCR/TCR repertoire analyses[146–148]. For instance, two separate groups have created language-like-models of immunoglobulins and TCRs, trained on 10s of millions of sequences, which have the ability to distibguish sequenes within repertoires in several ways, e.g., gene-segment usage, without any prior information about any of these features[149,150].

1.8 Conclusion

Given the efforts made recently in AIRR-seq community to bring the field into 'big data' era, it is time to invent new approaches to study this important aspect of the immune system. For example, in a recent effort, *Christley et al* created the VDJserver to provide a cloud-based portal for both computational resources for rep-seq analysis, as well as a data repository for rep-seq data[47]. This platform provides the ability carryout the processing of rep-seq data and also provides processed datasets[47]. This is only one of the frontiers of such efforts, and undoubtedly more such resources should become available for streamlining research in this field.

The available high-dimensional sequence data can be used to answer various questions in this open field. One question concerns with prediction of heavy and light chains binding based on sequence data. Whether there is a pattern in the sequence of these chains that determine the binding can be explored by the use of a hybrid deep network similar to the one used in Deeperbind, where the sequence motifs are identified by the convolutional network and the positional variance is taken into account by the recurrent neural network. Another interesting question, previously addressed by *Greiff*[18] *et al* (with a reasonable degree of accuracy, but still, with room for major improvements), is to identify sequence architecture that is shared across individuals, i.e. public repertoire, and sequence architectures unique to each individual (likely due to stochasticity and genomic differences), i.e. private repertoire.

Another way to explore such a complex system is through a holistic understanding of the system by simulation. I have briefly discussed this approach in previous studies through more traditional methods, such as in Partis, but a deep GAN model could potentially hold a better promise. This, as discussed above, was achieved by *Ghahramani et al*, and similarly can be applied to antibody repertoire data, and similarly decode underlying features that give rise to the complexity of a repertoire. For instance, one could encode metrics that characterise a repertoire (discussed above) in with the sequence data and simulate a repertoire using data. Consequently, one could use dimensionality reduction techniques, such as t-SNE, on the outputs of the first hidden layers to differentiate between repertoires at different timepoints from the time of challenge.

Having discussed the power of deep machine learning methods, one also needs to be aware of the issues. Generally, we do not have a theoretical understanding of how or why deep networks work so well. Furthermore, it was recently shown by *Recht et al* that Cifar-10 deep models overfit to the test sets, and drop in accuracy by 4-10%, even when tested on different, but similar, test sets[151]. However, they have also shown that more recent deeper, and more complex, models are more robust to this effect.

2 Sumrep

Author's declaration: My contributions to the publication[152] related to this chapter included providing some of the code for Sumrep and a significant amount of the testing of Sumrep on real datasets.

2.1 Introduction

The advances in the NGS in the recent years has made it possible to sequence the immune receptor repertoires at an unprecedented depth enabling the scientific community to gain a better systems understanding of the adaptive immune system. However, due to the complexity of this system it is not straightforward to quantify or gain insight from and to compare these datasets. Without further processing, repertoires are simply a list of DNA sequences. After genetic annotation, and some further processing steps, such as clustering of sequences to clonal families, some of the typical repertoire analyses involve comparison of several data profiles, e.g., gene usage frequencies [153–156] and CDR3 sequences among few other statistics. Comparison and evaluation of a repertoire of CDR3 sequences alone can be a very expensive task, and therefore it is common to simply compare CDR3 length distribution of a repertoire[157,158], leaving the full richness of CDR3 sequence unanalysed, as well as other interesting aspects of the germline-encoded regions.

As an alternative strategy one could transform a repertoire to a more convenient space and compare the transformed quantities according to some metric. For example, several studies reduce a set of nucleotide sequences to k-mer distributions for classification of immunization status or disease exposure[81,159,160]. These k-mer distributions can then be compared via a string metric, but still comprise a large space and lose important positional information. One can perform other dimension reduction techniques like t-SNE to project repertoires down to an even smaller space[65], but these projections lose a lot of information and will have questionable immunological meaning.

Sumrep[152] is an R package that facilitates the use of biologically interpretable summary statistics to capture many different aspects of AIRR-seq repertoires. In addition to enabling comparison of different sequencing data sets, summary statistics can also be used to compare such data sets to probabilistic models. Specifically, one can use a form of model checking that is common in statistics: after fitting a model to data, one assesses the similarity of the data generated by the model to the real data. In the present context, a sequence repertoire is generated using a model and subsequently compared to a real repertoire using summary statistics.

Prior to the publication of sumrep, there were no unified packages dedicated to the task of calculating and comparing summary statistics for AIRR-seq data sets. While the Immcantation pipeline[161–163] (including the alakazam and shazam packages) contains many summary functions for AIRR-seq data, it does not have general functionality for retrieving, comparing, and plotting these summaries. Many summaries of interest are implemented in separate packages, but differences in functionality and data structures make it troublesome to compute and compare summaries across packages. Some potentially interesting summaries, such as the distribution of positional distances between mutations, had not been previously implemented.

Development of Sumrep was led by *Frederick Matsen's* lab at the Fred Hutchinson Cancer Research Center, Seattle. My contribution to this project, aside from the coding/debugging aspects, has been through identifying appropriate longitudinal studies, and analysing repertoire data from such studies for various purposes. Namely, I have attempted to identify metrics, which best highlight changes in repertoires post-challenge, differences among the results processed by different frameworks and metrics which may be useful in the evaluation of repertoire simulators, such as Partis (see *figures 2-1* and *2-2*).

2.2 Methods

2.2.1 Datasets

Datasets from the following three studies were downloaded and analysed using sumrep:

- *Gupta et al*[164] (the Gupta dataset) downloaded from Zenodo (via the URL https://zenodo.org/record/802384#.XSxOgpNKjOQ). This dataset contains repertoires from three healthy individuals at multiple time points (days 0, 7 and 28) before and after vaccination against influenza. This data consists of resequenced samples from an earlier published study (*Lasserson et al*[31]).
- *Wu et al*[165] (the DDW dataset) downloaded from Zenodo (via the URL https://zenodo.org/record/1161143#.XSxOoZNKjOQ). This dataset contains repertoires from 12 healthy individuals (six of them young, aged 19 to 45, and six elderly, aged 70 to 89) at multiple timepoints (days 0, 7 and 28) before and after vaccination against influenza.
- *Levin et al*[166] (the Levin dataset) obtained from the Observed Antibody Space (OAS) database[167]. This dataset contains repertoires from eight individuals undergoing specific immunotherapy treatment against allergic disease caused by allergen-specific immunoglobulin-E B cells, with immunoglobulin-E only sequences available at multiple timepoints (days 0, 56 and 365).

Sequences from the Gupta and DDW datasets were processed using an in-house pipeline that incorporates IgBLAST[54] and multiple R scripts. Sequences obtained for the Levin dataset were already processed by the OAS in-house pipeline[167].

2.2.2 Sumrep

The processed, FASTA format sequences derived from the datasets described in section 2.2.1 were used as input into sumrep (which can be set up with either IgBLAST[54] or Partis[52,168,169] as the backend inference engines for BCR/TCR). The dataframes that the frameworks output were passed to the core sumrep functions listed in *Table 2-1*. The output for each sumrep function is a one-dimensional dataframe, the length of which depends on the metric – either length N, where N is the number of sequences, or length $L \times N$, where L is the number of values inferred for each sequence. These dataframes served as the input to various distribution-based visualisation and inference programmes (see section 2.2.3).

Summary Statistic	Description	Annotations	Tools
Pairwise distance distribution	Array of Levenshtein distances of each sequence to each other sequence	IgBlast/Partis	stringdist
Nearest Neighbor distribution	Array of nearest neighbor distances, where the NN distance of a sequence is the minimum Levenshtein distance to each other sequence	IgBlast/Partis	stringdist
GC-content distribution	Array of sequence-wise GC contents	IgBlast/Partis	ape
Hotspot motif count distribution	Array of sequence-wise hotspot counts	IgBlast/Partis	Biostrings
Coldspot motif count distribution	Array of sequence-wise coldspot counts	IgBlast/Partis	Biostrings
Distance from germline to sequence distribution	Array of Levenshtein distances from *germline_alignment* to *sequence_alignment*	IgBlast/Partis	stringdist
CDR3 length distribution	Array of CDR3 lengths, including conserved CDR3 anchors	IgBlast/Partis	Tool-provided
Pairwise CDR3 distance distribution	Array of pairwise Levenshtein distances of CDR3 sequences	IgBlast/Partis	stringdist
Atchley factor distributions	Array of each of the five Atchley factors	IgBlast/Partis	HDMD
Kidera factor distributions	Array of each of the ten Kidera factors	IgBlast/Partis	Peptides
Aliphatic index distribution	Array of sequence-wise aliphatic indices	IgBlast/Partis	Peptides
G.R.A.V.Y. index distribution	Array of GRAVY indices	IgBlast/Partis	alakazam
Polarity distribution	Array of sequence-wise polarity values	IgBlast/Partis	alakazam
Charge distribution	Array of sequence-wise charge values	IgBlast/Partis	alakazam
Basicity distribution	Array of sequence-wise basicity values	IgBlast/Partis	alakazam
Acidity distribution	Array of sequence-wise acidity values	IgBlast/Partis	alakazam
Aromaticity distribution	Vector of sequence-wise aromaticity values	IgBlast/Partis	alakazam
Bulkiness distribution	Vector of sequence-wise bulkiness values	IgBlast/Partis	alakazam
Positional distance between mutations distribution	Vector of positional distances between mutations over all sequences	IgBlast/Partis	sumrep
VJ insertion length distribution	Vector of VJ exon lengths	IgBlast/Partis	Tool-provided
VD insertion length distribution	Vector of VD exon lengths	IgBlast/Partis	Tool-provided
DJ insertion length distribution	Vector of DJ exon lengths	IgBlast/Partis	Tool-provided

Table 2-1 ***Core sumrep functions.*** *List and description of metrics used for analysing the repertoires in the three studies, along with the annotation frameworks they are built on, and the tools required for calculating these metrics.*

2.2.3 Data visualisation

A key challenge for sumrep is to provide effective ways for visualising the summary statistical information it generates; writing Python programs that offer different ways of doing this was one of my contributions to the project.

Initially, probability distribution functions (PDF) were inferred by directly calculating probability distributions for histograms of the input data, but this proved to be excessively expensive computationally. Instead, a kernel density estimation (KDE) technique was used to avoid this computational expense, KDEs being non-parametric and orders of magnitude faster than inferring the "true PDF" over a histogram. Further improvements in speed were achieved by exploiting sumrep's subsampling technique.

Whereas a PDF is highly informative about the general shape of a distribution and global trends in the data, the smoothing effects of a PDF are ill- suited to the visualisation of data distributions that contain spikes that may represent an important data feature. Consequently, an alternative approach using frequency polygons (FPg) and FPg-based empirical cumulative distribution functions (ECDF) were investigated.

FPg is a "discrete distribution" well-suited to spiky data distributions, which is simply plotted by inferring a histogram over the data and connecting the peak of each histogram bar to the flanking peaks. Given that the core of this process is reliant on the inferred histogram, the binning mechanism becomes the most important factor in plotting FPgs. Several binning techniques were explored, as follows: a fixed bin width of 50, the Sturges rule[170], the Freedman-Diaconis rule[171], the maximum of the Freedman-Diaconis and Sturges rules, and two Bayesian methods – Bayesian Blocks[172] and Knuth's rule[173]. Using a fixed value has clear disadvantages when dealing with diverse distributions, and while the set bin width may be appropriate for certain distributions, it likely fail to capture the correct shape of others. Both the Freedman-Diaconis and Sturges rules infer the bin width according to rules of thumb incorporating the number of observations and their standard deviation. Although preferable to a fixed bin width, these heuristics are not guaranteed to generate a ‘best-fit’ histogram or PDF. The Bayesian binning methods, on the other hand, overcome this problem by finding the bin width that results in a distribution of ‘best-fit’ by using maximum likelihood or marginal posterior functions to measure the model fitness[172,173]. Bayesian blocks takes this a step further by dispensing with a universal bin width, and instead evaluating fitness at varying widths for every bin, which often results in several bins being merged into one. However, in the context of this research, this raises two potential issues: Bayesian blocks are computationally expensive; and the inferred histograms may have bins with very large variance, leading to uninformative FPgs. Consequently, Knuth’s rule, which calculates the optimal universal bin width, was chosen, although the computational complexity remains comparatively high. The optimal number of bins, in Knuth’s rule’s equation, is the parameter M which maximizes the function:

$$F(M|x,I) = n\log(M) + \log\Gamma(\frac{M}{2}) - M\log\Gamma(\frac{1}{2}) - \log\Gamma(\frac{2n+M}{2}) + \sum_{k=1}^{M}\log\Gamma(nk+\frac{1}{2}) \quad \textit{(2-1)}$$

where Γ is the Gamma function, n is the number of data points, and nk is the number of measurements in bin k[173,174].

There remained a serious combinatorial problem. Given three studies covering a total of 23 donors, three timepoints per study, two annotation frameworks, and 35 metrics per framework, there were potentially 630 distribution plots to generate and visually compare. This was deemed intractable. To overcome this challenge, I implemented a data aggregation algorithm in Python, whereby distributions from all donors at a single timepoint (using the same framework) are aggregated into a single distribution. This approach was found to successfully captures key variations in the shape of aggregated distributions. The algorithm works as follows: a PDF is inferred for every distribution using the tree-based KDE algorithm of SciKit-Learn[175] and the Grid Search algorithm of Scikit-Learn[175] in order to identify the best bandwidth, whereby the resulting models are evaluated by a maximum-likelihood model fitting algorithm. The resultant models/functions of all the distributions are then used to infer a unified bootstrapped function. First, hundreds of points are randomly chosen from within the range of the min-max values across all distributions. Using the PDFs, bootstrapped values for all those points are then inferred, which generates Y-axis error bands for all the X-axis values. The result is a unified PDF with a confidence band that visually captures all intricacies of the underlying distributions.

2.3 Results and Discussion

Before presenting the results, it is important to note the following about the *Gupta et al* influenza vaccination study. Although ostensibly a “classical” single-challenge dataset, it is notable that all three individuals responded differently to vaccination, with individual GMC (who had been vaccinated against influenza the previous year) showing no high-frequency clonal responses, whereas individual FV had high-frequency clones prior to vaccination.

FPg and ECDF plots were generated from the Sumrep output for each metric, at each timepoint for each framework (*figures 2-1* and *2-2*). While FPg plots are informative in capturing the peaks in metric distributions, ECDF plots are more useful in demonstrating changes in the variance among the distributions and/or shifts of distributions. Most of the metrics for each study did not significantly change at different timepoints or between the distribution of different frameworks, most prominently Atchley factor distributions. Therefore, *figures 2-1* and *2-2* display a subset of these plots, which demonstrate various changes and differences, or lack thereof, among the distributions that may be of interest for varying reasons. The choice of FPg or ECDF to demonstrate the changes/differences among distributions are made on the basis the merits of each kind of plot, as pointed out above. Here is a breakdown of the summary statistics calculated and their descriptions:

- Pairwise distance distribution: This statistic captures the distribution of distances between all possible pairs of sequences in a repertoire. It provides information on the overall diversity and similarity of the repertoire.

- Nearest Neighbor distribution: This statistic captures the distribution of distances between each sequence and its nearest neighbour in the repertoire. It provides information on the local structure of the repertoire.
- GC-content distribution: This statistic captures the distribution of the percentage of guanine-cytosine (GC) base pairs in the nucleotide sequences of the repertoire. It provides information on the nucleotide composition of the repertoire.
- Hotspot motif count distribution: This statistic captures the distribution of the number of occurrences of hotspot motifs, which are DNA sequence patterns that are preferentially targeted by the V(D)J recombination machinery. It provides information on the usage of these motifs in the repertoire.
- Coldspot motif count distribution: This statistic captures the distribution of the number of occurrences of coldspot motifs, which are DNA sequence patterns that are less frequently targeted by the V(D)J recombination machinery. It provides information on the avoidance of these motifs in the repertoire.
- Distance from germline to sequence distribution: This statistic captures the distribution of the distances between the nucleotide sequences of the repertoire and their closest germline gene segments. It provides information on the extent of somatic hypermutation and selection in the repertoire.
- CDR3 length distribution: This statistic captures the distribution of the lengths of the CDR3 regions in the amino acid sequences of the repertoire. It provides information on the diversity and structure of the CDR3 regions, which are particularly important for antigen recognition.
- Pairwise CDR3 distance distribution: This statistic captures the distribution of Levenshtein distances between all possible pairs of CDR3 sequences in the repertoire. It provides information on the diversity and similarity of the CDR3 regions.
- Atchley factor distributions: This statistic captures the distributions of the Atchley factors[176], which are physicochemical properties of the amino acids. It provides information on the distribution of these properties in the repertoire.

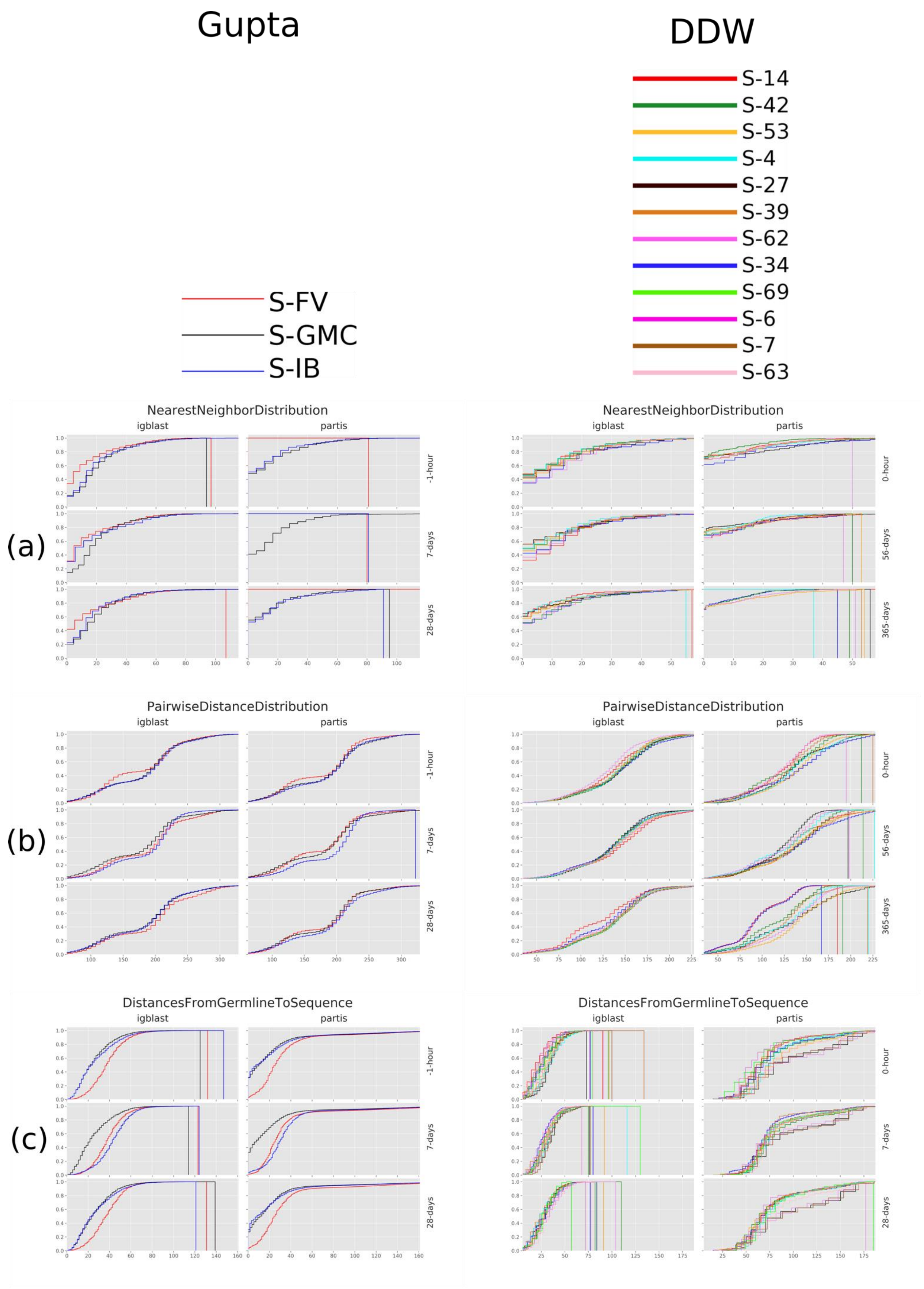
- Kidera factor distributions: This statistic captures the distributions of the Kidera factors[177], which are another set of physicochemical properties of the amino acids. It provides information on the distribution of these properties in the repertoire.
- Aliphatic index distribution: This statistic captures the distribution of the aliphatic indices across repertoires. The aliphatic index measures the relative volume of aliphatic side chains (alanine, valine, isoleucine, leucine) in proteins, indicating their hydrophobicity and thermostability, where a higher index suggests greater hydrophobicity and potential stability, especially at high temperatures[178].
- GRAVY index distribution: This statistic captures the distribution of the GRAVY indices across repertoires, which is a different measure of the hydrophobicity of the proteins. GRAVY is calculated as the sum of hydropathy values of all the amino acids, divided by the number of residues in the sequence [179].
- Polarity distribution: This statistic captures the distribution of the polarities of the amino acids in each repertoire. The polarity distribution in a protein refers to the arrangement and frequency of polar and non-polar amino acids within the protein's structure. Polar amino acids are those that have side chains that can participate in hydrogen bonding (due to the presence of electronegative atoms like oxygen or nitrogen), while non-polar amino acids have side chains that are hydrophobic and do not participate in hydrogen bonding[180].
- Charge distribution: This statistic captures the overall distribution of the arrangement of positively and negatively charged amino acids within antibodies across a repertoire.
- Basicity distribution: This statistic captures the distribution of the frequency of the basic amino acids per antibody across repertoires.
- Acidity distribution: This statistic captures the distribution of the frequency of acidic amino acids per antibody across repertoires.
- Aromaticity distribution: This statistic captures the distribution of the aromaticities of the amino acids. It provides information on the distribution of these properties in the repertoire.

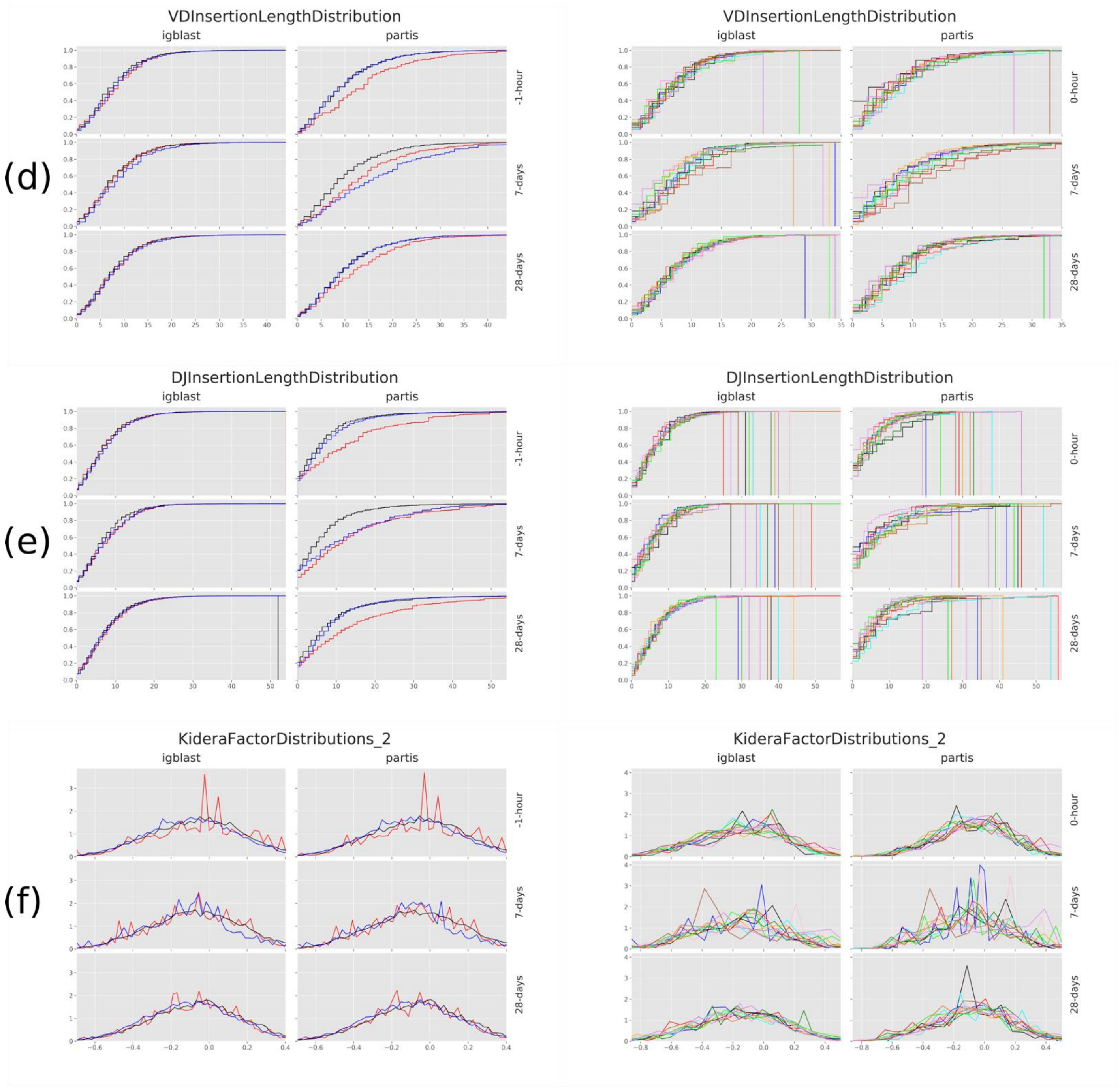
- Bulkiness distribution: This statistic captures the distribution of the bulkiness of the amino acids. It provides information on the distribution of these properties in the repertoire.
- Positional distance between mutations distribution: This statistic captures the distribution of the distances between somatic hypermutations in the nucleotide sequences of the repertoire. It provides information on the distribution of these mutations and their potential effects on the antigen recognition properties of the repertoire.
- VJ insertion length distribution: This statistic captures the distribution of the lengths of the insertions between the variable (V) and joining (J) gene segments in the nucleotide sequences of the repertoire. It provides information on the diversity and structure of these insertions.
- VD insertion length distribution: This statistic captures the distribution of the lengths of the insertions between the variable (V) and diversity (D) gene segments in the nucleotide sequences of the repertoire. It provides information on the diversity and structure of these insertions.
- DJ insertion length distribution: This statistic captures the distribution of the lengths of the insertions between the joining (J) and diversity (D) gene segments in the nucleotide sequences of the repertoire. It provides information on the diversity and structure of these insertions.

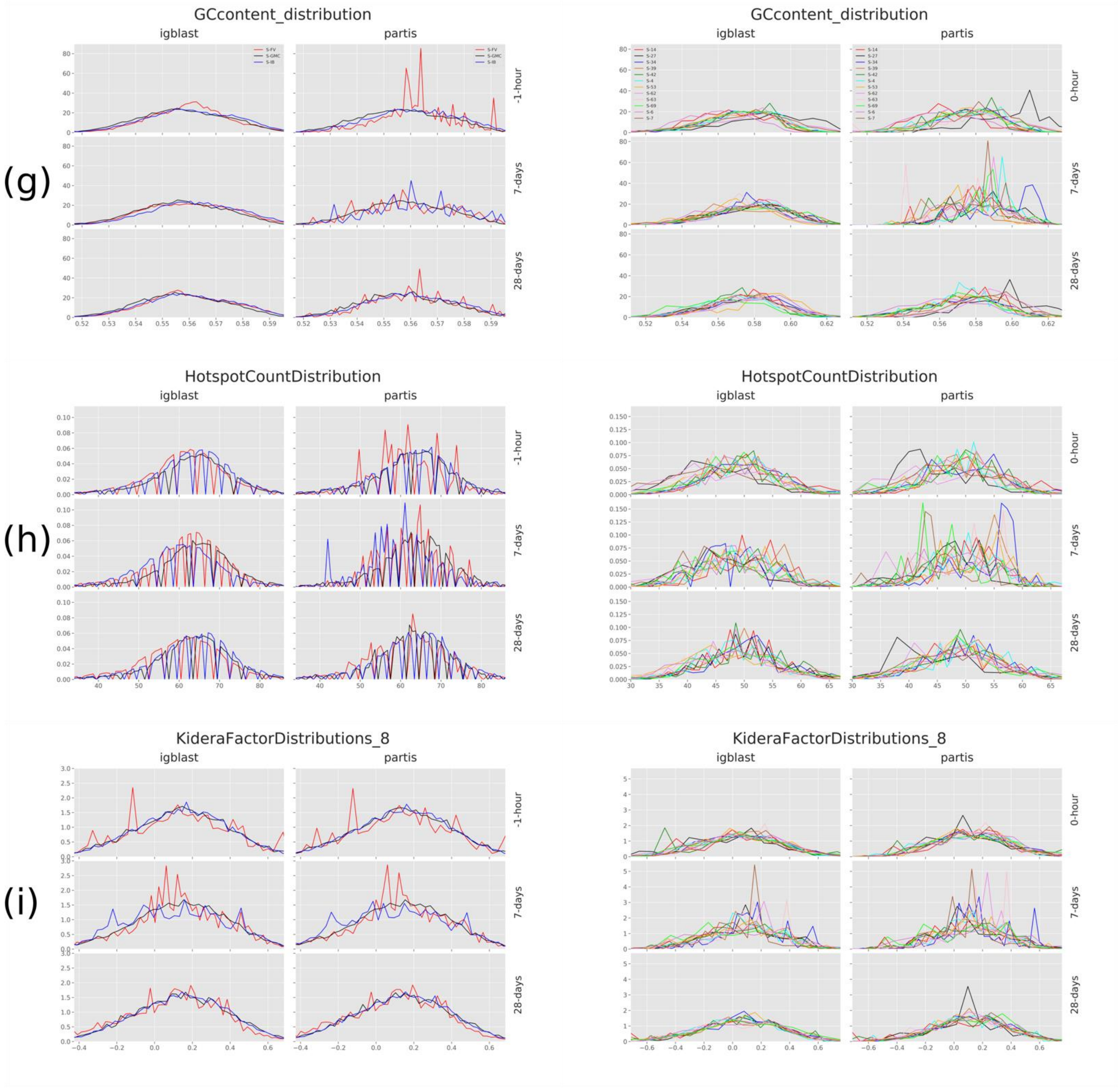
Nearest neighbour distributions largely differ between the two frameworks across both of the studies (*Figure 2-1a*), which is likely due to the big difference the varying annotation parameters of the Partis and IgBlast make on alignment distance among nearest neighbours. This is consistent with some of the other metrics, e.g. Distance from germline to sequence (*Figure 2-1c*), VD insertion length (*Figure 2-1d*) and DJ insertion length (*Figure 2-1e*), which are more significantly affected by the differing annotation protocols of Partis and IgBlast. The importance of comparing, and ultimately evaluating, framework annotations through these summary statistics is illustrated in Nearest neighbour metric (*Figure 2-1a*). We see that in the *Gupta et al* Nearest neighbour example that there are no significant differences across different timepoints of IgBlast distributions, however, there are clear shifts in the Partis distribution at different timepoints. This is also the case, to a certain extent, in the *DDW et al*

Partis distributions of day-7 to day-28. Again, this observation is somewhat consistent with the DJ insertion length distributions.

Interestingly, the Pairwise distance distributions (*Figure 2-1b*) show a subtle, if not insignificant, differences between Partis and IgBlast in the *Gupta et al* study , however, there are significant differences between the two frameworks in the *DDW et al* study. Furthermore, the *DDW et al* Partis distributions also vary at different timepoints. In contrast, the Kidera factor 2 (*Figure 2-2f*) distributions are very similar at the same timepoints across the board, where most of the subtle peaks and more general changes in the distribution shapes are captured in both Partis and IgBlast distributions. This is largely the case for the Kidera factors 7 and 8. Finaly, Hot spot count distribution hardly shows any variation in the IgBlast distributions, however, it shows large variation for Partis.


Gupta
DDW
S-14
S-42
S-53
S-4
S-27
S-39
S-62
S-34
S-69
S-6
S-7
S-63
S-FV
S-GMC
S-IB
(a)
NearestNeighborDistribution
igblast
partis
-1-hour
7-days
28-days
0-hour
56-days
365-days
(b)
PairwiseDistanceDistribution
(c)
DistancesFromGermlineToSequence


(d)
VDInsertionLengthDistribution
igblast
partis
-1-hour
7-days
28-days
0-hour
0.0
0.2
0.4
0.6
0.8
1.0
0
5
10
15
20
25
30
35
40
(e)
DJInsertionLengthDistribution
igblast
partis
-1-hour
7-days
28-days
0-hour
0
10
20
30
40
50
(f)
KideraFactorDistributions_2
igblast
partis
-1-hour
7-days
28-days
0-hour
0
1
2
3
4
−0.8
−0.6
−0.4
−0.2
0.0
0.2
0.4


(g)
GCcontent_distribution
igblast
partis
S-FV
S-GMC
S-IB
-1-hour
7-days
28-days
0-hour
S-14
S-27
S-34
S-39
S-42
S-4
S-53
S-62
S-63
S-69
S-6
S-7
(h)
HotspotCountDistribution
(i)
KideraFactorDistributions_8

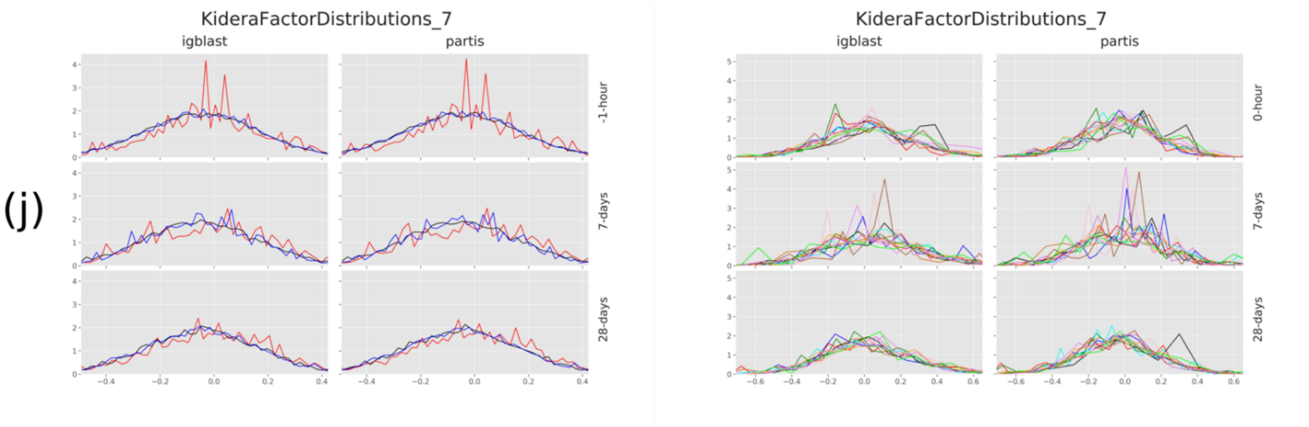


Figure 2-1& Figure 2-2 Selected samples of Summary Statistics distributions. *10 summary statistics distributions, from across two datasets, chosen as a representative subset of a larger set of summary statistics output of sumrep. Each plot has a unique x-axis where the value corresponds to the metric. The y-axis is the normalised probability density. Each column represents corresponding distributions from each study. Individuals within the DDW dataset are labelled as follows: donors S-4, S-6, S-7, S-14, S-27, S-42 are elderly (aged 70-89), whereas donors S-34, S-39, S-53, S-62, S-63, S-69 are young (aged 19-45).*

Corresponding aggregate distribution plots (as described in section 2.2.3) are shown in *Figure 2-3*, with a six-fold reduction in the volume of the data. As *supplementary figure 1* shows a lot of the intricate characteristics of distributions are still represented in the aggregate plots. For instance, comparing the Kidera factor distributions, which are particularly noisy, one can see that despite the compression, the aggregate plots still summarise the distributions quite well without too much unnecessary noise.

Though aggregate plots are helpful in reducing the volume of data visualisation, there is still a significant issue with sole use of visual characterisation of repertoires. It is often difficult to distinguish noise from actual signals when insignificant yet sharp peaks, or shifts, could be interpreted as signals due to human error. To this end, in our sumrep manuscript a lasso regression technique is introduced, by our co-authors, to systematically identify metrics most useful to repertoire characterisation. Such difficulties with human interpretations of high-dimensional data could be an opportunity for machine interpretation by utilising the deep learning models, which recently have demonstrated unprecedented results for analysing high-dimensional data[181–183]. Additionally, Davidsen *et al* used Sumrep's summary statistics for evaluating their deep generative variational autoencoder (VAE) model of TCR β repertoires[184], which is another use case for using summary statistics we have developed.

We observe that most of the distributions are very noisy and difficult to make meaningful interpretations of, even when looking at the aggregated distributions seen in the Supplementary Figure. In conclusion, interpreting such complex high-dimensional data as the comprehensive set of immune repertoire summary statistics through visual means poses many difficulties and, in many cases, may not be informative or even feasible. Furthermore, Sumrep can only handle small amounts of data compared to the size of AIRR sequencing data. We address these issues in the next chapter by reimplementing an high performace computing (HPC) version of Sumrep, capable of analysing hundreds of millions of sequences, which we use in conjunction with the deep learning pipeline, which given the observed challenges that arise from such a feature-rich system could justifiably prove to be a more promising approach.

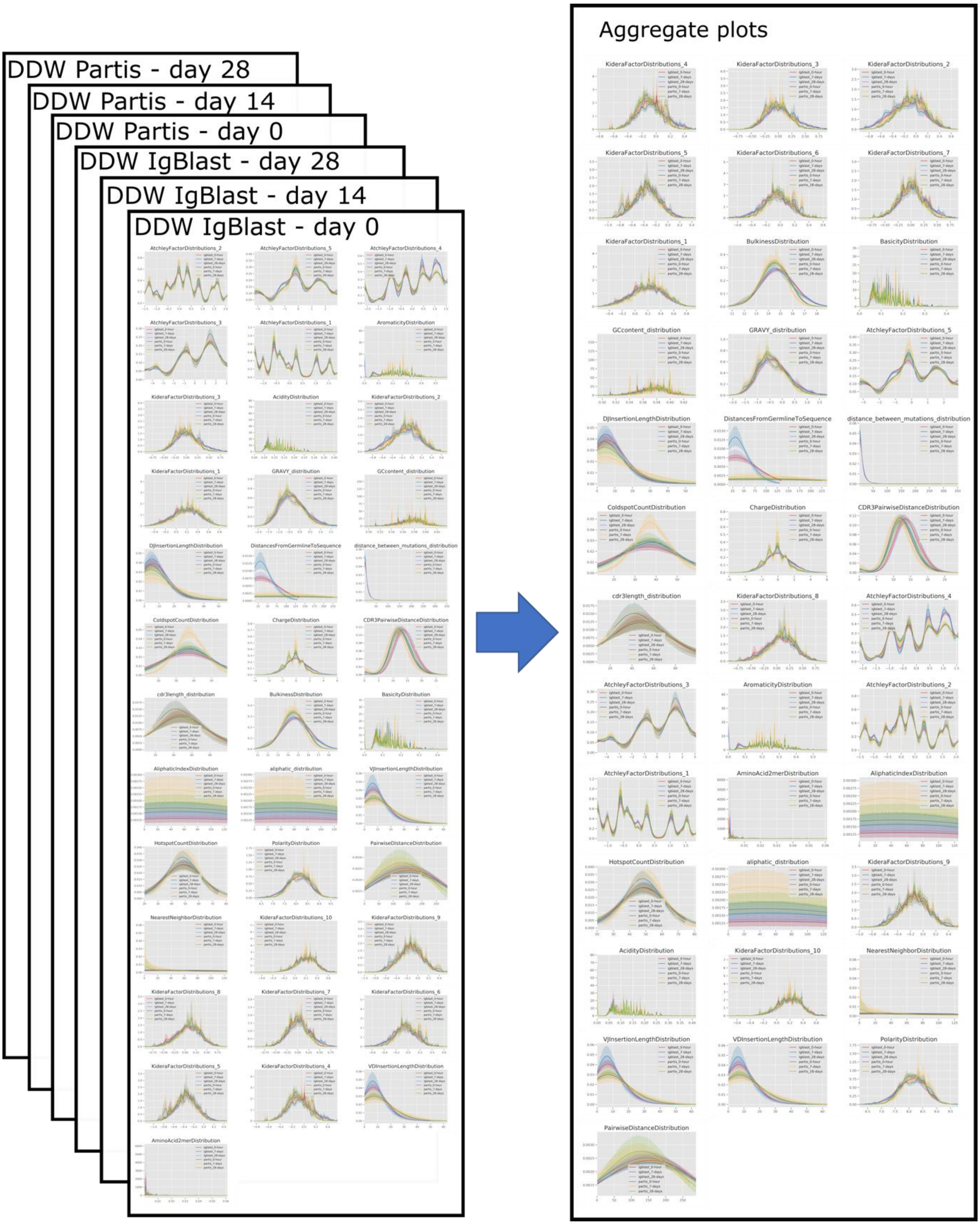


Figure 2-2 An outline of the reduction in the volume of the distribution data by the data aggregation method.

3 Prediction of the population-wide degree of commonality of antibody clones using deep learning

3.1 Introduction

3.1.1 Background and Motivation

Mammalian antibody repertoires exhibit very high levels of diversity, with the humanantibody repertoire recently estimated to contain 3 × 10^{15} unique antibodies[16]. This diversity is attributable to multiple processes described in chapter 1. V(D)J recombination, which occurs in the early stages of B cell development, introduces both combinatorial diversity (associated with the somatic rearrangement of immunoglobulin genes) and junctional diversity (involving the potential addition and/or removal of nucleotides). Additional diversity arises from the subsequent introduction of somatic hyper-mutations SHM during the B cell maturation process.

Although these processes generate high levels of diversity, antibody sequences are not randomly distributed throughout the space of potential antibodies; rather, the processes exhibit bias, such as the preferential usage of certain V and J genes[185] and sensitivity to sequence properties, including known hot and cold spot motifs, that affect the prevalence of SHMs at different positions[186]. For example, the probability that a given antibody sequence will be generated by V(D)J recombination varies by approximately 20 orders of magnitude and can be calculated with a fair degree of accuracy[187,188].

Nevertheless, if one compares the antibody repertories from two randomly selected healthy humans, most of the sequences will be unique to only one of them and are considered “private”, whereas the number of shared or “public” antibody sequences is correspondingly small. This concept of public and private antibodies is potentially important, both for our general understanding of the systemic properties of immune systems, and in the context of vaccine design, where it is considered desirable to stimulate responses that are both effective and common among individuals[189–191]. As shown later, the public-private characteristics of antibodies are only partially correlated with their frequency. The extent to which the frequency of precursor antibodies within the pre-vaccination repertoire may be an additional factor in determining the effectiveness of the response induced by a vaccine is poorly understood[192]. However, some relatively low-frequency, though public nonetheless, antibodies could play an important role in identifying better targets for vaccine design, notably, antibodies which bind to HIV gp120[190] and the stalk of influenza hemagglutinin[193]. The reasoning is that if we could predict how common, in the broader human population, the effective antibody response(s) to our vaccine target would be, we could identify vaccine targets that could evoke effective immune responses in larger portions of the population. This is because we would expect that, the higher the degree of commonality of an antibody, the greater the likelihood that each individual's repertoire would have the potential to express it. For these reasons, predicting whether specific antibodies are public or private is an interesting and potentially biomedically useful task.

In practice, the commonality between antibodies is typically not calculated based on the combined sequence identities of whole heavy and light chains (such an approach would be sensitive to the choice of protocols that affect sequencing length and the prevalence of sequencing errors), but rather calculations are made separately for heavy and light chains using some definition of a shared clone or clonotype. One common definition is sharing identical V and J gene segments usage as well as CDR3 sequence (V3J)[16–18]. Less stringent approaches, focusing exclusively on identical CDR3s, have also been adopted[194], and recently the concept of public antibodies has been extended to incorporate structural and functional equivalences[195]. Often it is only the heavy chain that is investigated[16,18], as it is more diverse than the light chain and its CDR3 is often considered the dominant contributor to antibody binding specificity (see, for example, *Xu & Davis*[196]).

With the advent of ultra-deep sequencing studies using leukapheresis (a process whereby, in the present context, B cells are separated from an individual's blood and the remaining blood constituents are returned to circulation), we now have a better picture of how many sequences are shared between individuals. Briney and co-workers observed a shared heavy chain V3J clonotype frequency of

0.022% between all 10 healthy subjects in their study, with 1.57% being the highest shared proportion between any pair of subjects[16]. Soto and co-workers observed a shared heavy chain V3J clonotype frequency of 0.3% between three healthy subjects, with 6% being the highest shared proportion between any pair[17]. As noted in the latter study, some of the shared clonotypes may be attributable to subjects having been exposed to the same, common antigens, but this is not the only factor, as shared V3J clonotypes were also observed in samples from the umbilical cords of three neonates[17].

Previous attempts have been made to predict whether human antibody sequences are public or private. Greiff and co-workers used support vector machines to predict whether naïve human heavy chain antibody sequences sharing the same heavy-chain CDR3 were "public clones", that is occurred in at least two of the three healthy subjects from which sequences had been collected[18]. This study achieved ~80% accuracy over their binary classification problem.

Given the number of subjects (three) used in that earlier research, their coarse-grained approach to the public-private question is understandable but has become less satisfactory as the number of potential subjects increases; in such circumstances, it seems reasonable that one should make a distinction between V3J clonotypes (or public clones) shared between just two subjects and those shared between many or all of them.

In this research, the focus is exclusively on heavy chain V3J clonotypes combined from the two ultra-deep datasets described above[16,17], producing an initial set of $>$ 2 billion BCR heavy chains from a total of 13 healthy subjects. Rather than address the public-private problem as a binary task, it has been reformulated as a regression task, where the challenge is to predict the degree to which a given V3J clonotype is shared between all individuals.

3.1.2 The Blind Mapmaker Hypothesis

Here, I briefly outline my thoughts on why some cases of immune convergence may be more profound than epitope- or antigen-specific examples; consequently providing an argument in favour of the existence of abstract genomic features that justify the use of machine learning to predict the Degree of Commonality (DoC), hereby referred to as the "Blind Mapmaker Hypothesis".

The existence of public clonotypes raises intriguing questions about the immune system's underlying mechanisms and optimisation strategies. The immune system is a complex adaptive system that has evolved to balance multiple goals simultaneously, such as effectively identifying and eliminating pathogens, reducing the risk of autoimmunity, and limiting damage to the host's tissues. Pareto optimality is a concept that helps us understand how the immune system navigates these competing goals by finding the best possible trade-offs. In simpler terms, the immune system looks for solutions that excel in one or more objectives without being worse in any others, which can be represented as "attractors" or points of equilibrium on a graph called the Pareto-optimal front, illustrating the ideal balance among the different goals.

Each of the individual objectives mentioned above could be a generalisation of an underlying set of many objectives; therefore, in viewing the immune system this way, it is essential to note that the optimisation is operating at different, but not independent, layers. Naturally, this results in a fitness phase space, the high-dimensional aggregate of fitness landscapes for all possible objectives. The attractor/non-dominated solutions in this phase space may not necessarily occupy an optimum on every subset of fitness landscapes but are not dominated by other local solutions in their neighbourhood of the fitness phase space.

These public clones could arise due to genetic, structural, or functional constraints, adapted over long and short evolutionary periods to enable recognition of common or conserved epitopes found in many pathogens while satisfying all other objectives. Consequently, shared clonotypes might represent attractors on the Pareto-optimal front in the fitness phase space of the adaptive immune system. For example, considering a hypothetical case in which the only multi-objective optimisation task concerns the effectiveness of clonotypes' primary response, the fitness phase space corresponds to the aggregate of BCRs' fitness landscapes against every possible epitope. It stands to reason that, in this phase space, attractor solutions are clonotypes capable of providing the broadest possible coverage of the epitopes compared to most other clonotypes in their neighbourhood. Subsequently, in this paradigm, by finding these attractor solutions, the immune system effectively learns how to roadmap the space of possible epitopes, enabling it to cover in a consistently tractable time thanks to these attractors. In a simple two-dimensional map analogy, one can think of these attractor solutions as somewhat similar to spaghetti junctions, providing a good starting point of travel when the destination is not predetermined, hence the name "Blind Mapmaker Hypothesis". Similarly, by reformulating this multi-objective immunity problem as a related problem, namely the maximum-flow problem, as attractors, the shared clonotypes would function as sources that maximise the flow of immune recognition and response in a multi-sink problem, where the sinks are epitopes we are likely to encounter.

Following these arguments, one can expect the sequences of shared clonotypes to contain genomic signals related to the plasticity required for their hypothetical function as these attractor solutions. Given the complexity of the immune system and the high-dimensional nature of the data it generates, we can expect high levels of noise-to-signal ratio in these sequences, which deep neural networks excel at learning. Regardless of the validity of this hypothesis, even the sequences of disease-specific public responses may still present some genomic patterns generic to convergence.

3.2 Methods

Here we briefly summarise the data processing steps used in the results section. It is important to note that in specific figures, the DoC labels were downshifted by 1, e.g. DoC_0 to DoC_7 labels would correspond to DoC_1 to DoC_8 labels. Additionally, specific figures only have DoC values up to a maximum of 7, owing to the lack of complete data at the time of the analysis.

In summary:

- Sequence data were de-duplicated.
- Sequences were removed if there was ambiguity in their nucleotide and or amino acid sequences; if they were light, rather than heavy, antibody chains; if they were considered unproductive; if

they were shorter than 85 amino acids in length; and if the CDR3 contained fewer than 5 or more than 35 amino acids residues.

- The remaining sequences were assigned to Convergent Clusters according to the V3J clonotype criteria.
- Sequences were then labelled with their respective DoC values calculated by identifying the number of subjects with sequences belonging to the same convergent cluster.
- The remaining sequences were undersampled according to DoC-based criteria informed by EDA.
- Leak-free splitting of the set into 10 partitions was performed to facilitate the 10-fold cross-validation of the final deep neural network model.

3.2.1 Data Collection

Arguably, one of the biggest challenges of working with immune repertoire data, besides the inherent complexity of antibody sequence generation mechanics, is the sheer amount of data due to the enormous size of unique clonotypes in immune repertoires. One can argue that we have only begun to experience this challenge, as most of the sequencing studies remained very shallow, until recently, by only sequencing at most several hundreds of thousands of sequences per subject at a coverage rate of less than 1%. Nonetheless, the lack of high-coverage sequencing data poses a bigger problem than the difficulties of handling such large data. A Particular issue in this research is that immune repertoires can be biased in many different ways (e.g. gene usage and VDJ recombination probabilities), resulting in wildly skewed clonotype frequencies, which are reported to be power-law-distributed[14,197,198]. Additionally, DoC is long-tail distributed (see Modelling Class Sample-Size Distribution). Consequently, extreme sequencing depth and width (number of individuals sampled) are required to assign the degree of commonality accurately.

In this research, I pooled together, to the best of my knowledge, the two largest open-source antibody repertoire datasets available, which were extracted from *Briney et al*[16] and *Soto et al.* [17], who reported sequencing almost 3 and 1.72 billion heavy-chain consensus sequences from ten and three individuals respectively. They report capturing over 50% (at the most conservative estimates) of clonotypes from each individual's repertoire, estimated by species richness analyses. The data for each individual from the *Briney et al.* study was collected from the corresponding GitHub repository (see https://github.com/briney/grp_paper) in CSV format. Each subject's data comprises 6 biological replicates, each composed of 3 technical replicates, adding up to 18 samples/files per subject. The *Soto et al.* dataset was obtained from OAS[167] and was subject to the OAS' in-house data cleaning pipeline, resulting in three large CSV files. It should be noted that during the initial phase of this study, data from only 8 subjects from the *Briney et al.* study was used before the data from an additional five subjects became available.

3.2.2 Machine Learning Pipeline and Bag-of-Tricks

All neural network models were constructed using the Tensorflow 2.0 Python library and trained on 8 Nvidia 1080ti GPUs. As far as possible, consistency was maintained with respect to the overall architecture and other parameters across different models. Where differences occurred, these are specified in the relevant sections of this thesis. For instance, the Mish activation function was consistently used for the hidden layers of all models, whereas the choice of output layer activation function depended on the context.

All models were trained in many steps as a result of dividing the training and validation splits into batches. To ensure that low-population classes were included in all batches, a batch size of 4096 was used. The data was randomly sampled into batches prior to the implementation of "leak-free" batching. *He normal* weight initialisation, a method which is particularly useful for avoiding the vanishing gradient problem[199], was used for all layers unless otherwise specified. The "Ranger" optimisation function was used, which is a combination of the Rectified Adam optimisation function[200] and the Lookahead technique[201], with the parameters summarised in *Table 3-1*.

*Max Le*arning Rate	Min Learning Rate	Warmup Proportion	Total Steps	Sync Period	Slow Step Size
0.01	0.0001	0.2	data-size/Batch-size	5	0.5

Table 3-1 The machine learning parameters. *Some of the machine learning parameters consistently used across different models.*

Large learning rates were chosen to counteract the effects of the large batch size. Here the aim is to introduce stochasticity into learning, as the larger the batch size gets, the more confidence the optimisation algorithm will have in the direction of the learning, which may or may not be converging to the global minimum or to an acceptable local minima. Early stopping was used to prevent overfitting of the models and reducing training time, with a patience factor of 10 steps. For classification model evaluation, the area under receiver operator characteristic curve (AUC) and confusion matrix (CM) were used, together with various loss functions depending on the type of modelling performed. Note that, in a multi-class model confusion matrix, the top-left-bottom-right diagonal is informative of how the model performs, i.e. the higher the concentration of predictions on that diagonal the better the performance of the model. Finally, for regression model evaluation, mean absolute error (MAE) and mean squared error (MSE) were used, together with MSE as the loss function. Confusion matrices were simply calculated using class-wide true positives, false negatives, false positives, and true negative information and plotted using the matplotlib Python library.

3.2.3 High-Performance Computing

In this research, a wide array of data science and programming tools and libraries written in the Python programming language were utilised. Numpy was used for general mathematical computing and vector operations. Scipy was used for scientific calculations. Pandas was the most extensively used library for dataframe-related operations. Scikit-learn was used for a variety of machine learning-related tasks, including data preparation. These are only few of the many programming libraries used to make this work possible, which was carried out on a server with a CPU with 40 threads, 375gb of memory and 8 1080ti Nvidia GPUs. Whilst, for a large part these versatile tools were indispensable, processing the volume of data that needed to be used in this study would have been impossible, given the relative paucity of the computational resources, without adopting HPC tools and methods. Arguably, the most limiting factor in this research has been memory usage, a consequence of the huge computational costs of operating on dataframes, which was the main (and natural) way that data was handled. Generally, the memory requirement for dataframe processing is several times the size of the dataframe. The computational complexity, however, was a close second as a limitation to this work. If HPC techniques had not been adopted, the memory and time needed to carry out the analyses would have multiplied several fold.

When it comes to the deep learning, Tensorflow offers a consistently well-optimised platform with the possibility of parallel computing, especially given the availability of GPUs. However, this is not the case for data processing libraries, which ultimately demand most of the development time. Beside utilising only a single-core for all operations, a significant shortcoming of Pandas is that data is copied multiple times for most operations, with the most expensive being the merge, join and groupby functions. Even if sufficient random-access memory (RAM) had been available to carry out such operations (which were routine in the work undertaken), processing time on a single core would have been orders of magnitude higher. This is, for the most parts, due to Pandas being written in pure Python, which is not optimised for speed, parallelism, or memory management (for instance in comparison to C++). This is further exacerbated by the fact that Pandas offers no out-of-core functionality or lazy-programming paradigm, which could play a useful role in reducing development time. Despite not having any real impact on reducing the time complexity of computationally intensive algorithms, lazy evaluation can significantly reduce development time, particularly in a dataframe context.

Data analysis of real-world problems involves long and complicated pipelines, where each independent step/line-of-code commonly contribute to a long and complicated procedural algorithm, which is reflected in the complexity of the design patterns required. This is often the case with the Pandas' programming paradigm, and when the task involves handling large amounts data, each step may have a large time complexity that adds a significant burden in terms of development time and effort. As an alternative, lazy evaluation (although implementations vary in different HPC libraries), can potentially eliminate this problem by allowing one to execute an algorithm with a very small subset of the data first, and postpone the full computation until such time as the results of that computation are needed elsewhere in the pipeline.

The most logical way to avoid large memory costs is by adopting a zero-copy policy, which has been implemented effectively by the Vaex library; instead of copying data, Vaex creates references to the data it needs. Although underdeveloped compared to pandas, Vaex was used extensively in this research, particularly for plotting in the EDA process (given it's efficient integration with the Python Matplotlib library).

Another way to avoid out-of-memory problems is by out-of-core computation, which is a concept is mainly used in the context of HPC and big data processing. Out-of-core algorithms are designed to process data that is too large to fit into a computer's main memory all at once. This is done by loading a portion of the data into memory, processing it, then moving it out and bringing in the next portion of data. This method is often used in libraries like Modin or Dask to handle large data sets efficiently. Though superficially similar to the concept of virtual memory, regarding the overflow of data to disk to prevent out-of-memory issues, the scope, dynamics and implementations of the two concepts are very different. Out-of-core computation is a technique used in specific algorithms and libraries and is typically implemented at the application level, whereas virtual memory is a feature of the operating system that abstracts memory management for all applications running on a system. Dask and Modin were used extensively to handle our large datasets, and while Dask was far more versatile, both suffered from a number of unexpected behaviours and/or errors (attributable to binary-formatted files lacking full integration into the underlying Ray engine). Vaex was also used as an HPC alternative for Pandas which handles parallelism and memory issues differently, however, this library could not provide an end-to-end for all of the necessary computations.

As a gold standard in the field, Apache Spark is arguably the most versatile tool available, especially since Spark 3.2.0 version integrated Pandas-on-Spark (formerly known as Koalas) into the Pyspark library. During most of the research conducted for this thesis, Pandas-on-spark was not available, although various Pyspark modules based on the Apache Spark engine were utilised, as this engine has various effective strategies for dealing with memory problems and promoting faster computation. The most fundamental and powerful aspect of the Spark engine is the resilient distributed dataset (RDD), which is a set of partitions of data distributed across multiple cores/nodes in a cluster. Importantly, RDDs are resilient to node failures and recoverable (a feature missing from Modin). Furthermore, the Spark engine is very flexible and efficient with respect to memory usage and uses in-memory Pyarrow columnar data for faster data transfer and computation. On the other hand, the Pyspark package suffers from the common pitfall of not having a full implementation of pandas functionality.

In summary, there is currently no single package that provides the end-to-end functionality required for research such as this with complex data processing requirements. At the time of writing, Modin and Vaex lack the maturity and the user base, making the development of an effective and reliable end-to-end data science pipeline both challenging and time-consuming. On the other hand, the combination of PySpark and Dask provides the best available solution for developing such complex end-to-end pipelines for big data processing.

3.3 Results

All the results presented here concern the prediction of the degree of commonality (DoC) of V3J clonotypes within a complete set of > 400 million antibody heavy chain sequences (or subsets thereof) compiled from (to the best of our knowledge) the two deepest AIRR-seq datasets yet published.

Here, we performed multitude of exploratory data analysis (EDA) and data processing steps and modelled this task in variety of ways, e.g. binary and multi-class classifications. Across all different types of modelling, we extensively tried and tested variety of neural network architectures, with as much consistency as possible. In the final type of modelling, i.e. the regression modelling, we additionally designed a Transformer neural network architecture to compare with the consistently used squeeze-and-excitation temporal convolutional neural network (SE-TCN) architecture.

3.3.1 Definitions, Data Processing and Labelling

Given the central aim of this study, a prerequisite for establishing a data processing pipeline is to define what constitutes the genomic convergence across clonotypes within a repertoire and, by extension, across repertoires in a population. In this study, we adopted the "V3J" definition from *Soto et al.* [17] for clonotyping BCR sequences, whereby sequences which share the same V and J gene segments and have identical CDR3 sequences are collapsed into one cluster. Hitherto, when referring to such clusters in the context of an individual's repertoire, we use the term clonotype, and when referring to such clusters shared across multiple repertoires, we use the term convergent cluster.

The authors of the published datasets used in this research had already carried out data pre-processing steps, namely, the standard high-throughput sequencing bioinformatics, e.g. filtering low-quality reads and identifying consensus sequences. Notably, the identified consensus sequences observed across multiple technical replicates of the same biological replicate were collapsed into a single sequence. If identical sequences were observed across multiple biological replicates, all copies were retained (as genuine recurrences rather than PCR-derived duplicates).

For this research, specific sequence annotations – notably V and J gene inferences and the specification of the CDR3 – are of paramount importance. Both studies used their bespoke annotation pipeline for handling their massive datasets: *Briney et al.* used Abstar, and *Soto et al.* used PYIR. The large number of different genes - some of which can be difficult to distinguish - and the extremely high mutation rate involved in the antibody sequence generation process, in addition to the biological and sequencing biases, make the gene annotation of antibodies a challenging and error-prone process. One of the factors that can significantly influence the reliability of annotations is sequence length, i.e. the longer the sequences, the more information available for accurate annotations. Therefore, a minimum-sequence-length criterion (85 amino acids) was adopted to minimise the possibility of incorporating misannotated sequences - a decision based on the observed sequence-length distribution in the data and knowledge about the minimum length required to minimise gene annotation errors. Sequences with CDR3 lengths (using IMGT's CDR3 definition) outside the range of 5-35 amino acids (approximately 99% of the data) were excluded to reduce the chances of including potentially aberrant sequences. Additionally, all sequences marked as unproductive by the authors were removed.

The filtered subject-specific databases (in the context of the Python programming language) of biological and technical replicates were merged into a unified database for each subject, and only one copy of identical sequences was retained to avoid overfitting the downstream machine learning mode. However, the number of copies of a given sequence was recorded for subsequent analyses. Finally, all subjects' dataframes were merged, and sequences were collapsed into "convergent clusters" based on the V3J definition. The cluster members were assigned a DoC label (1 to 13), depending on how many subjects their cluster represented.

3.3.2 Binary Classification

The simplest way to model the clonotype "commonality" problem is as a binary classification task, which additionally provided the added benefit of establishing a baseline comparison to the shallow ML model published by Greiff *et al.*[18] before progressing to developing granular models. The study published by Greiff *et al.*[18] was the only published work on an ML approach to this problem at the time, where they used an SVM to model this problem as a binary classification task. In this scenario, BCR clonotypes unique to only one individual, i.e. clonotypes with DoC_1, were considered private, and all other clonotypes as public. Following this definition, all BCR's DoC values were binary encoded with zero or one, with zero corresponding to the private class and one to the public class. BCR amino acid sequences and their binary labels were used as inputs into a simple deep convolutional neural network (CNN), with a single neuron in the final layer to output a single prediction value, namely a logit, normalised to probabilities by the Sigmoid activation function. The Sigmoid function is described by *Equation 3-2*[202–204]:

$$\mathcal{S}(l) = \frac{1}{1+e^{-l}} \tag{3-1}$$

where e is Euler's constant.

Cross Entropy (log-loss) function calculates the model's error by calculating the distance between the probabilities and the ground truth labels. The backpropagation algorithm minimises the error by adjusting the model parameters with respect to the magnitude of the error. As with all the models developed for this research, the mean error is calculated for mini-batches of data, which in the case of binary classification was of size 256 to ensures at least a few examples of the public class are included given the large class-imbalance. The mini-batch cross entropy loss is described in *Equation 3-2*[202–205]:

$$\mathcal{CE}(\theta) = -\sum_{i=1}^{m}[y_i \log(p_i) + (1-y_i)\log(1-p_i)] \tag{3-2}$$

where θ is the model parameters, m is the mini-batch size, y_i and p_i are the ground-truth label and model's prediction for instance i respectively.

This model achieved an accuracy of 78.1% on the test set. However, the model accurately classifies private sequences 97% of the time, while only achieving 70% for the public class, which can be attributed to the large class imbalance in the train and validation splits in favour of the private class.

Therefore, the test set was deliberately undersampled with a 2:1 public to private class-ratio, to examine whether the model's training and validation performance is due to overfitting to the dominant private class. To acquire a more fine-grained view of model performance we also computed the confusion matrices (*Figure 3-1*), enabl*ing* us to examine model performance for every class.

This model achieves a good Matthew's Correlation Coefficient (MCC) score of 0.615 while achieving the Area Under the Receiver AUC scores of 0.764 and 0.696 for the training and test sets respectively (see *Figure 3-2*). Thoug*h t*he MCC score indicates good performance and could suggest that the model may be well-calibrated, there is no information about any possible underlying class imbalance within the public class, i.e. the frequency ratio among public sequences with varying DoC values. For instance, if the pool of public sequences is dominated by the BCRs with lower DoC values, those sequences may be solely, or more significantly, contributive to the true negative predictions, while those with higher DoC values may contribute to false positive misclassifications. As a result, the model's learning could hypothetically become limited to differentiating between the instances of these more abundant "sub-classes" of the public class and the highly abundant private class instances, resulting in a model that could still be uncalibrated while exhibiting good scores. Furthermore, the large ratio of false negatives and false positives in *Figure 3-1 along with* the low AUC score for the test set point to the fact that more granular inspection of the data is required. Finally, the large difference between the training and test sets' AUC values, i.e. 0.068, indicates that the model is overfitting despite heavy regularisation and that expanding the public class into more classes could be beneficial. In future subchapters, we address these issues.

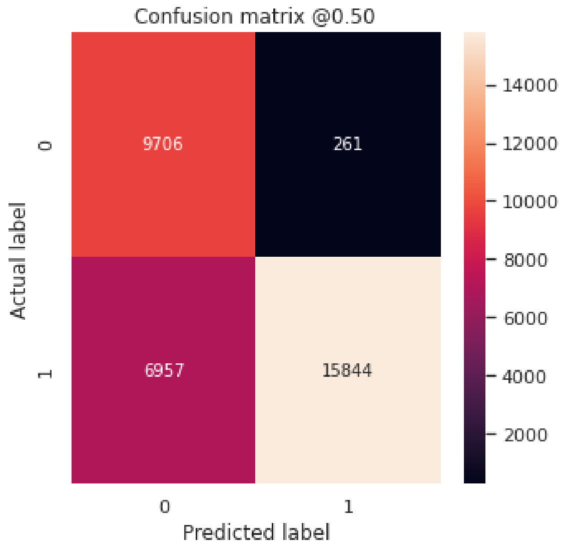


***Figure 3-1 Binary classification confusion matrix for the test results**. The label zero corresponds to the private class and label one corresponds to the public class. The heatmap is scaled in terms of the number of test examples.*

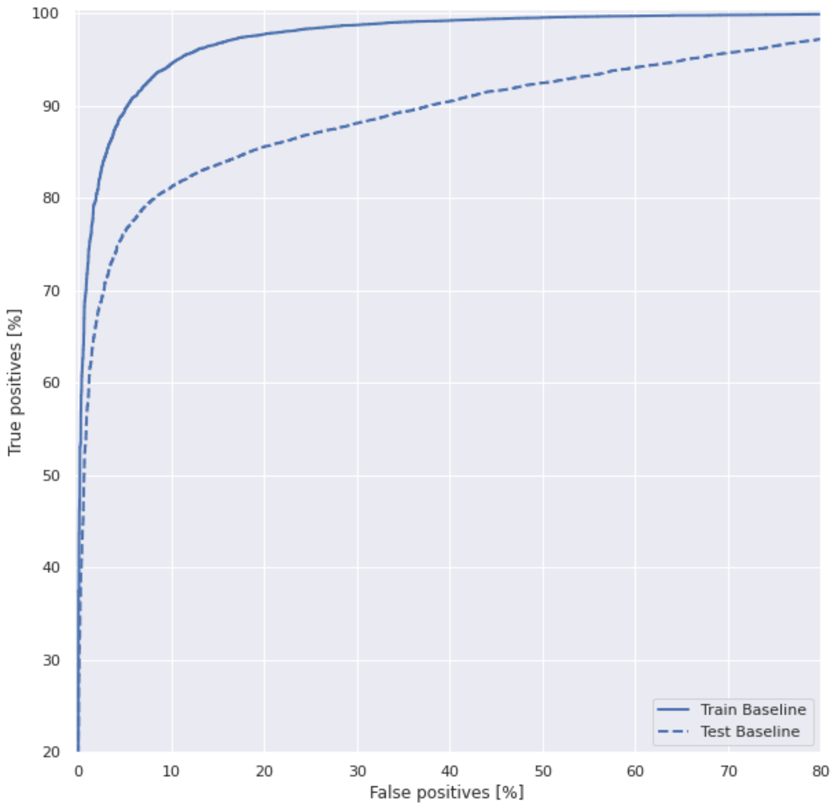


***Figure 3-2 Binary classification AUC curve for the test results**. The ratio of true-positive vs. false-positive predictions of the model is well above the diagonal threshold and close to 100% true-positive prediction, indicative of a well-performing model.*

3.3.3 Exploratory Data Analysis

All attempts for successful application of machine learning problems with “real world” data – data that, unlike benchmark datasets, is not cleaned and tailored for machine learning - particularly those with complex underlying generative processes, imbalance and noise - must start with extensive and careful EDA. Moreover, employing domain expertise, one must determine whether there are correlations between domain-specific factors, which could avoid the need for machine learning altogether, factors that could have been missed from EDA. Given the previously discussed “Blind Mapmaker hypothesis”, one such factor could potentially be a positive correlation between DoC and similarity to germline sequences. To that end, before starting with this chapter's main machine learning contributions, we carried out a series of preliminary EDA, which either guided the following machine learning subsections or were directly or indirectly used in the implementations.

Building upon our prior discussion of the Blind Mapmaker Hypothesis, we sought to investigate the relationship between the DoC and similarity to germline sequences among convergent clusters. By better understanding this relationship, we hoped to gain insights into the factors that drive the emergence and distribution of shared clonotypes.

To analyze BCR sequence data and assess the relationship between DoC and similarity to germline sequences, we randomly sampled 1000 sequences per DoC and employed the *Change-O* R package for the germline-similarity calculations [49]. This package allowed us to calculate the percentage of sequence identity to germline sequences, mutation frequency, and synonymous and non-synonymous mutations in the CDR regions. We then visualized the results using box plots.

As demonstrated by *Figure 3-3, our analy*sis revealed no significant, positive or negative, relationship between the increase in DoC and higher similarity to germline sequences among convergent clusters. This finding was consistent across all calculated metrics, including sequence identity to germline sequences, mutation frequency, and synonymous and non-synonymous mutations in the CDR regions.

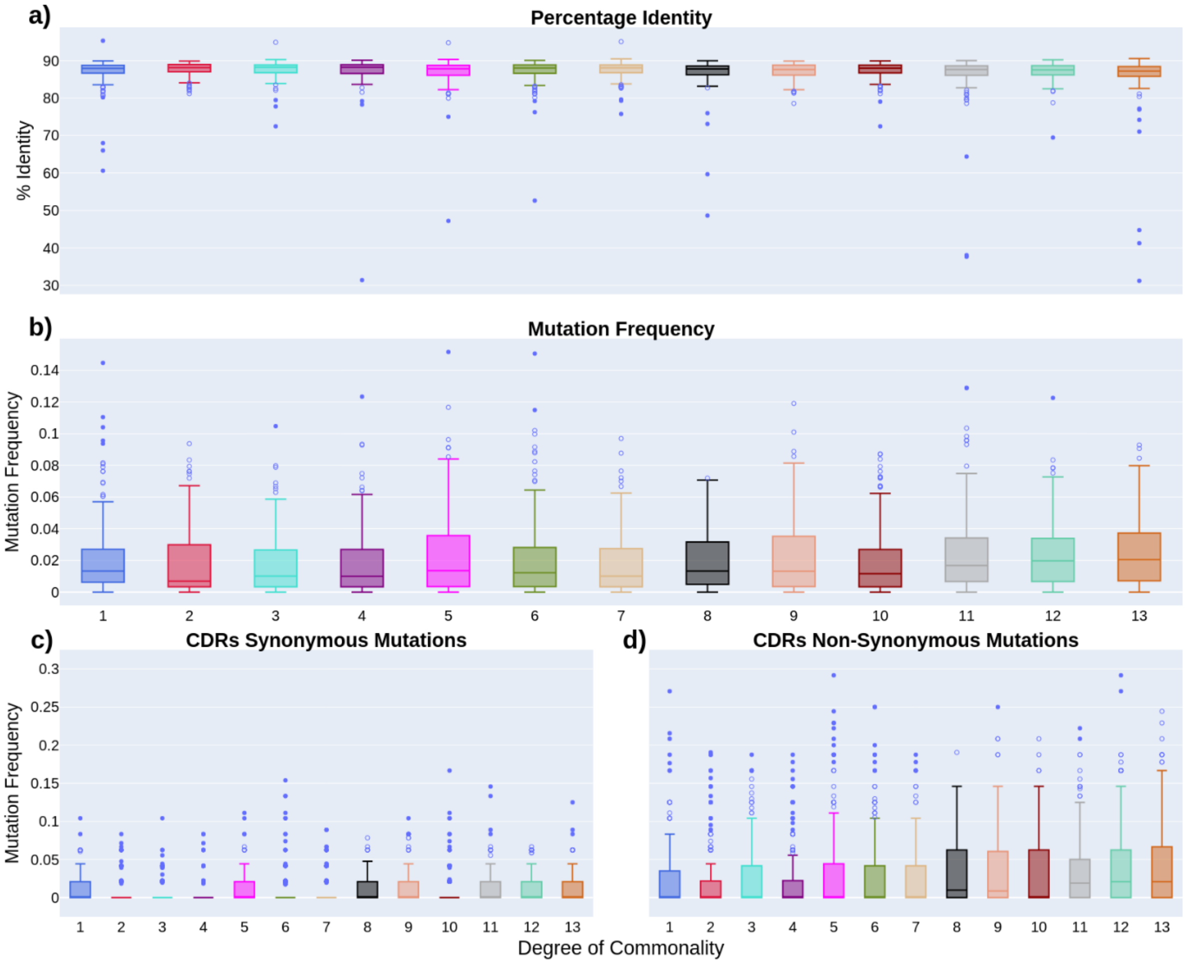


Figure 3-3 Box plots illustrating the relationship between degree of commonality (DoC) and germline similarity measures in convergent BCR clusters*. The figure displays four separate box plots representing (A) percentage of sequence identity to germline sequences, (B) mutation frequency, (C) synonymous mutations in complementarity-determining regions (CDR), and (D) non-synonymous mutations in CDR regions. Each box plot shows the distribution of these measures across different DoC values, demonstrating no significant positive or negative correlation between increased DoC and higher similarity to germline sequences among convergent clusters. The horizontal line inside each box represents the median value, while the box boundaries indicate the first and third quartiles. Whiskers extend to the minimum and maximum data points within 1.5 times the interquartile range. Outliers are represented as individual points beyond the whiskers.*

The absence of a significant relationship between DoC and similarity to germline sequences suggests that other factors may be driving the emergence and distribution of shared clonotypes. One potential explanation for the lack of a significant relationship between DoC and similarity to germline sequences could be the influence of convergent recombination, where distinct recombination events generate similar or identical BCR sequences[206]. This process may result in shared clonotypes with high DoC values but are not necessarily more similar to germline sequences.

Moreover, the role of immune memory in shaping the repertoire of shared B-cell clonotypes should also be considered. Memory cells generated during past immune responses can persist in the body for long periods, providing rapid and robust responses to re-exposure to the same pathogen. Investigating the presence and distribution of shared B-cell clonotypes in memory and naïve cell populations could provide valuable insights into the influence of immune memory on the relationship between DoC and similarity to germline sequences. Moreover, it was recently suggested that processes underlying immune memory formation might favour generating multiple sets of memory B cells, with varying levels of specificity, to strike a balance between diversity and specificity, consequently providing immune competency against future or related pathogenic strains[207].

Following the lack of relationship between DoC and sequence similarity to germline sequences (see *Figure 3-3*), we progressed to EDA for identifying possible patterns in the data. The most basic first step in the EDA of novel data used in predictive tasks is the analysis of label distribution; as such, we started with modelling the distribution of the total frequency of DoC. As *Figure 3-4 depicts, t*his is an extremely imbalanced probability mass function (PMF), sharply decaying with the increase in DoC values, likely due to the enormous theoretical space of immunoglobulins. Despite the enormity of this theoretical space, we still observe all possible DoC values, including fully-public Convergent Clusters, which could be due to the underlying generative processes' biases, e.g. gene recombination biases. Additionally, observing a distribution closely related to fractal structure in the underlying system may suggest complex underlying immunological processes relating to concepts described in the Blind Mapmaker theory governing clonotype convergence.

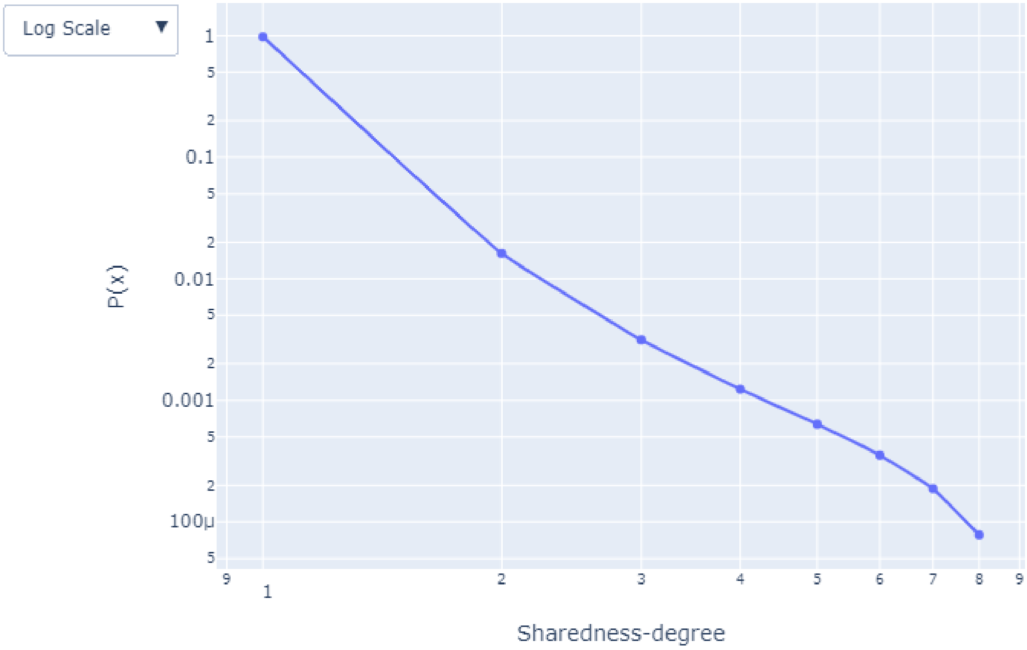
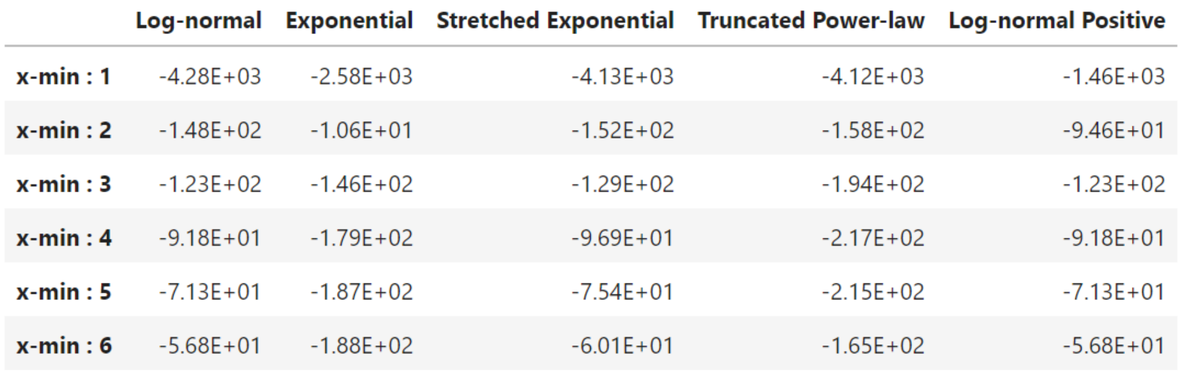


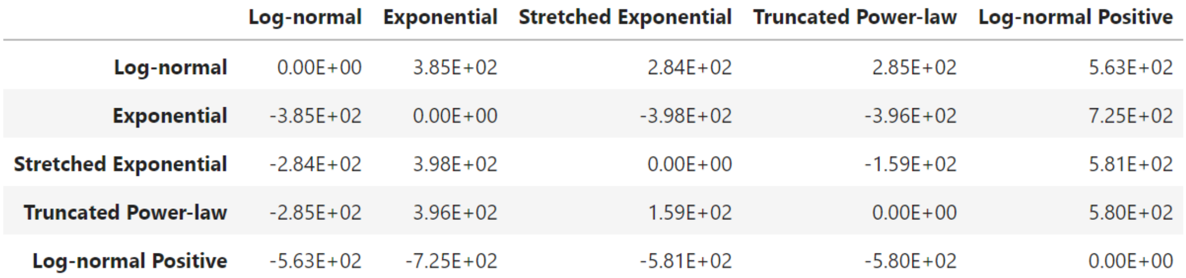
Figure 3-4 PMF of degree of commonality. The PMF of the DoCs plotted with a logarithmic Y-axis almost follows a straight line.

To further inspect this long-tailed distribution, we calculated and evaluated the fitness of five long-tailed distributions, based on various evaluation criteria, using the *Powerlaw* Python library[208]. First, the best minimum x value (DoC 1 to 8) for the fit was found using the Kolmogorov-Smirnov test, summarised in *Table 3-2. Given* $x = 1$ shows the best fit, the models were tested against each other using the Kolmogorov-Smirnov, all with $\min(x) = 1$, as summarised in *Table 3-3.*



	Log-normal	Exponential	Stretched Exponential	Truncated Power-law	Log-normal Positive
x-min : 1	-4.28E+03	-2.58E+03	-4.13E+03	-4.12E+03	-1.46E+03
x-min : 2	-1.48E+02	-1.06E+01	-1.52E+02	-1.58E+02	-9.46E+01
x-min : 3	-1.23E+02	-1.46E+02	-1.29E+02	-1.94E+02	-1.23E+02
x-min : 4	-9.18E+01	-1.79E+02	-9.69E+01	-2.17E+02	-9.18E+01
x-min : 5	-7.13E+01	-1.87E+02	-7.54E+01	-2.15E+02	-7.13E+01
x-min : 6	-5.68E+01	-1.88E+02	-6.01E+01	-1.65E+02	-5.68E+01

Table ***3-2 Log-likelihood of the top-5 heavy-tailed distribution models of the DoC distribution.*** *5 models which describe the distribution of the DoC classes were tested with every DoC as the initial point of the distribution*. *The log-likelihood values for the resulting distributions, given the initial point are recorded here, where the larger values indicate better goodness-of-fit.*



	Log-normal	Exponential	Stretched Exponential	Truncated Power-law	Log-normal Positive
Log-normal	0.00E+00	3.85E+02	2.84E+02	2.85E+02	5.63E+02
Exponential	-3.85E+02	0.00E+00	-3.98E+02	-3.96E+02	7.25E+02
Stretched Exponential	-2.84E+02	3.98E+02	0.00E+00	-1.59E+02	5.81E+02
Truncated Power-law	-2.85E+02	3.96E+02	1.59E+02	0.00E+00	5.80E+02
Log-normal Positive	-5.63E+02	-7.25E+02	-5.81E+02	-5.80E+02	0.00E+00

Table 3-3 Kolmogorov–Smirnov test of the top-5 heavy-tailed distribution models against each other. *Top-5 performant models which describe the distribution of the DoC classes were tested against each other by the Kolmogorov–Smirnov test, where the smaller values indicate better goodness-of-fit. This results indicate that the log-normal, truncated power-law and stretched exponential perform better than the other two models.*

Based on *Table 3-3, the three* best-fit distributions are, in descending order: Lognormal, Truncated Power-law and Stretched Exponential distributions. Although the Kolmogorov-Smirnov test is a gold-standard for evaluating the goodness-of-fit[208], in the absence of a large sample-size, as is the case here, one needs to evaluate the goodness-of-fit graphically and investigate the extrapolation of the theoretical distributions given the approximated parameters. This is demonstrated in *Figure 3-5 and Figure 3-6. Figure 3-5 shows that* the lognormal unexpectedly predicts a larger point-probability value for x_8, and this overfitting may be a contributing factor to its better performance compared to the other two model. Furthermore, the approximated model was extrapolated to larger values to check for infinite or undefined boundaries. The empirical distributions were also extrapolated using the generic power-law model, with the approximated parameters calculated by the Powerlaw library. The extrapolated distributions were plotted up to x_{500} and were truncated if, at any value of x, they were undefined. Despite the log-normal distribution scoring higher than other models based on the empirical data, as shown in *Figure 3-6, it is und*efined beyond the boundaries of the empirical data, i.e. beyond x_8.

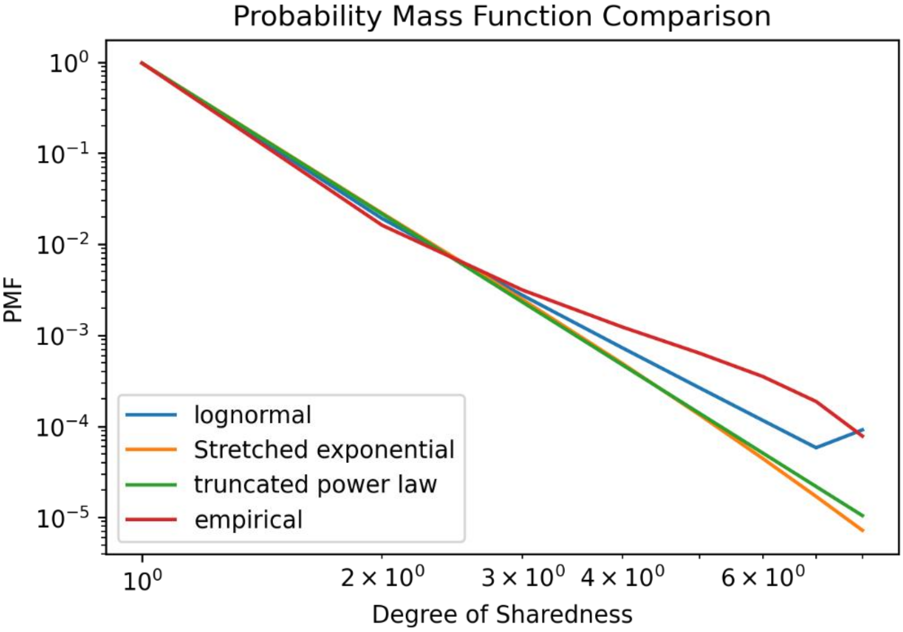


Figure 3-5 Pobability mass funtion of the empirical and top-3 heavy-tailed model distributions. *The distribution of the top-3 models of the class distribution, given the best initial points, were calculated and plotted against the empirical ditribution of the classes. The log-normal distribution displays the closest relationship, however, seems to overfit as is evident by the tail of the distribution giving a very close value to the empirical distribution for the DoC-8.*

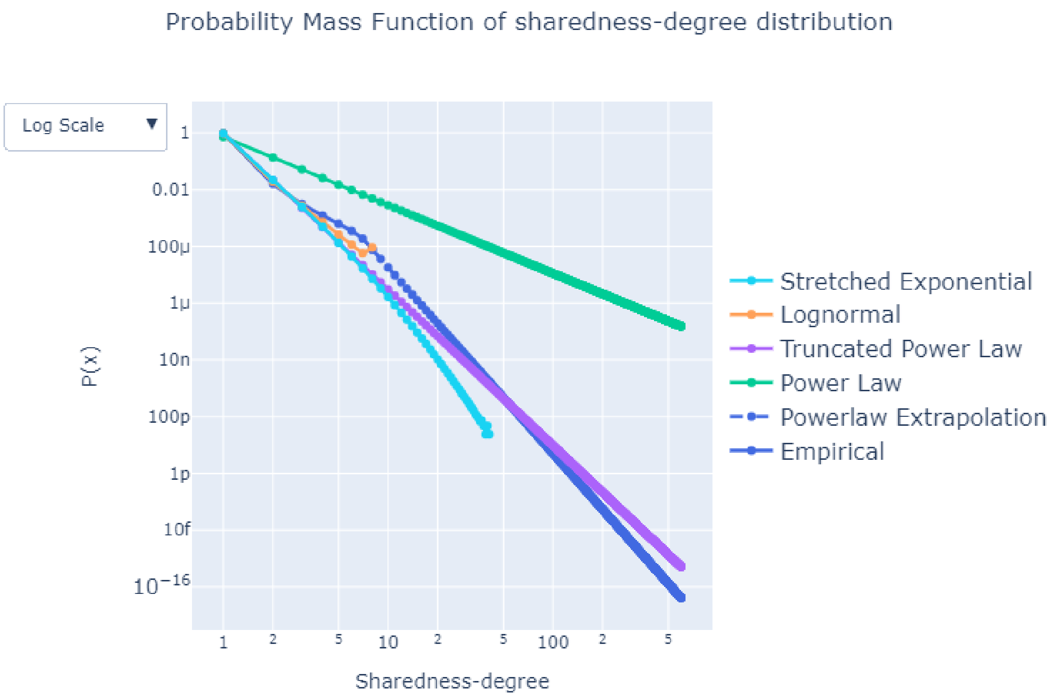


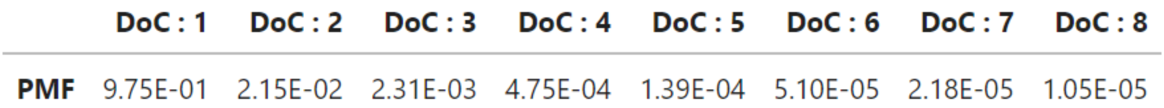
Figure 3-6 PMF of the empirical and top-3 heavy-tailed model distributions extrapolated to larger values. *To investigate the behaviour of the top-3 models of the class distribution, these models' extrapolated distributions were calculated and plotted against the empirical ditribution of the classes, which was also extrapolated using a power-law function. Depite the good fit of the log-normal distribution up to the maximum value of the empirical data, this distribution falls to negative infinity beyond the empirical observations, which is indicative of its goodness-of-fit, within the empirical boundaris, being the product of overfitting. As result, the truncated power-law seems to provide the best fit to the empirical distribution..*

As a result, the second-best model, i.e. truncated power-law, was chosen as the model best describing the distribution of DoC across the theoretical population. Given this model, the probability mass function (*Equation 3-3*) for the *D*oC distribution in the theoretical population was calculated,

$$PMF(n) = \sum_{x=1}^{n} [\frac{e^{-\lambda x}}{\zeta(\alpha) x^{\alpha}}] \qquad \textit{(3-3)}$$

Where ζ is the Riemann zeta function, e is Euler's constant, α and λ are the shape and exponential decay parameters of the truncated power-law respectively and n is the number of classes/max-DoC.

The resulting point probabilities for every DoC are shown in *Table 3-4. Addition*ally, the cumulative distribution function (CDF) for truncated power-law, simply the point probabilities for all points preceding x (denoted as k), is calculated by *Equation 3-4.*



	DoC : 1	DoC : 2	DoC : 3	DoC : 4	DoC : 5	DoC : 6	DoC : 7	DoC : 8
PMF	9.75E-01	2.15E-02	2.31E-03	4.75E-04	1.39E-04	5.10E-05	2.18E-05	1.05E-05

Table ***3-4 The PMF of the truncated power-law for all DoCs.***

$$CDF(x) = \sum_{xk \leq x} PX(xk) \qquad (3\text{-}4)$$

To clarify whether the use of machine learning is likely to be justified for this task, a preliminary analysis was undertaken to explore whether the degree of commonality is correlated with other factors, notably convergent cluster size. In other words, to what extent does the clonotype frequency across one or more repertoires offer an insight into the likelihood of its observation in other repertoires?

To investigate whether convergent cluster size is correlated with the degree of commonality, the distribution of the top 100 and top 3000 convergent clusters, with the largest average size across represented repertoires, over degrees of commonality was analysed. We used a variety of statistical distributions, including exponential, Gumbel, logistic, and normal, to fit the data and performed two tests: the Kolmogorov-Smirnov test and the Anderson-Darling test, to assess goodness-of-fit. Based on the results, we found that the shifted Gumbel distribution best fits the data, as it has the highest log-likelihood and the lowest p-values for both tests. This indicates that the shifted Gumbel distribution is a suitable generative model of the distribution of the largest Convergent Clusters across DoC.

As evident in *Figures 3-7* and *3-8*, the fraction of large (top-*k*) convergent clusters associated with high degrees of commonality is comparatively low while, surprisingly, high across public Clusters with low DoC values. The shifted Gumbel distribution is commonly used to model extreme events, emphasising the extreme asymmetric decay of Convergent Clusters/clonotypes size as the DoC rises, suggesting that the underlying immunological process governing the population-wide Convergent Cluster size is more complex than can be accounted for by simple models. One interpretation could be that the public Convergent Clusters, at the lower bound of the DoC scale, are likely clonotypes with immunoglobulins responding to more common immunogenic epitopes presently pervading the population. In contrast, Convergent Clusters of the higher bounds of DoC may represent those "attractor clonotypes” hypothesised by the Blind Mapmaker theory, which naturally would not be required to exist in large numbers, but be more likely to be converged to as Pareto-optimal solutions. In multi-objective optimisation, there is no single global optimum, but rather, a set of optimal solutions that balance the objectives in different ways referred to as the Pareto-optimal solutions. Further investigation is needed to explore the biological mechanisms underlying the distribution of clonotype frequencies and to determine if our findings generalize to other datasets.

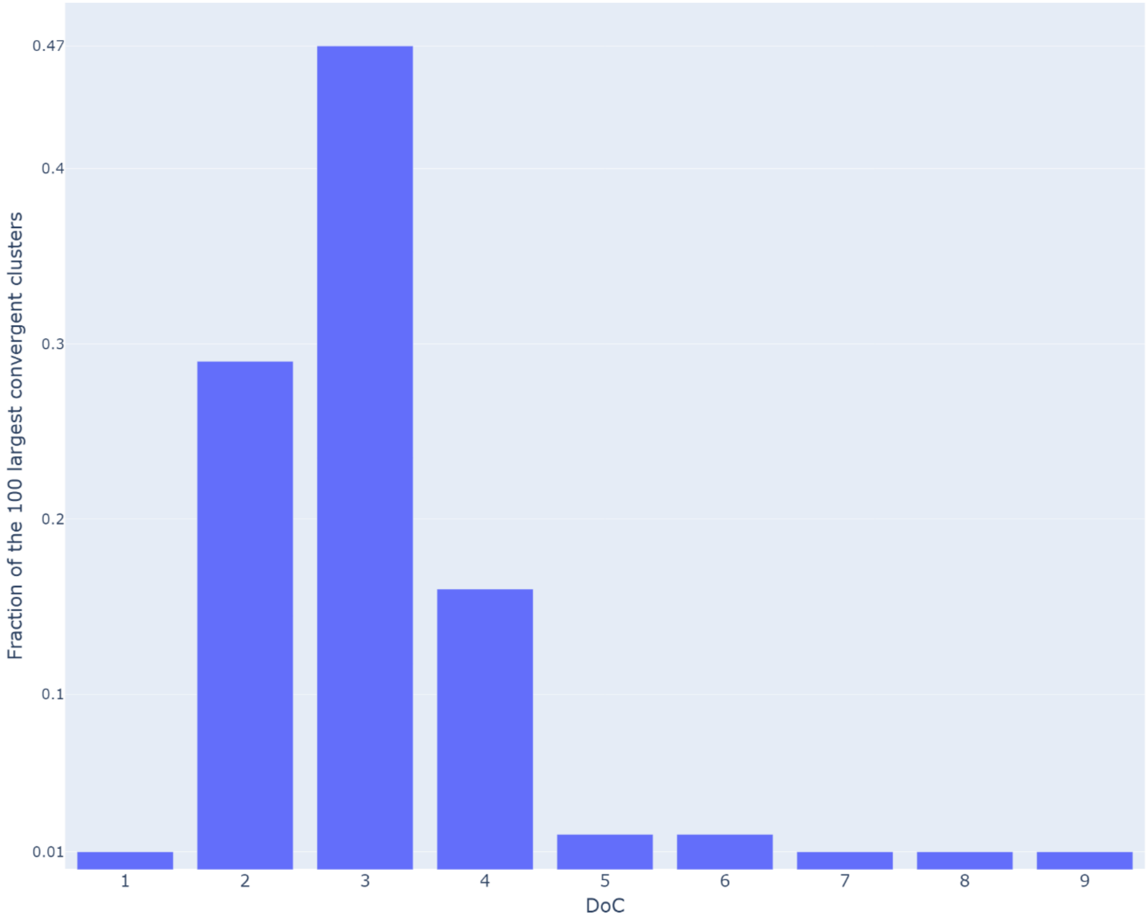


Figure 3-7 Top-100 mean Convergent cluster Size Distribution Over degrees of Commonality. Top 100 *largest convergent clusters were selected, and the fractions of these groups w.r.t. DoCs were plotted. This distribution follows the shape of the shifted Gumbel distribution (p-value: 7.41x10^{-9}), and evidently, even as we increase the number of largest Convergent Clusters included, only the lower-DoC Convergent Clusters are most abundant. In contrast, with increasing DoC, Convergent Clusters become more sparse.*

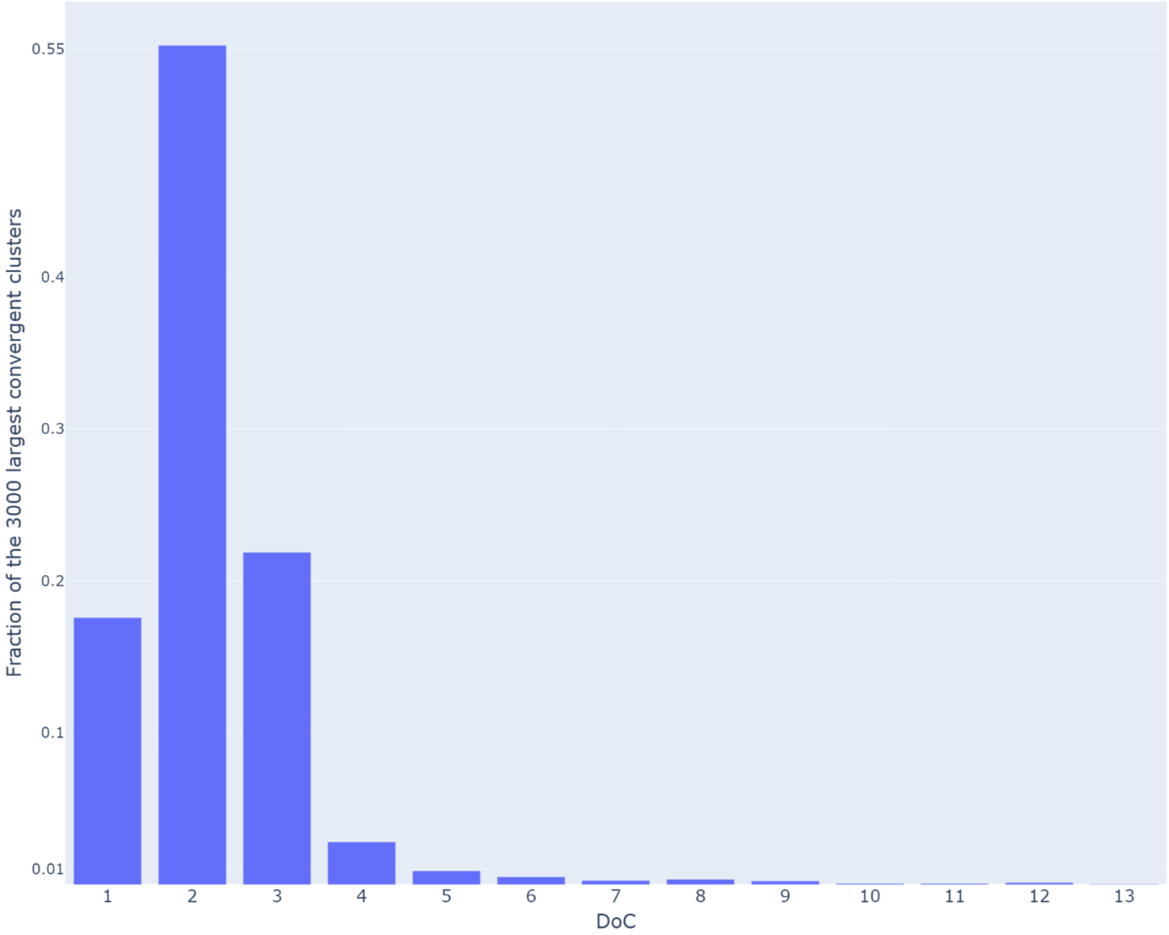


***Figure 3-8 Top-3000 Convergent cluster Size Distribution Over degrees of Commonality**. The 3000 largest convergent clusters were selected, and the fractions of these groups w.r.t. DoCs were plotted. As with Figure 3-13, the fract*ion distribution f*ollows the shape of the shifted Gumbel distribution (p-value: 6.62x10^{-9}), and evidently, even as we increase the number of largest Convergent Clusters included, only the lower-DoC Convergent Clusters are most abundant. In contrast, with increasing DoC, Convergent Clusters become more sparse.*

A second analysis was performed focusing on fully public clonotypes (i.e. VDJ clonotypes observed in all 13 subjects), of which there are 212. These convergent clusters were then ranked based on their total size (i.e. summed across all subjects), and two values plotted for each: the median frequency of sequences belonging to that group for all subjects; and the minimum frequency within any subject. *Figure 3-9 shows that* most convergent clusters have low median and very low minimum frequencies.

Taking these results together, it is clear that high-frequency clones are generally not widely shared, and shared clones typically occur at low frequency within most individuals. To the extent that there may be some correlation between Convergent Cluster size and DoC, it does not appear to be helpful for identifying widely shared clonotypes.

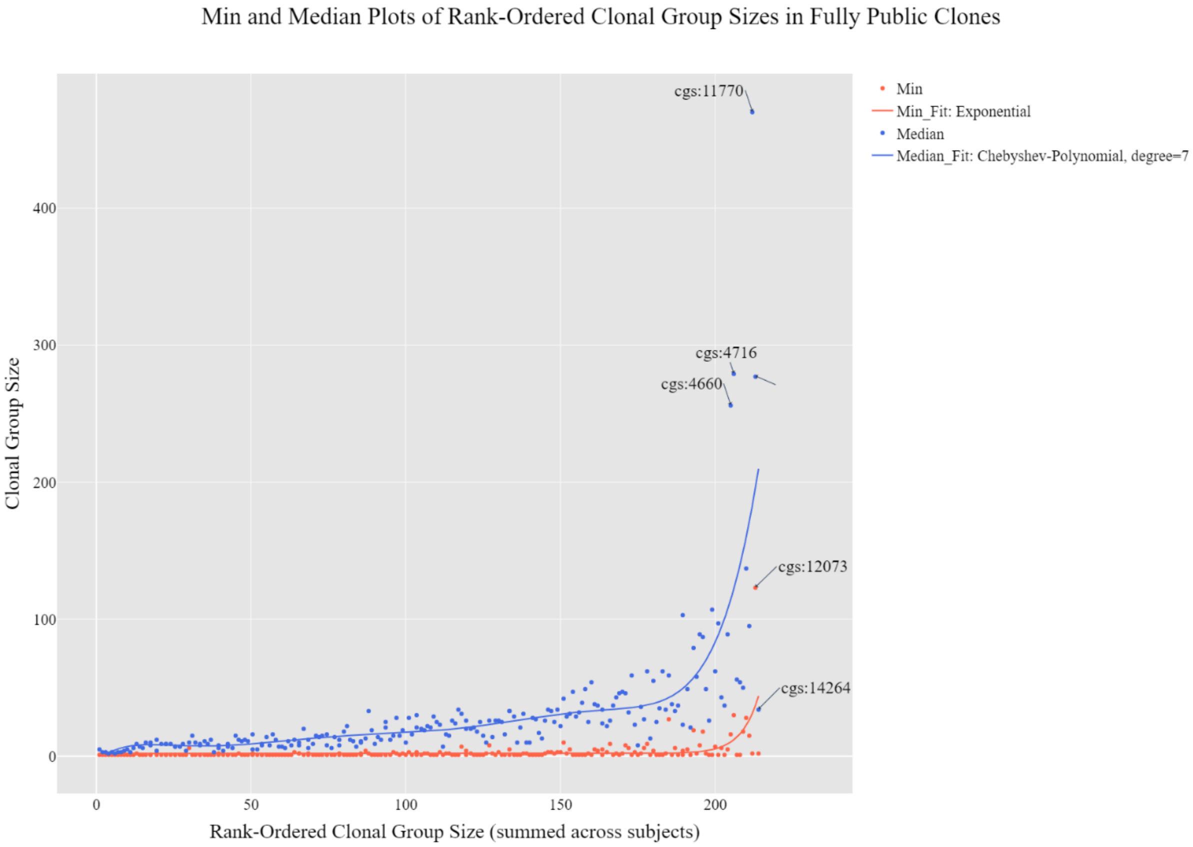


Figure 3-9 Minimum and Median values of rank-ordered sum of Convergent cluster Sizes in Fully public Clones. *The clona-groups of the class DoC-13, i.e. the fully-public class, were ranked in terms of the sum of the convergent cluster size across all 13 subjects. The minimum and median convergent cluster sizes per group were then plotted against the convergent cluster size mean. This shows that even in the fully-public class most convergent clusters have members (i.e. subject representatives) with very small frequencies, therefore, making it impossible to determine how common a clonotype is in a population only based on its frequency in a sample.*

3.3.4 Weighted and Unweighted Multi-Class Models Vs. Multi-Label Model

One way of modelling DoC is as a categorical classification task whereby there are multiple classes instead of two in the binary classification task. In this case, DoCs can be treated as dependent/ordinal/independent classes, depending on the model type and architecture. As a natural progression following Binary Classification (Section 3.3.2), Multi-class (one-versus-all) classification was used as the first approach, a form of categorical classification whereby classes are typically independent of each other. Labels passed into categorical classification models are almost invariably one-hot-encoded, simply the binarization of categorical labels into a sparse 1-dimensional vector of size n, where n is the number of label categories within the dataset. In the case of DoC one-hot-encoding, all values in the one-hot-encoded vector are equal to zero, except the index corresponding to the DoC value, which would be assigned one, as summarised in *Equation 3-1*.

Under the assumption of independence, labels are explicitly one-hot-encoded, usually without consideration for the order of the vector indices, and when using TensorFlow, the `categorical_crossentropy` loss function is used as the loss function. It is also possible to integer-encode the labels passed to a TensorFlow categorical classification model. However, these labels would be automatically converted to one-hot representations before calculating the `sparse_categorical_crossentropy` loss, whereby each integer label is mapped to the corresponding index of the one-hot vector to preserve and allow learning of the ordinal relationships. As an example, the integer-encoded label vector $[1, 2, 3]$ would be one-hot-encoded to $[[1, 0, 0], [0, 1, 0], [0, 0, 1]]$. In fact, the only difference between TensorFlow's `categorical_crossentropy` and `sparse_categorical_crossentropy` is the execution of this transformation process in the latter's internal workings. It is essential to note that, although binarised labels are passed to our multi-class classification models, indices of these one-hot-representations follow the ordinal order of DoC values, thus, rendering these models utterly interchangeable with equivalent "pseudo ordinal" classification models, for which integer labels are used.

However, setting aside the difference in the data structures and transformation of input labels, the two TensorFlow loss functions behave identically. This model is effectively identical to a comparable ordinal classification model discussed above and, as previously noted, learns the ordinality across the classes. A practical feature of multi-class modelling is that the model predictions are a one-dimensional vector of class probabilities.

The necessity for binarised representation is a result of the nature of multi-class classification, whereby the model is expected to produce as many outputs as there are label categories. Therefore, all categorical models' output layers contain as many neurons as the number of classes, each producing a numerical value for their represented class, namely a logit. Another distinguishing element of multi-class modelling with respect to non-probabilistic models, e.g. ordinary regression models, is the SoftMax normalisation of (see *Equation 3-5*) of the logits into a dense vector of discrete prediction-probability distribution over the classes before calculation of the Cross-Entropy loss (as seen in section 3.3.2). In other words, the model outputs a vector of probabilities where each probability is the model's confidence about which class the input belongs to, given the training data and model parameters. Therefore it is crucial to note that this is not the ground-truth probability distribution over the number of individuals who potentially share an antibody. The class with the higher probability in the output vector is naturally chosen as the model's prediction:

$$\mathcal{SM}(l_i) = \frac{e^{l_i}}{\sum_j^C e^{l_j}} \qquad (3\text{-}5)$$

where e is the Euler's constant, C is the number of classes, l_i is the label vector and l_j is the prediction vector.

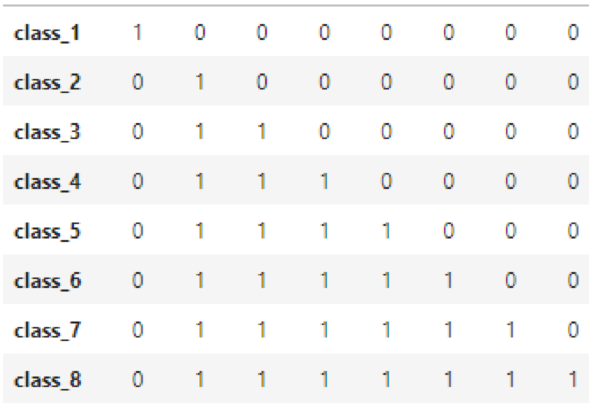
Unlike the binary classification model, the Categorical Cross-Entropy loss function calculates the model's error by calculating the distance between the one-hot-encoded label vector and the output vector. Since the one-hot-encoded label vector is all zeros, except for the index which corresponds to the input’s class, and given SoftMax (see *Equation 3-5*) and cross entropy *(s*ee *Equation 3-4*), the distance/error calculated by SoftMax-cross-entropy is simply the negative log of the SoftMax applied to the prediction for the ground-truth class summed over the mini-batch

$$\mathcal{SMCE}(\theta) = -\sum_{i=1}^{m}[\log(\frac{e^{l_i}}{\sum_{j}^{C} e^{l_j}})] \qquad \text{(3-6)}$$

where C is the vector of indices for classes.

Such a probabilistic model provides a clear advantage by providing the means of examining model calibration, a feature generally crucial for biological/clinical predictive tasks but in particular to a task such as ours where labels suffer from asymmetric noise and have ordinal relationships. Consider the hypothetical data instance with DoC equal to 5 (out of possible DoC range of 1 to 10), for which two hypothetical models, A and B, predict the probability vectors [0.025, 0.025, 0.025, 0.05, 0.585, 0.1, 0.075, 0.065, 0.05] and [0.0, 0.0, 0.025, 0.05, 0.955, 0.05, 0.0075, 0.0025, 0.0] with 0.536 and 0.132 Cross-Entropy error values respectively. Despite correct predictions by both models, at first glance and judging by model error alone, model B strongly outperforms model A, but is model B’s high confidence level warranted? Depending on the data instance, the answer is likely to be no in most cases because of several different ways this data could have been mislabelled due to asymmetric noise inherent to the generative processes giving rise to this data. Some of these underlying processes cause the class imbalance we observe in this data, which demands a joint model confidence calibration to account for label noise and class imbalance. Therefore, for an accurate model to be generalisable, it should not be too confident about its predictions in most cases, in addition to assigning relatively high probabilities to adjacent DoCs. In addition to providing granular predictions suited to this problem, model calibration is more straightforward in classification tasks, resulting in a rich literature on classification model calibration compared to approaches such as regression. As a result, multi-class classification serves as a suitable approach, at least as a first approach, for examining model behaviour, before moving on to more granular regression modelling.

The variations of classification modelling we have addressed so far have suffered from different pitfalls. Though reasonably successful, Binary Classification fails to capture the granularity of that ordinal nature. Conversely, multi-class modelling produces granular predictions but does not have explicit assumptions about data ordinality. Moreover, one could argue that treating DoC prediction as a classification might introduce a type of label correlation, whereby instances with DoC values higher than one would be members of every class in the range $[1, n]$, where n is the groud-truth DoC. Under this assumption, DoC classes should not be treated as independent, which calls for multi-label classification modelling as one possible way, which also provides an alternative approach for capturing the ordinality in DoC. Given the ordinal nature of the classes, for an example sequence with DoC 5, a typical label for a multi-label model would be a one-hot vector where indices 1 to 5 are set to one and all other indices are set to zero. However, it was decided to set the index corresponding to the private class zero for all public classes (see *Table 3-5*) for tw*o r*easons. Firstly, this provides a hard separation between the public and private classes. Secondly, it prevents the massive majority size of the private class from causing biased predictions and model laziness, as with such overtly imbalanced data, deep neural networks almost invariably resort to memorising the majority class’ features while successfully achieving low error rates.



class_1	1	0	0	0	0	0	0	0
class_2	0	1	0	0	0	0	0	0
class_3	0	1	1	0	0	0	0	0
class_4	0	1	1	1	0	0	0	0
class_5	0	1	1	1	1	0	0	0
class_6	0	1	1	1	1	1	0	0
class_7	0	1	1	1	1	1	1	0
class_8	0	1	1	1	1	1	1	1

Table 3-5 Multi-label model’s labels. Each row here represents the label vector of the corresponding DoC.

Besides the unique labels, the multi-label model is identical to the multi-label model, including the number of output neurons, except in applying Sigmoid normalisation (instead of SofMax) to the output logits before calculating the Cross-Entropy loss. As a result, each probability is [0, 1] bound as opposed to the multi-class case, where probabilities sum up to one across the whole output vector.

A bespoke neural network architecture was developed for use in this section, namely the SE-TCN, which is also used in some of the subsequent sections for evaluating the different modelling approaches fairly and consistently. Although recurrent neural networks are typically the preferred choice for sequential data, to avoid the potential vanishing-exploding gradient problem and slow training due to the sequential nature of their forward-pass, TCN were adopted as the baseline architecture. The defining characteristic of TCN is the dilated nature of convolution operations, whereby each convolution kernel can skip a specified number of features between every convolution operation, as demonstrated in *Figure 3-10. The TCN b*lock of the architecture is particularly inspired by the WaveNet architecture[209], though different in parts. For instance, "same" padding was used, which makes the convolution operations bidirectional, instead of the unidirectional operations in WaveNet[209], which uses causal padding.

Additionally, kernel strides equal to one were used for the convolution operations instead of two. Inspiration was taken from squeeze-and-excitation (SE) residual networks[210], an SE block was added downstream of the TCN block, and residual/skip connections were used. The SE block effectively acts as a representation learning module, or an embedding, whereby the model learns the abstract features shaping the data manifold. This is achieved by reducing the dimensionality of the data and then projecting back to the original dimensions, which allows the model to learn the most important features of the data by exploiting the features which allow the model to compress and decompress the data most effectively. A Sigmoid function is applied to the de-compressed tensor to effectively turn off the neurons with less important features and turn on the neurons with important features in a binary manner. This tensor then multiples the output of the TCN block (the input to the SE block) to only preserve the useful features from the convolution operation. To prevent the vanishing gradient problem, as a consequence of the depth of the network, a residual/skip connection was added, which is simply a convolution operation with kernel size and stride of one, to the output of SE block by tensor addition. These connections allow preservation of gradient by allowing their direct flow through the network rather than passing through non-linear activation functions. Batch normalisation was used after all convolution layers, this being a regularisation technique that is used for making the learning process faster and more stable. Additionally, heavy dropout regularisation was used wherever possible to prevent overfitting of the model. This technique helps by eliminating neurons randomly and *ipso facto* preventing over-reliance on certain features and memorisation by the model. More detailed hyper-parameter and model specifications are shown in *Figure 3-11*.

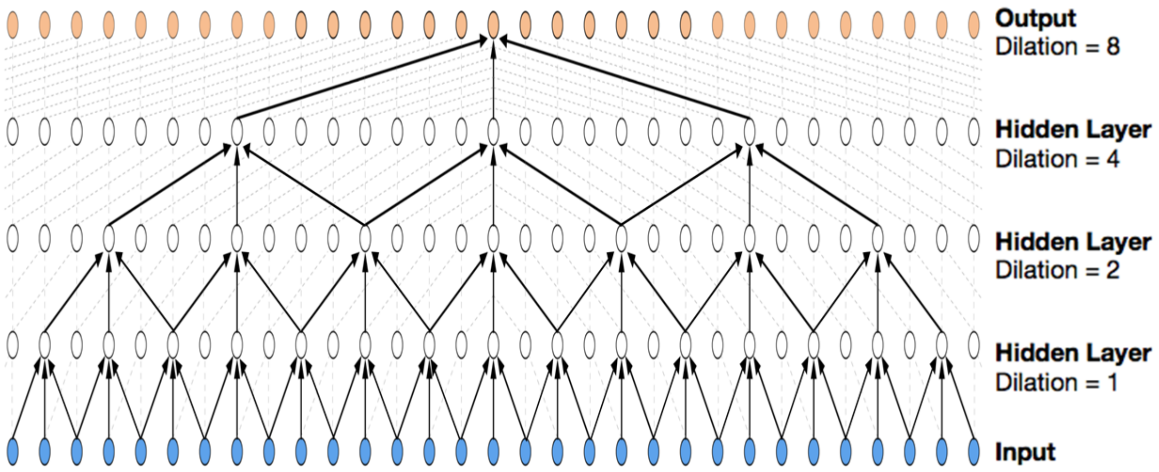


Figure 3-***10 Schematic representation of the temporal convolutional block.*** *In this given example, only the connections of one output neuron (the one pointed to by the arrows) are traced all the way back to the input features to demonstrate the functionality of dilated convolution and the extent of its receptive fields. Each temporal convolution layer is given a fixed kernel size (3 in this example), but varying dilation factors, unlike the common residual blocks with standard convolution layers. This results in "patchy" receptive fields covering a wider span of input features from the previous layer, while only connecting to the same number of neurons defined by the kernel size. In other words, each neuron in the hidden layers with a dilation factor greater than one skips a number of features equal to the value of the dilation factor, forming a sparse connectivity with the previous layer's neurons, or input features (in the case of the 1st hidden layer). For instance, tracing back the connections of the example output neuron, we can see that it is connected to three neurons from the final hidden layer, as specified by the kernel size, but, seven neurons are skipped between each of those three neurons, specified by the dilation factor 8. Tracing back further, we'll notice that the number of neurons skipped is equal to the dilation factor minus one. Consequently, as the data passes through the block, features with increasing distances from each other are processed, resulting in a large coverage of the original input features.*

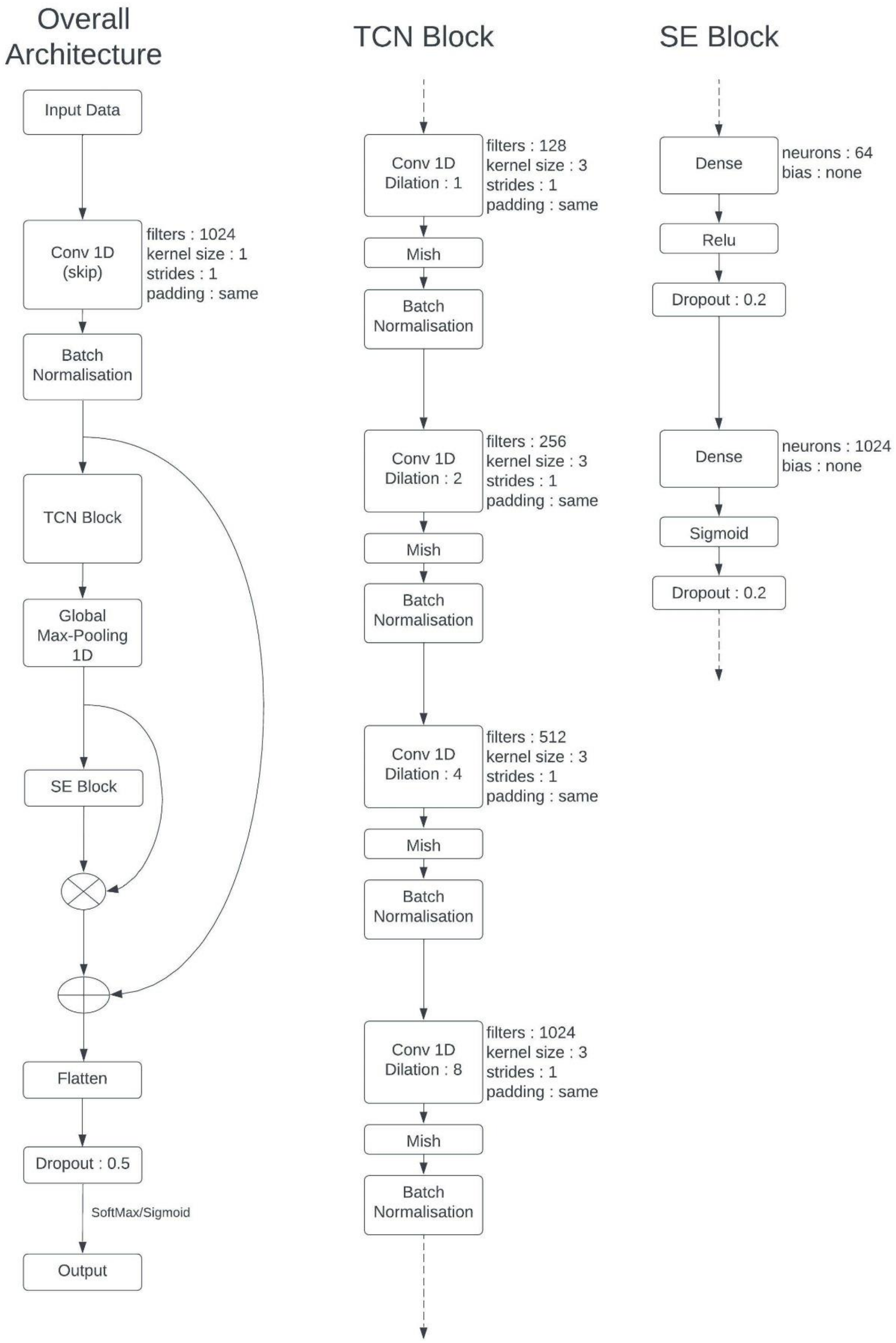


Figure 3-11 The overview of the SE-TCN model.

The SE-TCN architecture was used within the multi-class and multi-label modelling paradigms. Additionally, an imbalance-aware version of the multi-class model was also implemented, which weights the error calculation by the ratio of label frequency to the total data.

As seen in *Figure 3-12*, *The multi*-class approach significantly outperformed the multi-label approach in terms of the AUC metric. Notably, but unsurprisingly given the extreme imbalance in the DoC sizes, the imbalance-aware multi-class model outperforms the other two models by a large margin. Furthermore, the weighted multi-class model has a much faster convergence time than the unweighted multi-class model, which is likely to be due to a smoother error landscape.

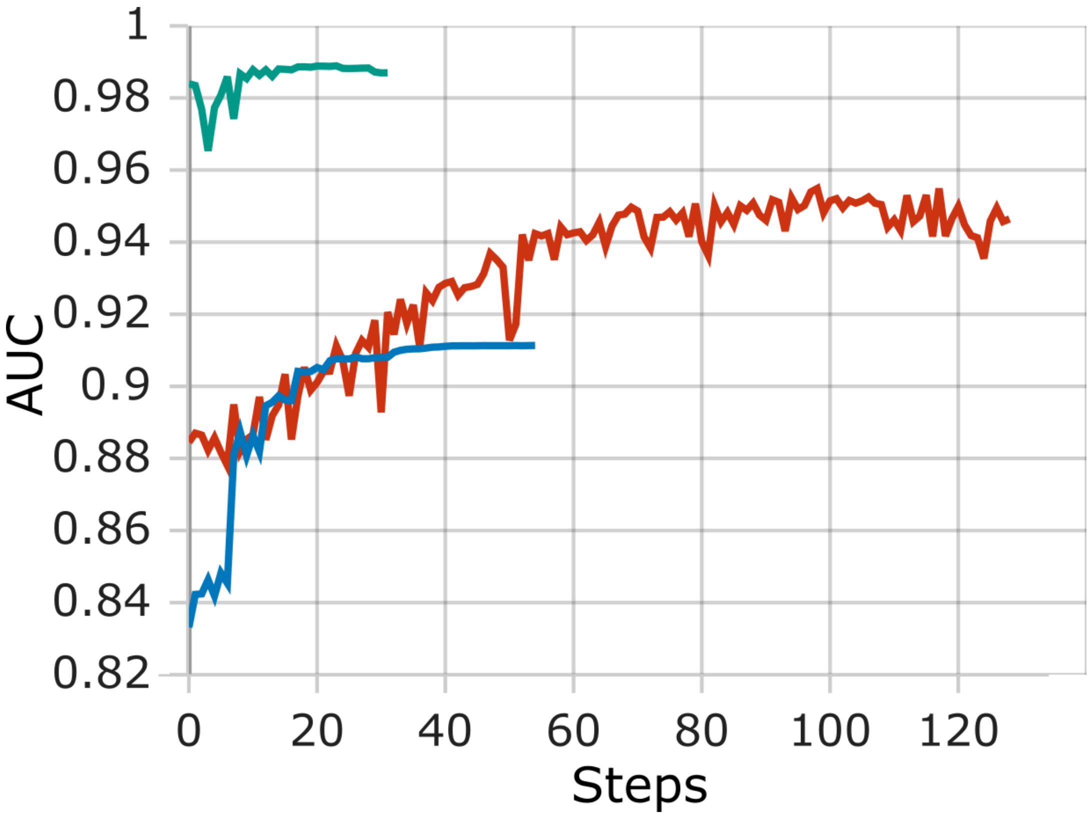


Figure 3-12 AUC value of the validation sets of the unweighted multi-label, weighted and unweighted categorical models.

Here we evaluate the performance of the unweighted multi-label model (blue line), weighted categorical model (green line) *and unweighted categorical model (red line) models. Though the unweighted multi-label model performs closely to the unweighted categorical model, the weighted categorical model by far and away outperforms both of the other two models.*

Whereas the performances of the weighted and unweighted multi-class models are consistent in terms of loss, the loss value of the multi-label model is much lower and unchanging, as shown in *Figure 3-13.* The CMs in *Figure 3-14 help to e*xplain this behaviour by showing how the multi-label model over-predicts the lower DoC labels at the expense of most other DoCs, which is attributable to the large imbalance in favour of the lower DoC classes and the model being unweighted. Whilst this is also the case for the unweighted multi-class model, due to the uni-directional inter-dependence of the labels in the multi-label model, provided the model can differentiate the lower DoCs reasonably well, it can afford to predict indiscriminately the probabilities of higher than 0.5 for all lower DoC output neurons even when the ground-truth is a high DoC, and still achieve a low loss value.

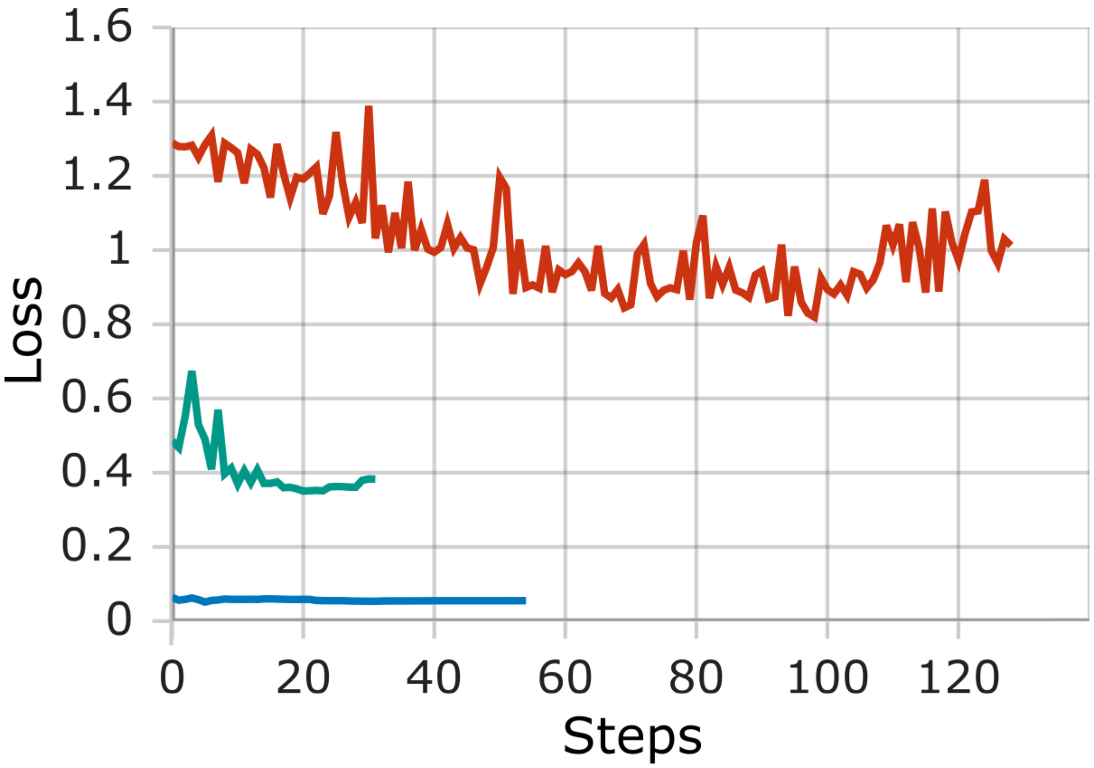


Figure 3-13 Cross-entropy loss of the validation sets of the unweighted multi-label, weighted and unweighted categorical models. *Here we evaluate the performance of the unweighted multi-label model (blue line), weighted categorical models (green line) and unweighted categorical models (red line) models by their respective cross entropy loss values. Though the unweighted multi-label model Shows the smallest loss value, it should be noted that a multi-label model has a different scale*

of cross entropy loss and therefore is not directly comparable. The weighted categorical model here also outperforms the unweighted categorical model.

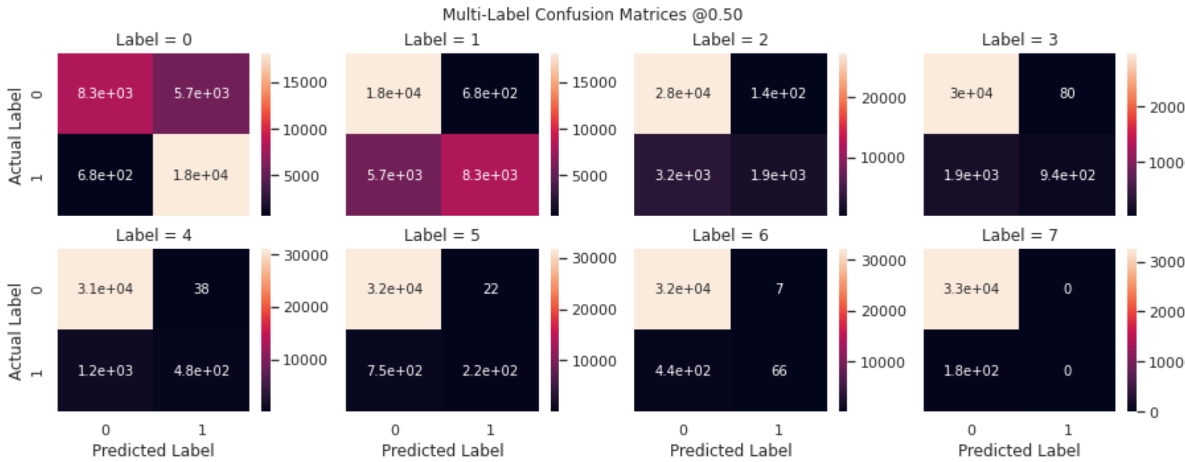
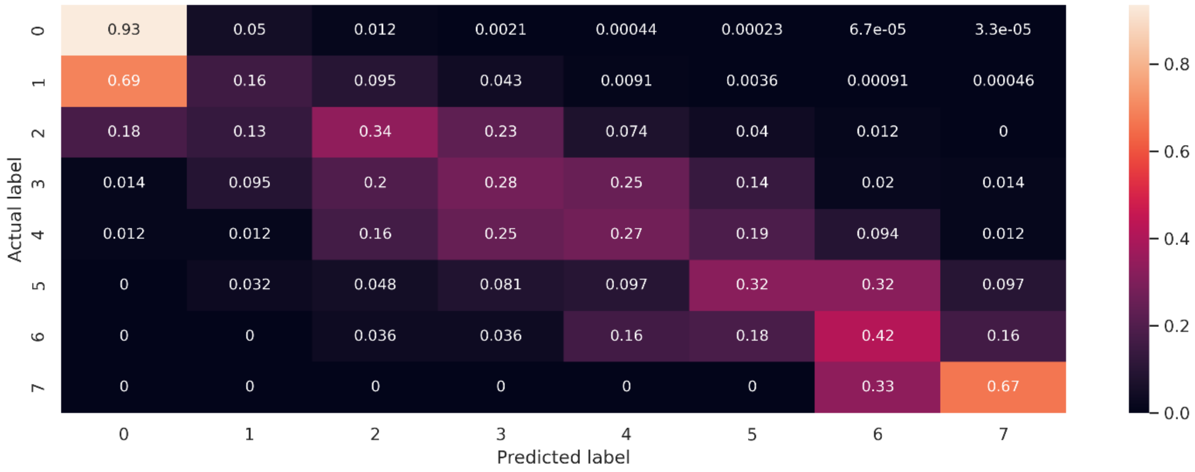


***Figure 3-14 Multi-label test-set confusion matrices.** Here every block represents a binary confusion matrix per DoC, where zero corresponds to the absence of the input in the DoC class and one corresponds to the presence of the input in the DoC class. We see that the model is underperforming for all classes except for the first two classes, clarifying that the multi-label model is ot performing well at all, its AUC and loss performance can be attributed to only the majority classes, and that, a confusion matrix much more clearly summarises model performance over our class-imbalanced data. Mathews Correlation Coefficients: label-0=0.62, label-1=0.62, label-2=0.55, label-3=0.4, label-4=0.5, label-5=0.49, label-6=0.34, label-7=0.05.*

In fact, by taking a closer look at the confusion matrices of the two multi-class models, we can see that the unweighted model is extremely overconfident about the private class at the expense of other classes, in particular DoC$_2$, which is itself informative. The biased false-positive of DoC$_1$ at the expense of DoC$_2$, compared to other classes, shows that the model is learning the ordinal nature of the data, despite its large uncertainty about DoC$_1$ versus DoC$_2$, which results in a relatively bad performance overall. This is further supported by the approximate diagonal of the CM in *Figure 3*-15, which indicates that misclassifications rarely fall far from the ground truth, although ordinality is not explicitly represented in the labels of the multi-class models. When interpreting the confusion matrices, it is crucial to note that the values printed on each cell are simply the fraction of the instances predicted as the class corresponding to the cell (the column), out of all possible classes, for every ground truth DoC (the rows). In other words, the values in each row add up to one. For example, the first row of the confusion matrix in *Figure 3-15 shows that* the model classified approximately 93%, 5%, 1.2%, 0.21%, 0.04%, 0.02%, 0.007%, 0.003% of all test-set instances of private antibodies as DoC classes 1 to 13 respectively.



Actual label \ Predicted label	0	1	2	3	4	5	6	7
0	0.93	0.05	0.012	0.0021	0.00044	0.00023	6.7e-05	3.3e-05
1	0.69	0.16	0.095	0.043	0.0091	0.0036	0.00091	0.00046
2	0.18	0.13	0.34	0.23	0.074	0.04	0.012	0
3	0.014	0.095	0.2	0.28	0.25	0.14	0.02	0.014
4	0.012	0.012	0.16	0.25	0.27	0.19	0.094	0.012
5	0	0.032	0.048	0.081	0.097	0.32	0.32	0.097
6	0	0	0.036	0.036	0.16	0.18	0.42	0.16
7	0	0	0	0	0	0	0.33	0.67

Figure 3-15 Unweighted multi-class model*’s confusion matrix results of the test set.* *In a multi-class confusion matrix, the top-left-bottom-right diagonal is informative of how the model performs, i.e. the higher the concentration of predictions on that diagonal the better the performance of the model. This model, depite of some deviation from the diagonal, still demonstrates a good performance with the ability to learn the ordinal relationship among the classes, as the majority of misclassifications fall close to the diagonal.*

Unsurprisingly, the CM (*Figure 3-16*) of the *we*ighted multi-class model is significantly better than those of the other models. *Figure 3-16 shows that* the model can still implicitly learn ordinality across classes, as the model’s errors are densely concentrated around the ground-truth. In addition to having a clearer diagonal, the performance for DoC$_2$ is significantly better, although there remains a large uncertainty between the DoC$_1$ and DoC$_2$. It is worth noting that DoC$_1$, the largest class, is likely to contain many misleadingly-labelled examples owing to the small cohort size. Furthermore, the performance improves towards larger DoCs, which is potentially more useful for practical applications.

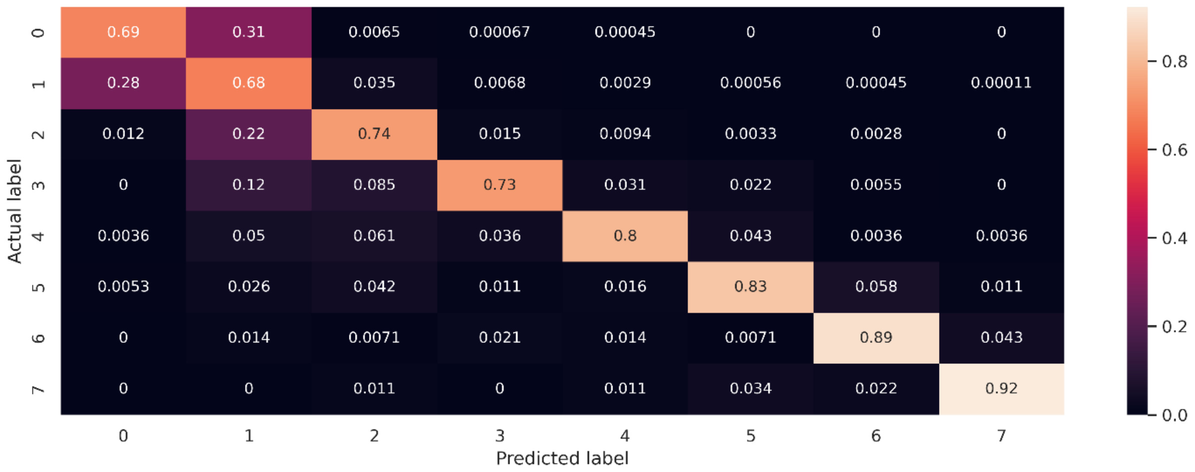


Figure 3-16 *Weighted multi-class model's confusion matrix results of the test set.* The *superiority of the weighted* model *over other models is clearly demonstrated here, as the model demonstrates a clear diagonal as well as misclassifications always falling very closely to the diagonal*.

3.3.5 SE-TCN with Label-Smoothing Regularisation

Typically, "hard labels", e.g. binarised one-hot-encoded labels, used in conjunction with common loss functions such as Cross-Entropy, help ML models learn effective decision boundaries within the latent space that maximise class margins, which in theory, should result in a generalisable model. However, recent research has demonstrated that this may not be the case when dealing with datasets with noisy labels, and that hard labels may, in fact, harm model performance and generalisability [211,212]. Subsequently, they demonstrate that smoothing labels can rescue model generalisability by acting as a form of regularisation for label noise [211,212]. In theory, Label-smoothing relaxes the margins of decision boundaries by softening the error penalisation for ambiguous/noisy examples, enabling the model to learn a manifold of the feature space with larger class margins than that of a manifold learnt from training on hard labels. Additionally, this should help with model calibration by softening the difference in predicted class probabilities when working with noisy labels. As stated in section 3.3.4, model calibration refers to coherence between the average probability outputs of the model for the ground-truth classes and the accuracy of the model [211,212]. As previously noted, when modelling this research problem as a classification task, neither label purity nor an axiomatic knowledge of label dependence/independence can be expected, which provided a secondary line of reasoning in favour of label smoothing. Nonetheless, the incorporation of label smoothing, compared to other regularisation techniques, requires a careful and domain-specific examination of the data, highlighting the importance of the results in previous subsections. In particular, the results in subsection 3.3.5 are incorporated into a specific form of label smoothing.

In previous work, implementation of label-smoothing is done by using vectors of "soft labels" instead of one-hot-encodings[211,212]:

$$ls(y) = (1 - \alpha)y_{hot} + \alpha/K \qquad (3\text{-}7)$$

Where y_{hot} is the one-hot-encoded vector, α is the smoothing constant, and K is the number of classes. This equation uniformly assigns the $\frac{\alpha}{k}$ "smoothing factor" to all candidate labels and subtracts α from the ground-truth label, so the vector still sums to one. While effective for dealing with labels with symmetric noise, uniform smoothing ignores the nature of the noise specific to our immunological data observed in sections 3.3.2 and 3.3.5. In this dataset, label noise is highly directional and asymmetric, i.e. far more instances currently labelled with the lower-end of DoC values are highly likely to have underestimated DoC values than the other way around. Moreover, once a sequence is observed in n individuals in our datasets, obviously, it is guaranteed to have a DoC value no lower than what it is currently assigned.

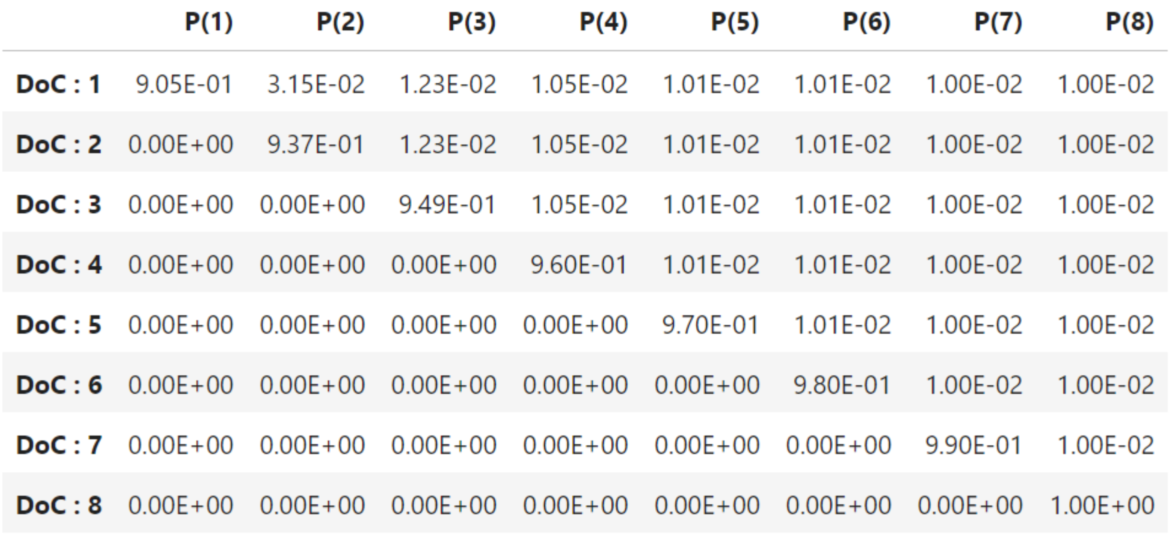
Since our labels have an ordinal relationship and given the truncated power-law distribution of these labels, label smoothing informed by the CDF of the truncated power-law was undertaken, i.e. the value of α was dynamically and differentially decided by the cumulative point-probability of each DoC. This is calculated by:

$$\mathcal{LS}(x) = \begin{cases} 0 & k < x \\ CDF(x) & k = x \\ \sum_{x+1}^{8} PMF(k_i) & k > x \\ 1 - CDF(8) & \lim_{k<j\to\infty} \end{cases} \tag{3-8}$$

where k is the label index corresponding to the DoCs in the same order. Note that since the values of the input vector for SoftMax must add up to one, the smoothing parameter for DoC_8, where the ground-truth label is anything other than DoC_8, is the integral of the area under the curve of the $CDF_{DoC=8}$. Whilst this accounts for mislabelling, due to cohort size, the smoothing factors become vanishingly small to make a difference. Hence, we extended this protocol to incorporate the standard label smoothing technique by introducing an uncertainty constant C, which is a relatively large uncertainty constant uniformly applied to all labels of larger DoCs than the ground-truth label

$$\mathcal{LS}(x) = \begin{cases} 0 & x_i < x \\ CDF(x) - C(k-x) & x_i = x \\ \sum_{x+1}^{k} PMF(k_i) + C & k \geq x_i > x \\ 1 - CDF(k) + C & \lim_{k<j\to\infty} \end{cases} \tag{3-9}$$

With the value for C is chosen as 0.1. Given this formula, the smoothed label vector for every DoC is summarised in *Table 3-6. Despite t*he sound basis, label smoothing proved detrimental to model performance as shown in *Figures 3-17* and *3-18.* The asymmetrical nature of the smoothing – the central feature which in theory should have helped counter the issues arising from the asymmetrical class imbalance - could force the model to be less confident about the instances of lower DoC, i.e. the vast majority of the data. While this might be a desirable behaviour, as *Figure 3-18 shows, it* also results in misclassifying a greater portion of the majority classes' instances (see *Figure 3-16 for compar*ison), which disproportionally increases the model's error. Nonetheless, *Figure 3-18 shows that* the error rate for higher DoC classes is also relatively high, leading to the conclusion that label-smoothing is not appropriate technique for this task.



	P(1)	P(2)	P(3)	P(4)	P(5)	P(6)	P(7)	P(8)
DoC : 1	9.05E-01	3.15E-02	1.23E-02	1.05E-02	1.01E-02	1.01E-02	1.00E-02	1.00E-02
DoC : 2	0.00E+00	9.37E-01	1.23E-02	1.05E-02	1.01E-02	1.01E-02	1.00E-02	1.00E-02
DoC : 3	0.00E+00	0.00E+00	9.49E-01	1.05E-02	1.01E-02	1.01E-02	1.00E-02	1.00E-02
DoC : 4	0.00E+00	0.00E+00	0.00E+00	9.60E-01	1.01E-02	1.01E-02	1.00E-02	1.00E-02
DoC : 5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.70E-01	1.01E-02	1.00E-02	1.00E-02
DoC : 6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.80E-01	1.00E-02	1.00E-02
DoC : 7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.90E-01	1.00E-02
DoC : 8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00

Table 3-6 Smooth labels. Each row here represents the smooth-label vector of the corresponding DoC.

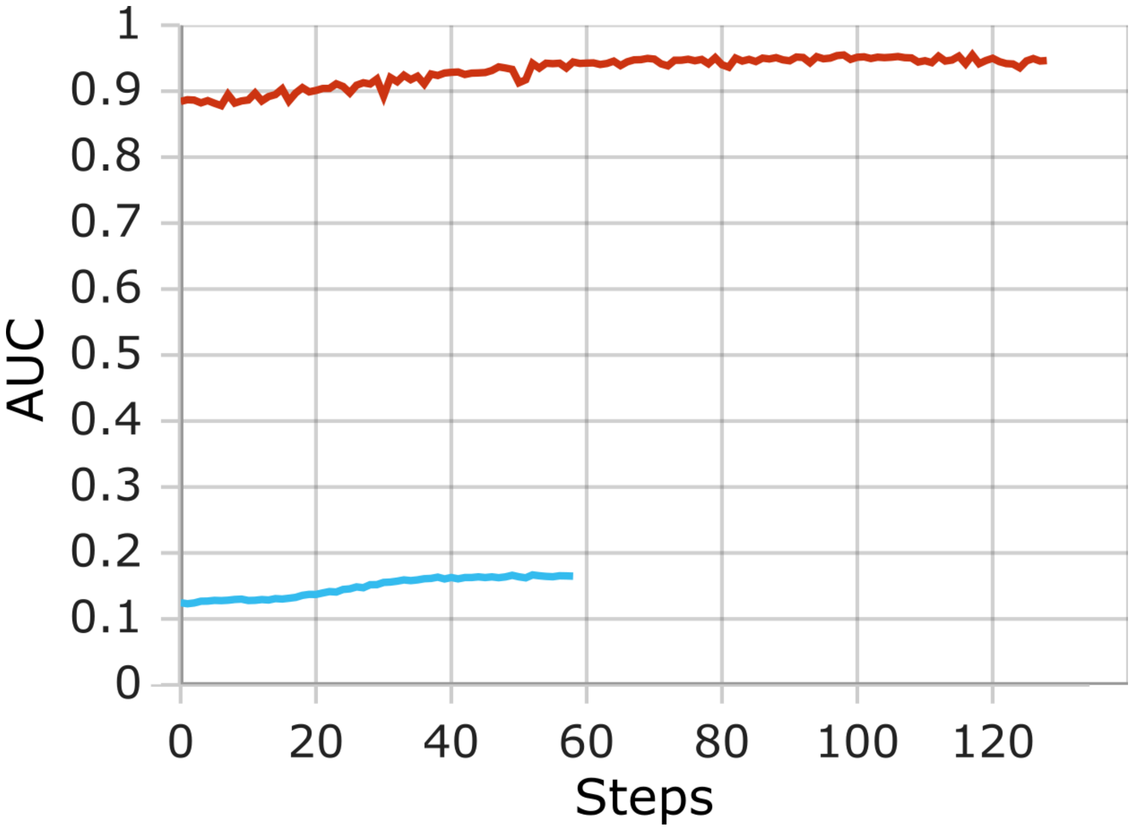
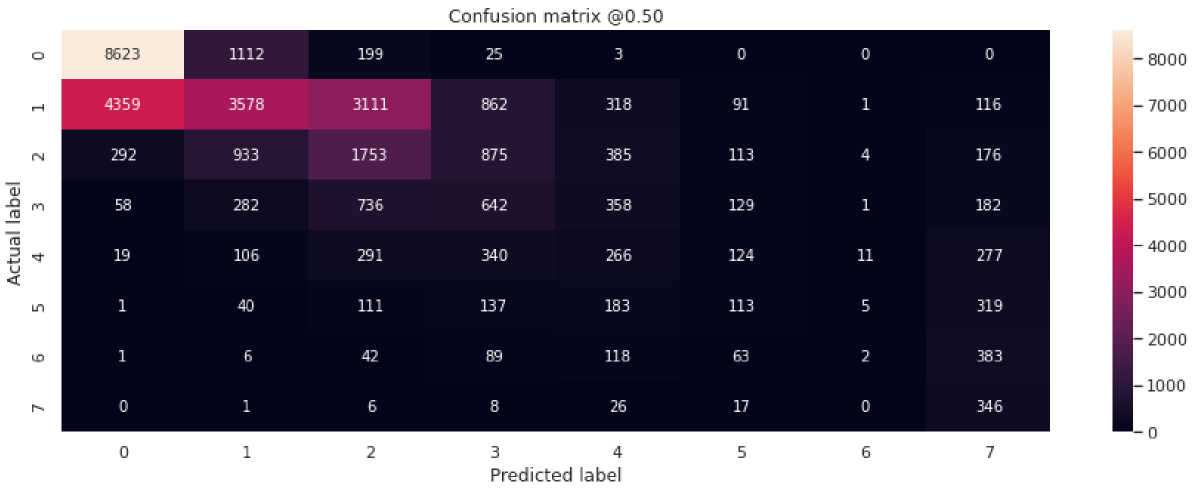


Figure 3-17 Hard-label vs. smooth-label weighted multi-class model validation AUC. *The AUC value for the weighted multi-class model with and without label-smoothing (blue and red lines respectively). As shown, the label-smoothing technique does not help the model at all.*



	0	1	2	3	4	5	6	7
0	8623	1112	199	25	3	0	0	0
1	4359	3578	3111	862	318	91	1	116
2	292	933	1753	875	385	113	4	176
3	58	282	736	642	358	129	1	182
4	19	106	291	340	266	124	11	277
5	1	40	111	137	183	113	5	319
6	1	6	42	89	118	63	2	383
7	0	1	6	8	26	17	0	346

Figure 3-18 Smooth-label model validation confusion matrix. *In further support of Figure 3-17, this confusion matrix shows that the label-smoothing technique does not help the weighted multi-class model when compared to Figure 3-16.*

3.3.6 Ordinal Classification with Weighted Phi Loss Function

The results from the previous chapter (see *Figures 3-17* and *3-18*) demonstrated that using label-smoothing proved detrimental to model performance, likely due to the significant class imbalance. Moreover, all classification approaches used so far do not explicitly model the ordinality of the problem, albeit they demonstrate an implicit learning of the ordinal relationships through the results shown by the confusion matrices. Since the discussed challenges of this dataset, namely the severely noisy and imbalanced labels, are not independent of the ordinal nature, we set out to develop a model that jointly addresses these issues.

While the multi-class error weighting regime used in subsection 3.3.4 worked quite well, it was not specific enough to the domain-specific nature of the labels. To further align the error weighting with the class imbalance, a novel loss function was invented to address the problems with noisy labels and class imbalance, and notably, to model this problem as an explicit ordinal classification task.

First, we took inspiration from the well-established focal loss successfully applied to object detection tasks, where high cross entropy error is calculated even when the model correctly predicts labels, pushing the model towards maximising the probability predictions for the ground truth labels and, as a result, overconfidence[205]. Secondly, we took inspiration from the weighted kappa loss, whereby in the case of ordinal regression tasks, a quadratic weighting is applied to the calculated error for misclassifications depending on how far they are from the ground truth label[213]. Here, we designed a loss function, which unlike those above, weakly penalises the probability predictions marginally below or above a defined threshold whilst applying a polynomial spline function to predicted probabilities, but with a large bias towards predicted probabilities lower DoC indices in the label vector. Together, this would penalise the model if it predicts high probabilities for classes of lower DoC than the ground truth label, especially when the sample is misclassified, whilst being more forgiving towards higher DoC classes, especially when the label is correctly classified. Again, we maintained all experimental conditions, e.g. the neural architecture and parameters, identical.

The design of this loss function was inspired by the successful focal loss used for object detection[205] and the weighted kappa loss function designed for multi-class ordinal regression tasks[213]. Focal loss introduces a term to SoftMax-cross-entropy to reduce the penalty calculated by cross entropy for well-classified examples, as cross entropy tries to maximise the probability output for the ground-truth label to 100%, which is the basis of the mode-overconfidence. Therefore, focal loss diverts the focus of the model from well-classified examples to the misclassified and less confident predictions, i.e. the ones with marginally higher output probabilities than the output for incorrect labels:

$$\mathcal{FL}(pt) = -(1-p_t)^{\gamma} log(p_t) \tag{3-10}$$

Where $-(1-p_t)^{\gamma}$ is the term added to standard cross entropy to create the focal loss, and γ is the focusing parameter. It is also possible to add class weights to this equation to address the imbalance:

$$\mathcal{FL}(pt) = -\alpha t(1-pt)^{\gamma} log(pt) \tag{3-11}$$

Where α is the class-weight. The weighted kappa loss introduces an ordinal weighting parameter ω, which can also be raised to any power (e.g. quadratic) and multiplies the probability and label vectors by this parameter:

$$\kappa = 1 - \frac{\sum_{ij} \omega_{ij} O_{ij}}{\sum_{ij} \omega_{ij} E_{ij}}, \quad where \quad \kappa \in [-1, 1] \tag{3-12}$$

where O is the prediction vector, and E is the label vector. Hence, the loss function which can be minimised, is defined as:

$$L = log(1-\kappa) \ \ where \quad L \in (-\infty, log2] \tag{3-13}$$

where κ is the weighted kappa function. The Phi loss function modifies aspects of the two loss functions to improve the penalisation strategy for noisy long-tailed label distributions, which suffer from decaying asymmetric noise:

$$\varphi = log(\frac{\sum_{ij} \psi_{ij}^2 O_{ij}^\gamma}{\sum_i^n (\psi^2 ((\sum_{j=1}^n O_{ij})^T)^\gamma (\sum_{j=1}^n E_{ij})\alpha)}), \quad where \quad \varphi \in (-\infty, log2] \tag{3-14}$$

where Ψ is a polynomial weighting spline function, α the standard class-weighting based on the class-size proportion and γ the focus parameter. Note that Ψ is a well-defined and continuous bespoke spline designed as:

$$\psi_{i,j} = \begin{cases} hx^2 \ , & x \leq 0 \\ \frac{-2zx\sin(a)\cos(a)+2zw\sin(a)+1-\sqrt{-4kz\sin^2(a)-4zx\sin(a)\cos(a)+4zw\sin(a)+1}}{2z\sin^2(a)} \ ; \quad z\sin^2(a) \neq 0 \ , & x > 0 \end{cases}$$

(3-15)

where a, h, w and z are scaling constants tuned as neural network hyperparameters. The *Phi* loss in combination with the *Psi* weighting parameter heavily penalises mispredictions where a DoC lower than the ground-truth label is predicted, whilst being more lenient on mispredictions of higher DoC than the ground-truth and ignoring predicted probabilities of lower DoC classes if the prediction matches the ground-truth.

Despite being sound in theory and practice, the model with the Phi loss function failed to outperform the model with the SoftMax-cross-entropy (i.e. the weighted multi-class model) in terms of AUC, as shown in *Figure 3-19. Nonethele*ss, the Phi loss function still performed quite well and managed to learn the ordinal nature of the labels (see figures *3-19* to *3-21*).

Given the extreme class imbalance in favour of the private class, the AUC results sometimes can sometimes be misleading. To that end, the AUC was calculated for the public classes only, but again the model with SoftMax-cross-entropy outperformed that with the Phi loss function (see *Figure 3-20*). The C*M in* Figu*re 3-21* also supports the observation that the model with SoftMax-cross-entropy learned the ordinal relationship among the classes better and with less uncertainty.

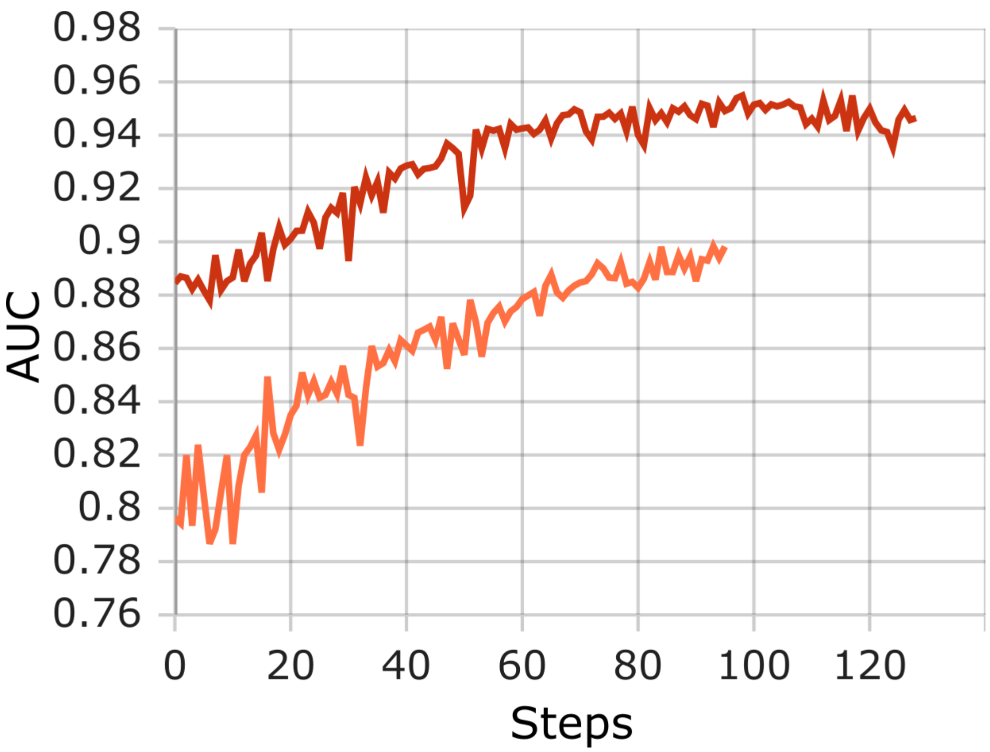


Figure 3-19 Weighted SoftMax-cross-entropy vs. weighted-phi loss multi-class validation AUC. *Comparison of the previously used weighted multi-class model with and without the weighted-phi loss function over the entire validation data.*

Though the model with the phi loss function performs well, it still does not perform as well as the model with cross entropy loss function.

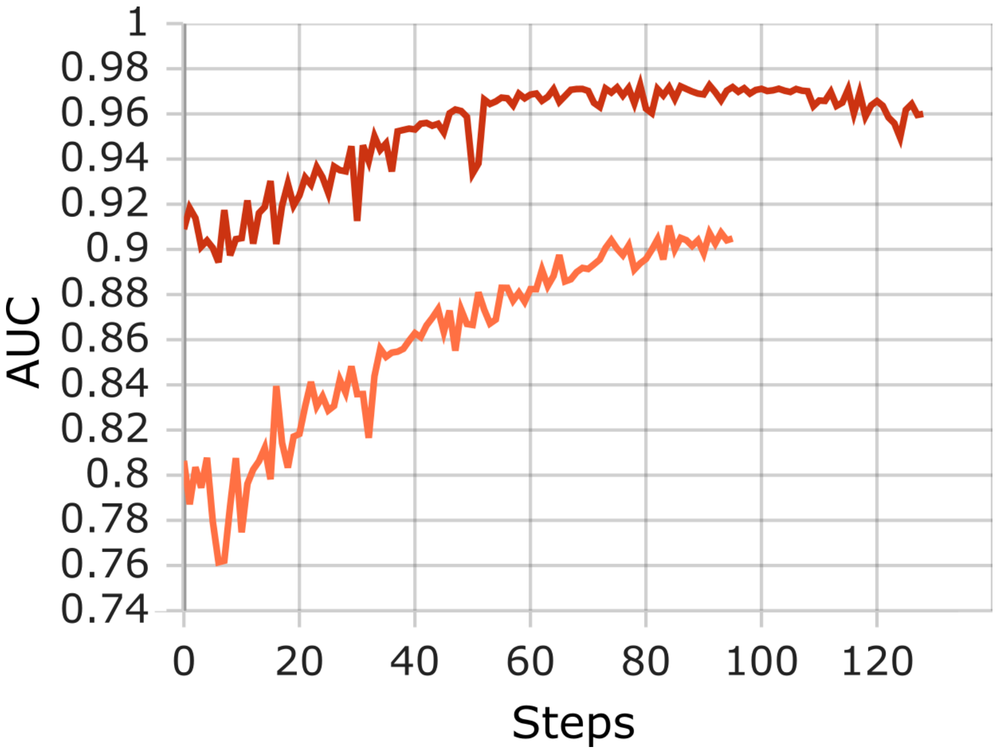
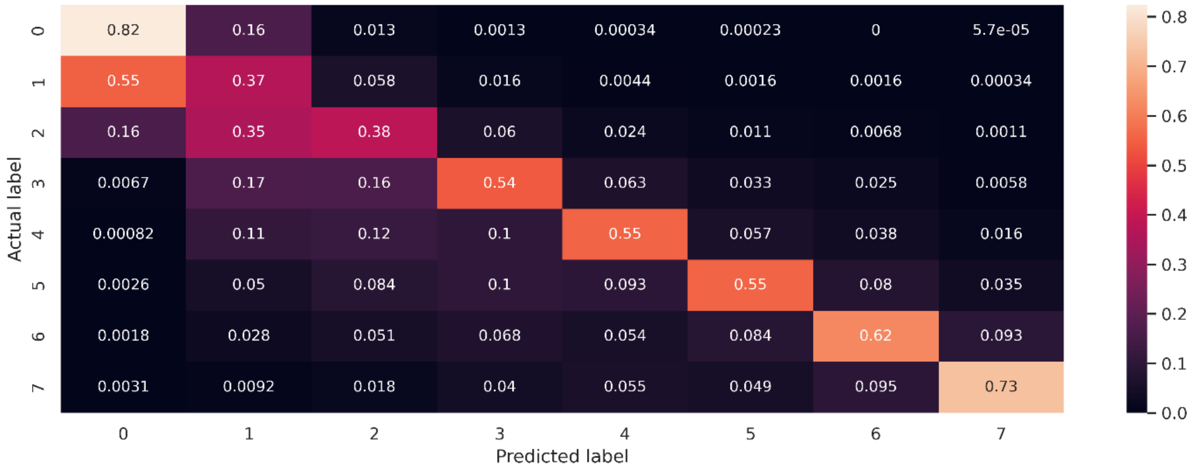


Figure 3-20 Weighted SoftMax-cross-entropy vs. weighted-phi loss multi-class validation AUC. *Comparison of the previously used weighted multi-class model with and without the weighted-phi loss function over the public classes of the validation data. Though the model with the phi loss function performs well, it still does not perform as well as the model with cross entropy loss function.*

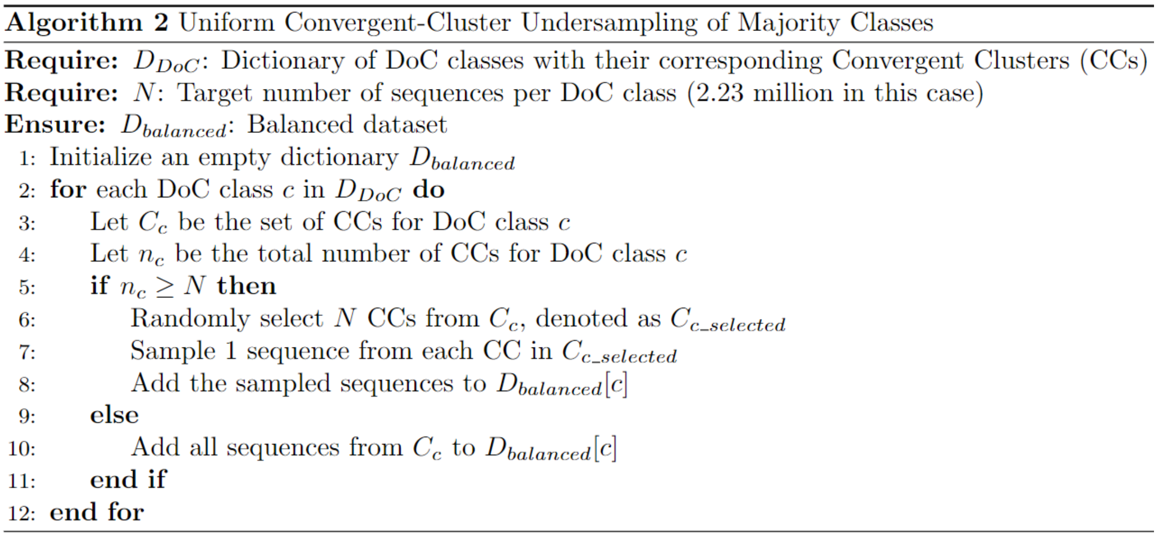


Actual label	0	1	2	3	4	5	6	7
0	0.82	0.16	0.013	0.0013	0.00034	0.00023	0	5.7e-05
1	0.55	0.37	0.058	0.016	0.0044	0.0016	0.0016	0.00034
2	0.16	0.35	0.38	0.06	0.024	0.011	0.0068	0.0011
3	0.0067	0.17	0.16	0.54	0.063	0.033	0.025	0.0058
4	0.00082	0.11	0.12	0.1	0.55	0.057	0.038	0.016
5	0.0026	0.05	0.084	0.1	0.093	0.55	0.08	0.035
6	0.0018	0.028	0.051	0.068	0.054	0.084	0.62	0.093
7	0.0031	0.0092	0.018	0.04	0.055	0.049	0.095	0.73

Figure 3-21 Multi-class model with weighted-phi loss confusion matrix. *As with the previous* figures when w*e compare this figure with Figure 3-16, we can see that this model performs really well, both in terms of the diagonal and close-to-diagonal misclassifications, it still does not perform quite as well as the model with the Cross Entropy loss function.*

3.3.7 Informed Undersampling and Leak-Free Splitting

Training of a neural network with 100s of millions of sequences imposes a huge cost on the time-complexity of the model, and the usage of all this data, in the case of the prediction of DoC, is unjustified. This is because over 99% of the data comes from the first three DoCs, which causes a quick convergence of ML models down deep local minima where models favour overprediction of lower DoCs at the cost of other classes. Taken together, these issues called for measures to be taken to tackle the imbalance problem. Whilst there are well-established techniques to tackle the problem with ordinary imbalanced data[214], when considering biological data such considerations call for case-by-case, bespoke techniques[69,215–217]. To this end, an undersampling approach was devised by drawing a random uniform distribution over convergent clusters to ensure equal representation, which allows the preservation of as much variability of majority classes as possible. Additionally, the minimum subsample size was tuned by minimising the Kullback-Leibler (KL) divergence of the sumrep summary statistics distributions of the subsample and the full dataset supported by *Table 3-7*. Kullback-Leibler divergence quantifies the information loss when one distribution approximates another. Minimising this divergence ensures the subsampled data retains essential characteristics of the original data, thereby providing a robust representation. It is a measure of how one probability distribution diverges from a second, expected probability distribution. This fine-tuning allowed the maximum class size of 2,230,147 sequences, which is the size of the DoC$_4$ class, and was shown to capture the variability of the population sample. To elaborate further, we undersampled DoC$_{1-3}$ classes by sampling 1 sequence from 2,230,147 randomly-sampled convergent clusters for each class without replacement if the total number convergent clusters for the class exceeded 2,230,147, and with replacement if it did not. As seen in *Figure 3-22, by using* this approach a large total, and per-class, sequence size was maintained after undersampling whilst significantly reducing the data imbalance to an acceptable ratio. Furthermore, the variability of the data in terms of the breadth of convergent clusters representation is retained. The undersampling protocol is summarised in *Algorithm 2*.



```
**Algorithm 2** Uniform Convergent-Cluster Undersampling of Majority Classes
**Require:** $D_{DoC}$: Dictionary of DoC classes with their corresponding Convergent Clusters (CCs)
**Require:** $N$: Target number of sequences per DoC class (2.23 million in this case)
**Ensure:** $D_{balanced}$: Balanced dataset
Initialize an empty dictionary $D_{balanced}$
**for** each DoC class $c$ in $D_{DoC}$ **do**
    Let $C_c$ be the set of CCs for DoC class $c$
    Let $n_c$ be the total number of CCs for DoC class $c$
    **if** $n_c \geq N$ **then**
        Randomly select $N$ CCs from $C_c$, denoted as $C_{c\_selected}$
        Sample 1 sequence from each CC in $C_{c\_selected}$
        Add the sampled sequences to $D_{balanced}[c]$
    **else**
        Add all sequences from $C_c$ to $D_{balanced}[c]$
    **end if**
**end for**
```

Algorithm 1 The undersampling algorithm.

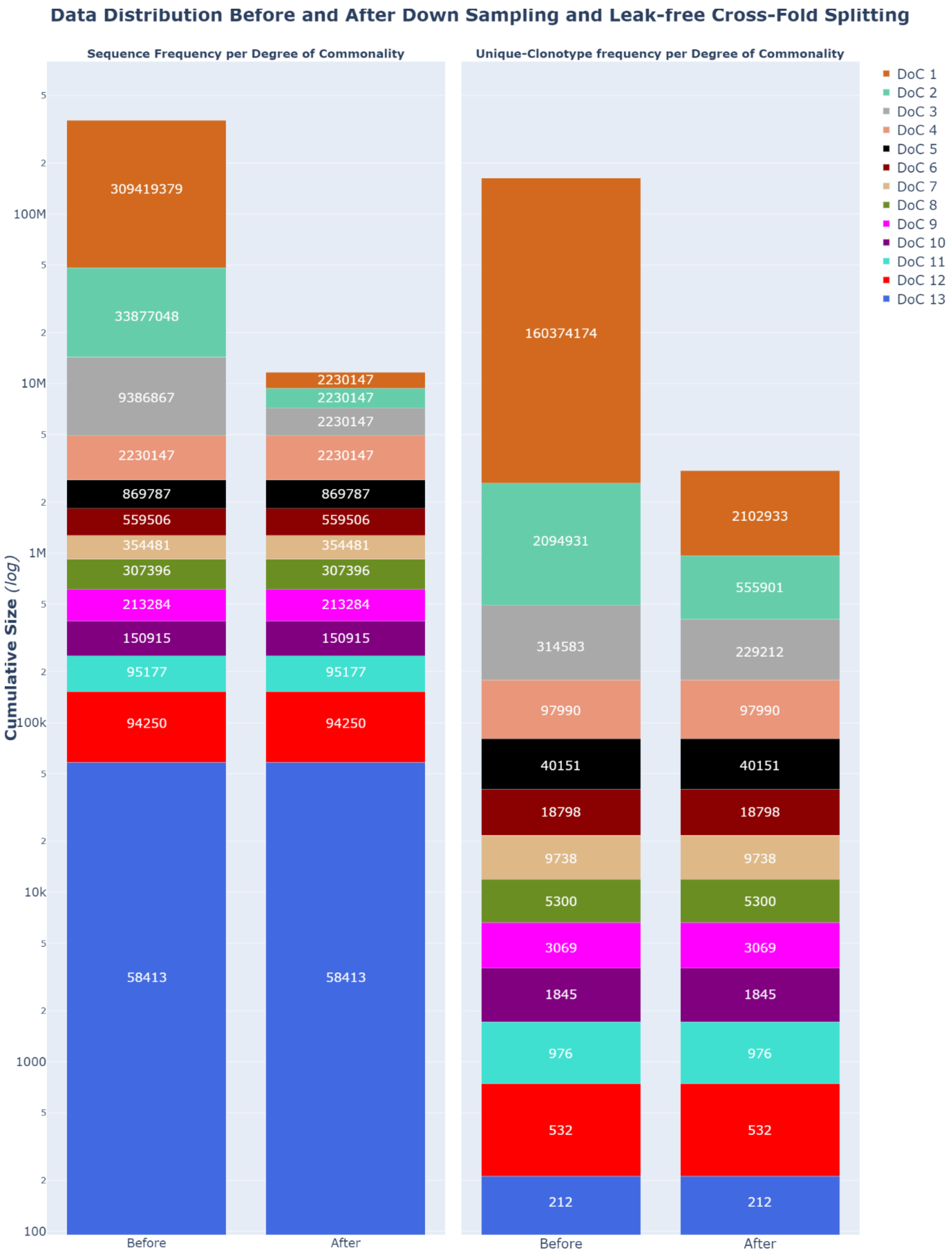
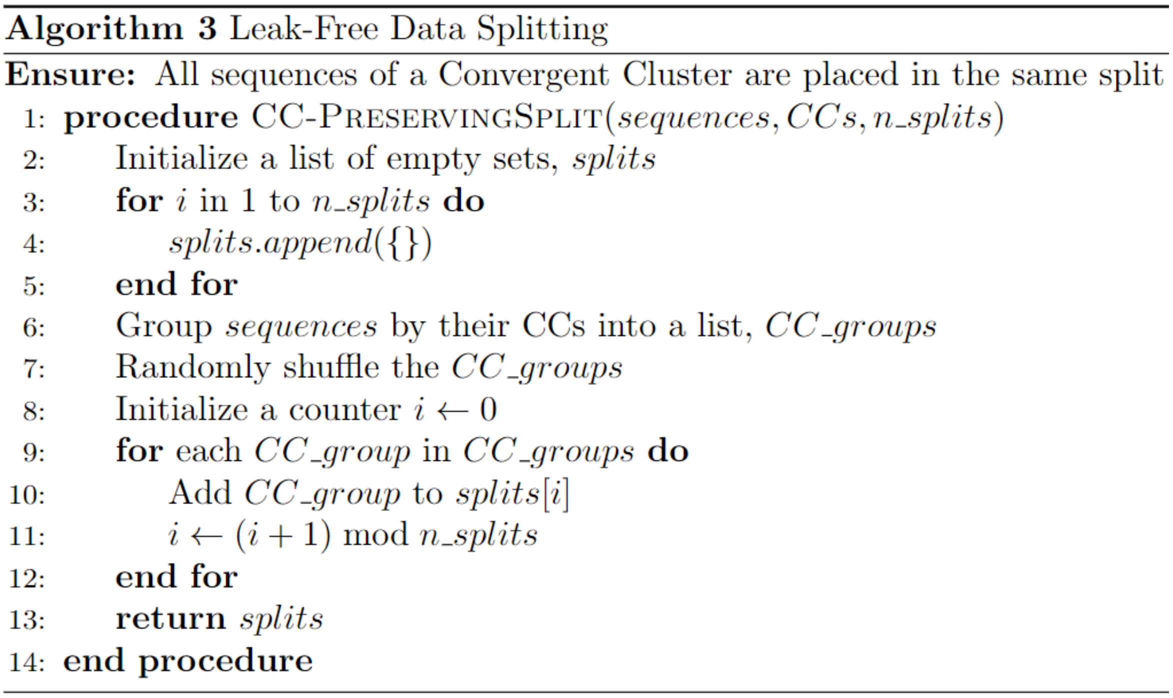


Figure 3-22 Data statistics before and after undersampling. The cummulatice frequencies of sequence and convergent clusters of all DoCs are plotted with a logarithmic Y-axis, where the classes are stacked in the increasing order of size for better visualisation, as given the class-imbalance in our data several classes would be too small to be visible. As can be seen here sequences ans convergent clusters are retained in the smaller minority classes. While the relatively larger minority

classes are aggressively undersampled in terms of their sequences, they still maintain high number of sequences and a very large proportions of their convergent clusters compared to before undersampling.

Another fundamental issue that needed to be addressed is the potential leakage of data from the same convergent clusters from training set to validation and test sets. Despite removing duplicate sequences, all non-redundant sequences from the same convergent cluster were retained. The logic behind this decision is that, although these sequences represent a single convergent cluster/sample, they arguably represent the variance within a given convergent cluster and in effect act as a proxy for data augmentation. Retaining such “organic data-augmentation” represented a useful opportunity, as data augmentation is a well-established technique for improving the performance of deep neural networks[218,219], particularly with respect to the avoidance of overfitting, notably when there is insufficient data and/or class-imbalances[220,221]. The standard approach for splitting data into train, validation and test sets is by random sampling. Data augmentation is usually a synthetic process that is carried out on-the-fly during the training-validation process and applied after the data-splitting. Instead, in the present case, a single convergent cluster is considered a single sample with all its constituent sequences considered “organic data-augmentations” of a theoretical consensus sample. Given that all of these sequences exist in the dataset prior to splitting, if the data is split following the standard protocol, many “data-augmented versions” of the same sample are liable to end up in different splits. To avoid this, a protocol was devised that combines a “leak-free data-splitting” algorithm with 10-fold cross-validation (CV), as summarised in *Algorithm 3-3. In summar*y, this protocol splits the undersampled data into 10 splits of approximately equal size (with respect to the number of sequences), where each split contains all the sequences belonging to every convergent cluster contained within that split. Following a 80%:10%:10% train-validation-test split strategy and 10-fold CV, for every fold a unique set of 8, 1 and 1 splits are chosen for train, validation and train sets respectively. The leak-free data-splitting and 10-fold CV strategy is summarised in *Figure 3-23.*



Algorithm 3 Leak-Free Data Splitting

```
Ensure: All sequences of a Convergent Cluster are placed in the same split
procedure CC-PreservingSplit($sequences, CCs, n\_splits$)
    Initialize a list of empty sets, $splits$
    for $i$ in 1 to $n\_splits$ do
        $splits.append(\{\})$
    end for
    Group $sequences$ by their CCs into a list, $CC\_groups$
    Randomly shuffle the $CC\_groups$
    Initialize a counter $i \leftarrow 0$
    for each $CC\_group$ in $CC\_groups$ do
        Add $CC\_group$ to $splits[i]$
        $i \leftarrow (i + 1) \bmod n\_splits$
    end for
    return $splits$
end procedure
```

Algorithm 2 ***Leak-free Splitting.***

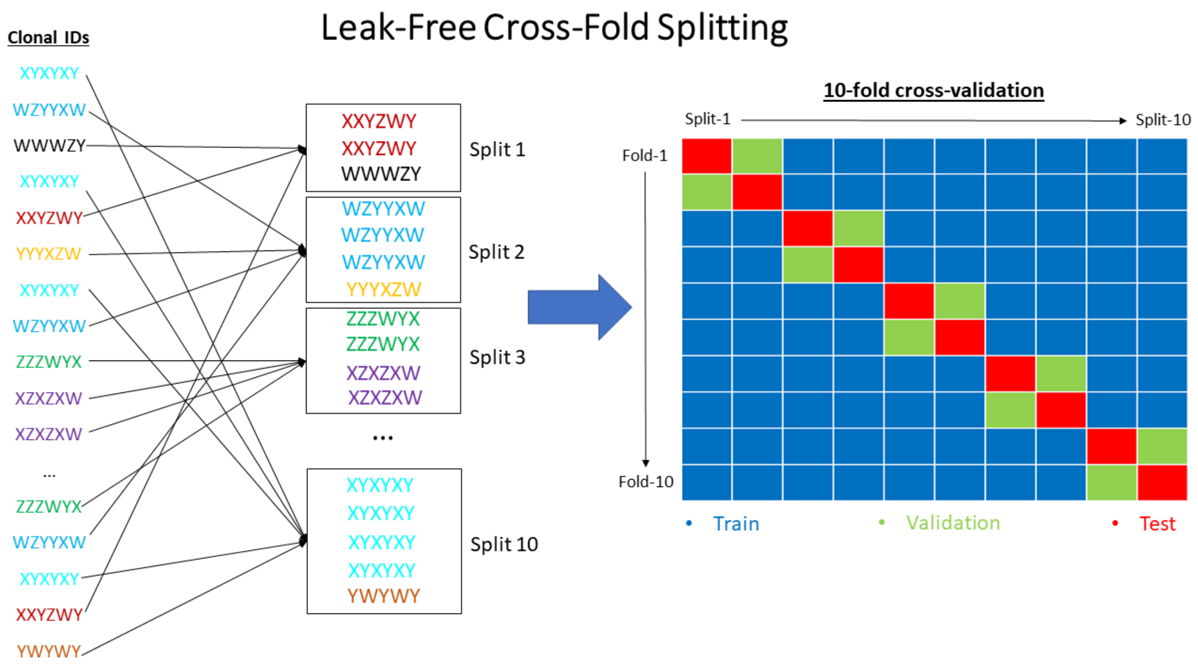


Figure 3-23 ***Leak-free cross-validation cross-fold splitting of the data.*** *To prevent the leakage of data across training, validation and test sets, we devised a leak-free strategy to prevent the leakage whilst creating 10 cross-folds for cross-validation. Each of the 10 splits receives a fair sample from each DoC label, but importantly, all sequences from the same convergent cluster is selected for the same fold.*

To evaluate the impact of introducing the highly principled leak-free data splitting strategy discussed in section 3.3.3, the best weighted multi-class SE-TCN model was retrained with the leak-free cross validation strategy. As anticipated, there was a significant reduction in performance. This is supported by the validation data results in *Figures 3*-24, *3-25* and *3-26*, as the AUC and public-AUC values show a good performance, with a weighted top-3 accuracy of 78%. Additionally, comparing the CM in *Figure 3-16 with that* of the same model with leaky cross-validation in *Figure 3-27 provides f*urther clear evidence for the reduction of the model performance. Nevertheless, *Figure 3-27 demonstrat*es that the model has learned the ordinal relationship among the classes, with most of the misclassifications of the higher DoC classes assigned to neighbouring classes. However, the model also appears to be susceptible to the underlying class imbalance, as there is a distinct tendency towards underestimation of the DoCs, as evident in the skew towards below diagonal squares in the CM (*Figure 3-27*).

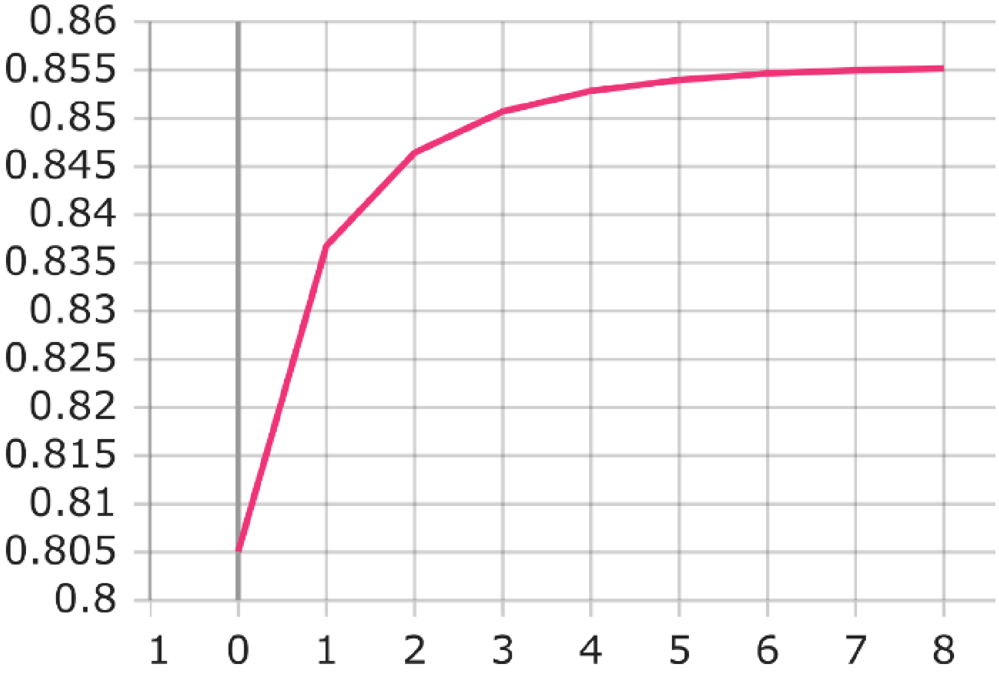


Figure 3-**24 Weighted multi-class post-leak-free cross-fold splitting AUC performance.** *Here, the Y-axis is the AUC and X-axis is the steps of the model training. Although the performance of the best weighted multi-class model with cross entropy* *loss expectedly drops after the leak-free data splitting, when compared to Figure 3-16. However, it still achieves reasonably high AUC performance, indicative of existence of signals that can be learnt from the genomic features of the data.*

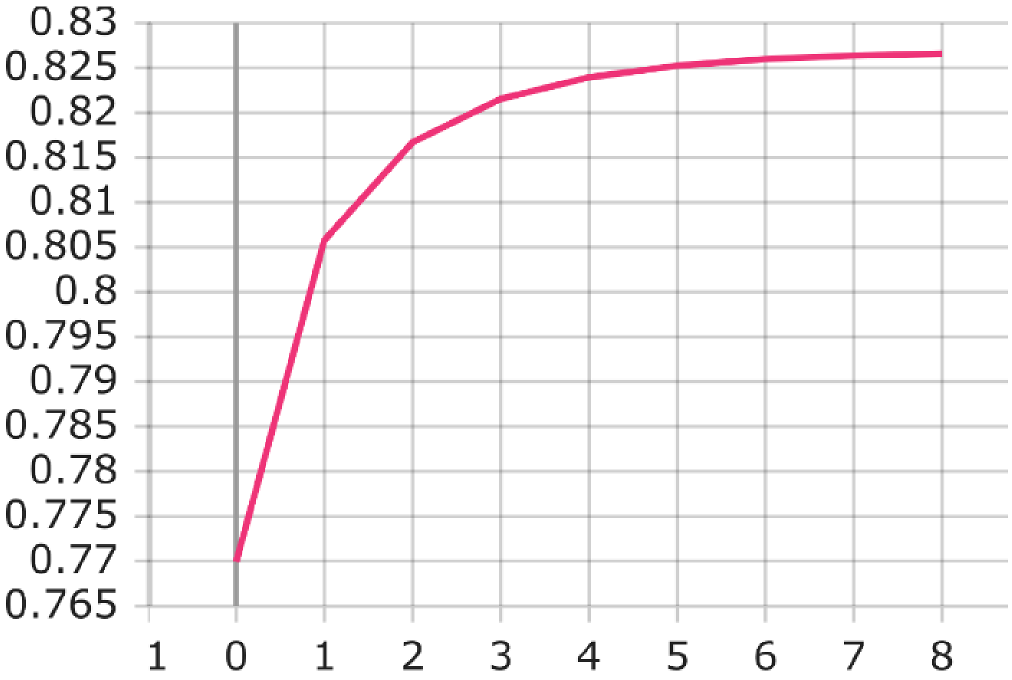


Figure 3-25 ***Weighted multi-class post-leak-free cross-fold splitting public-AUC performance***. *Here, the Y-axis is the AUC and X-axis is the steps of the model training. Similarly to Figure 3-26, the weighted multi-class model still achieves a reasonably high AUC performance after leak-free data-splitting even specifically for the public classes.*

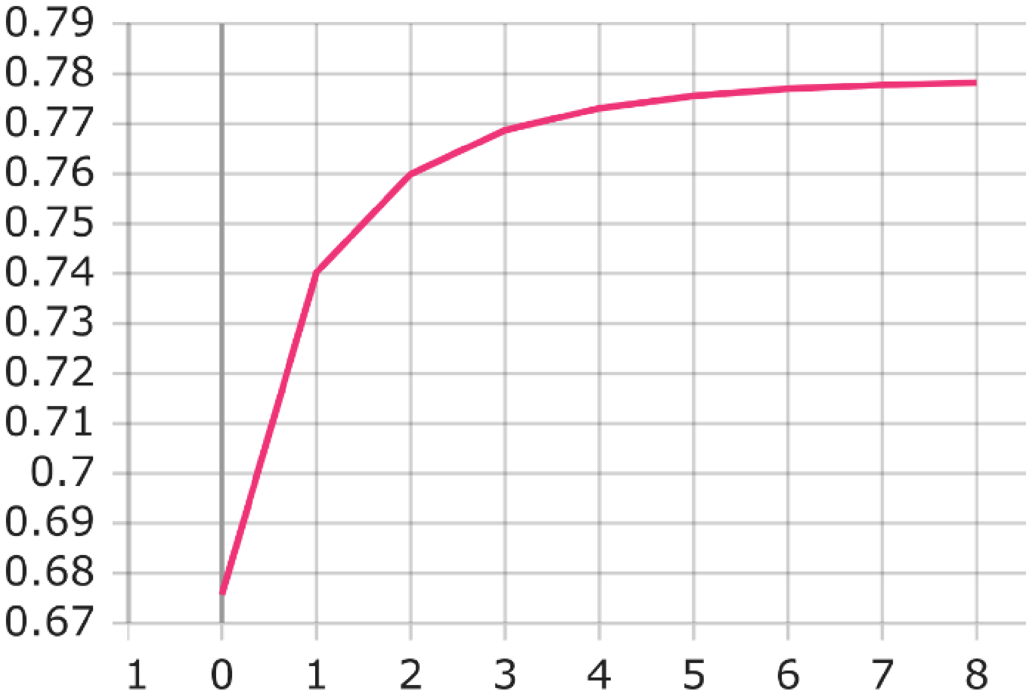


Figure 3-26 Weighted multi-class post-leak-free cross-fold splitting top-3 accuracy performance weighted by class size*. Here, the Y-axis is the modell top-3 accuracy and X-axis is the steps of the model training. Top-3 accuracy in a multi-class model is calculated by the percentage of the times the ground truth label is among the top-3 model predictions, in terms of the probability, for each label. In line with the precious two* figures, this *metric also diplays that the model is performing moderately well.*

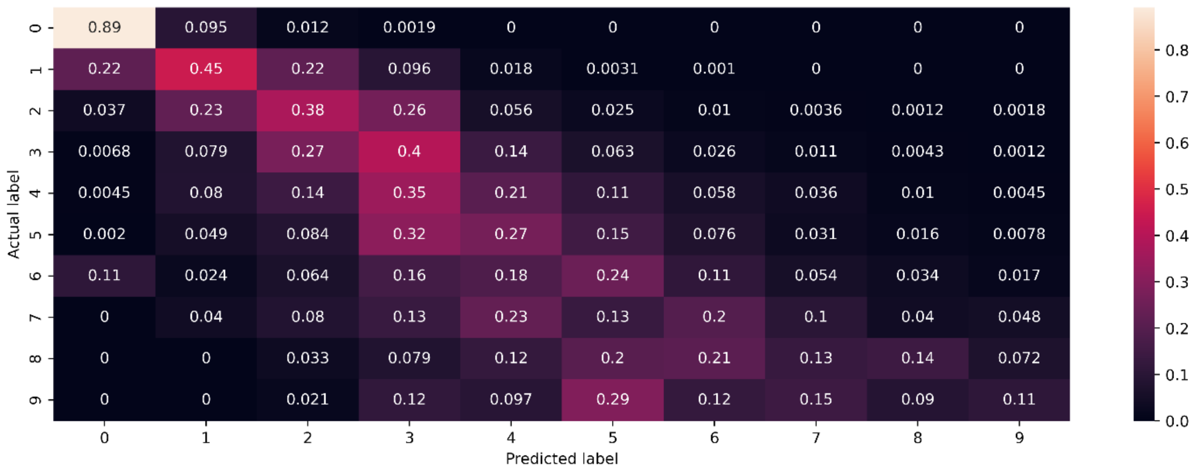


Figure 3-27 Best class-weighted multi-class SE-TCN model post-leak-free processing. *By looking at the diagonal in this confusion matrix it is clear that the fall in model's performance for the most part can be attributed to the public classes, while the performance for the private class remains stable. This is an intriguing observation, as at least for classes with the lower DoCs The amount of data is comparable to the data from the private class, yet the performance is dramatically*

lowered in comparison. This difference in performance could be the result of greater relative variability in the private-class data, in every split, in comparison to the public classes, as there is only a single representative sample from each convergent cluster for the private class, whereas there are multiple representatives per convergent cluster for the public classes.

3.3.8 High-Performant Sumrep Implementation

As shown in Chapter 2, many features can be calculated from antibody sequence data using Sumrep that can be used to summarise statistically and to compare antibody repertoires. Such features can also play an important part in the application of machine learning to the analysis of immune repertoires. Sumrep features were used both for informed data processing upstream of the machine learning applications and as features for predicting DoC values.

Although Sumrep was designed to handle repertoire sequencing data of "standard" size, it was not designed to handle the ultra-large sequencing data used in this study. To calculate a subset of Sumrep features for these datasets, a smaller HPC version was implemented based on the Apache Spark computing platform.

With a few exceptions, the calculated features were derived from tables of amino acids and their associated values corresponding to their respective features, which were determined empirically across multiple studies and compiled in Biopython[222] and the Expasy server (ProtScale tools)[223]. The calculation of these features was carried out over a sliding window of 5 residues of stride 1. Features calculated for full-length sequences include[1]: percentage of buried and accessible residues[224], average polypeptide-area buried[225], relative mutability of each amino acid (with Alanine=100 as a reference)[226], and a protein instability index (values above 40 indicate short half-life, and therefore, instability)[227]. Features calculated for CDR3 regions include: GRAVY index (sum of all amino acids hydropathy values divided by the sequence length)[179], the sum of Miyazawa hydrophobicity (3d-structure derived reside contact energy)[228], the sum of amino acids hydrophilicity[229], the sum of amino

[1] For the sake of brevity, studies which empirically determined the amino acid table of values that form the basis of the algorithmic implementation of each feature, are simply cited after the description of the feature.

acid bulkiness[230], the sum of amino acid isoelectric point[231], the sum of amino acid polarity[230], percentage of buried and accessible residues[224], average polypeptide-area buried[225], relative mutability for each amino acids (with Alanine=100 as reference)[226]. Additionally, we computed the full-sequence and CDR3 lengths. To inspect the distributions of these features, similar to the approach described in chapter 2, KDE of all features were calculated for every DoC using the pandas-on-spark library of the Apache Spark platform[232–235] and were plotted using Plotly Python graphing library (see *Figures 3-28* and *3-29*). We chose a small bandwidth parameter for these KDE calculations to maintain a high-resolution KDE shape in line with the frequency polygons in chapter 2.

From a visual analysis of these distributions, it is difficult to detect any significant difference between the different DoCs across most of these features. Even for those features which exhibit a variation in their distributions for different DoCs, such as full-sequence residue hydrophilicity or the fraction of buried residues in the CDR3 region, these variations are not discernible enough and/or correlative with the DoCs. For a summary description of these features refer to *Table 2-1.*

*To inves*tigate whether these features are collectively able to partition DoCs into separate clusters, PCA was applied to these features and the top 2 principal components plotted in *Figure 3-30. These PCA* results corroborate the judgement that summary statistics alone are insufficient for differentiating between DoCs. Additionally, to investigate the effectiveness of these features for predicting DoC when used in a machine learning context, a Gradient Boosted Trees (GBT) regression model was trained on these features. See the next section for more detail.

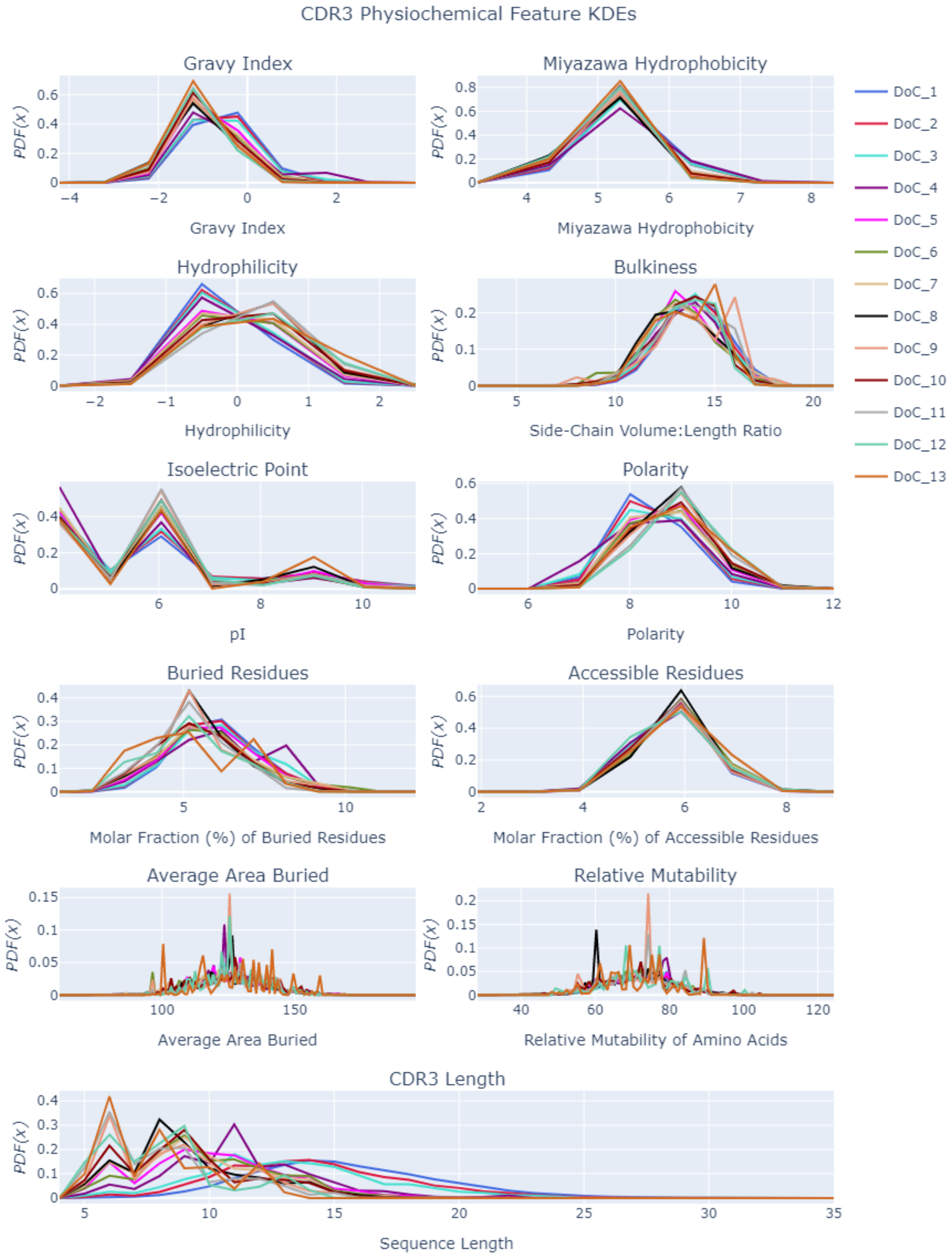


Figure 3-28 Sumrep CDR3 feature distributions. *KDE was performed over the calculated summary statistics for CDR3s of every DoC with the bandwidth parameter of 0.3 (to aquire high-resolution distributions) and pre-determined KDE boundaries set to the minimum and maximum values of every distribution.*

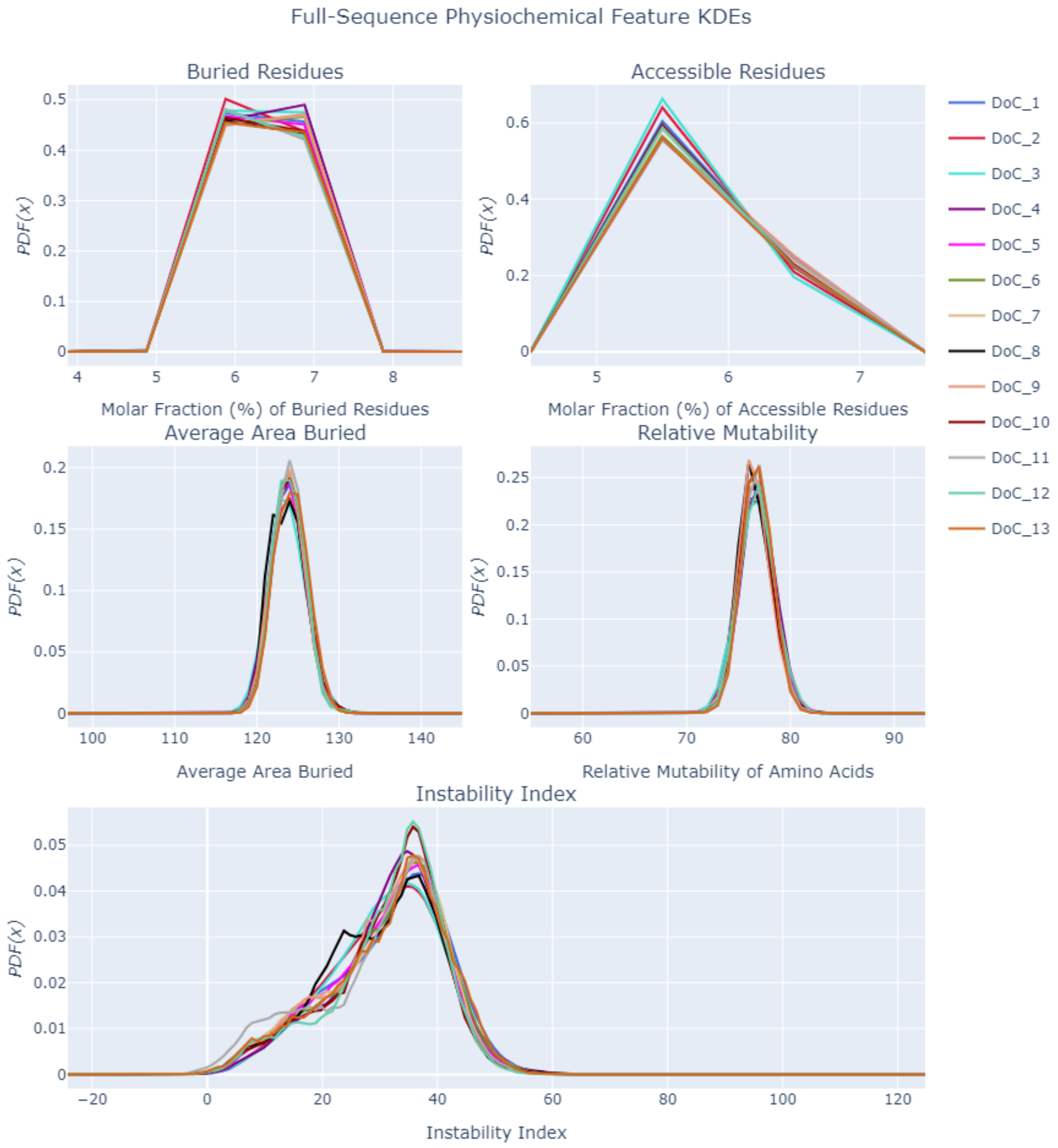


Figure 3-29 Sumrep full-sequence feature distributions. *KDE was performed over the calculated summary statistics for full-length sequences of every DoC with the bandwidth parameter of 0.3 (to aquire high-resolution distributions) and pre-determined KDE boundaries set to the minimum and maximum values of every distribution.*

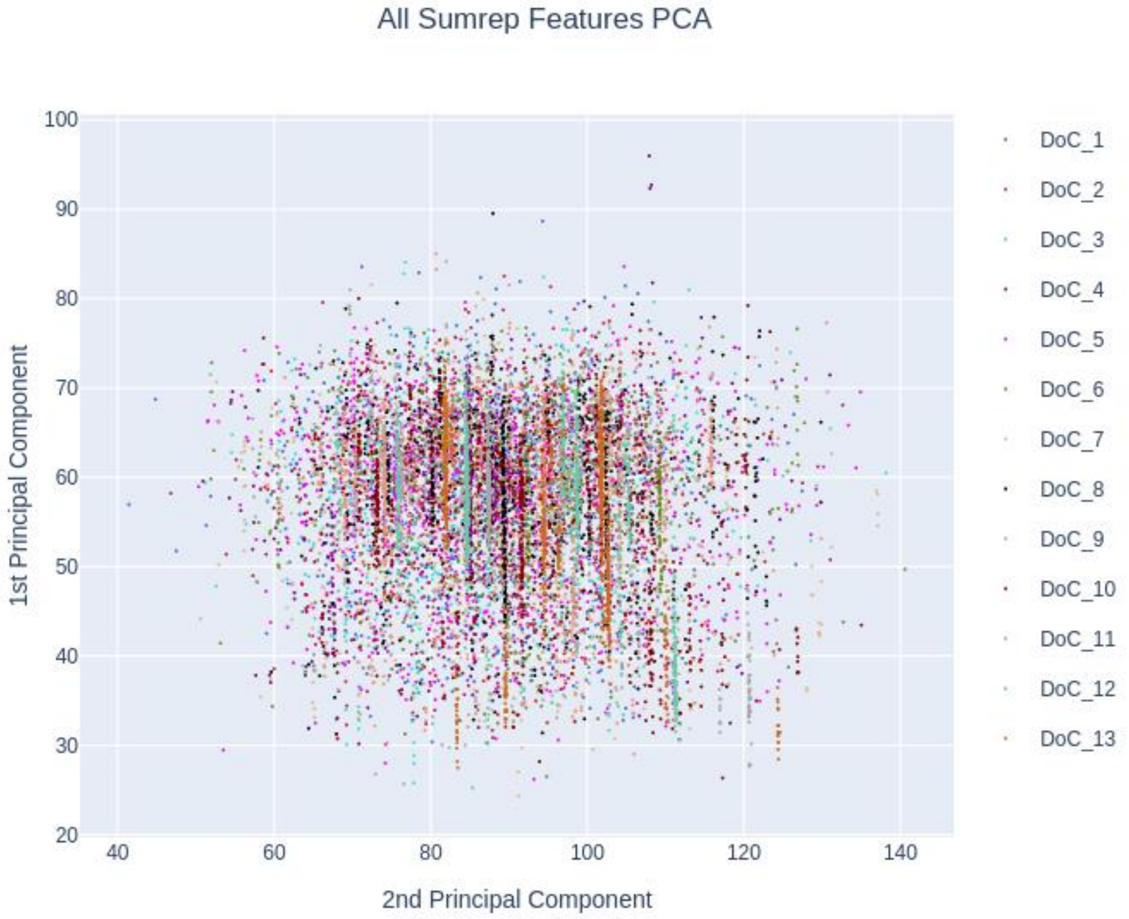
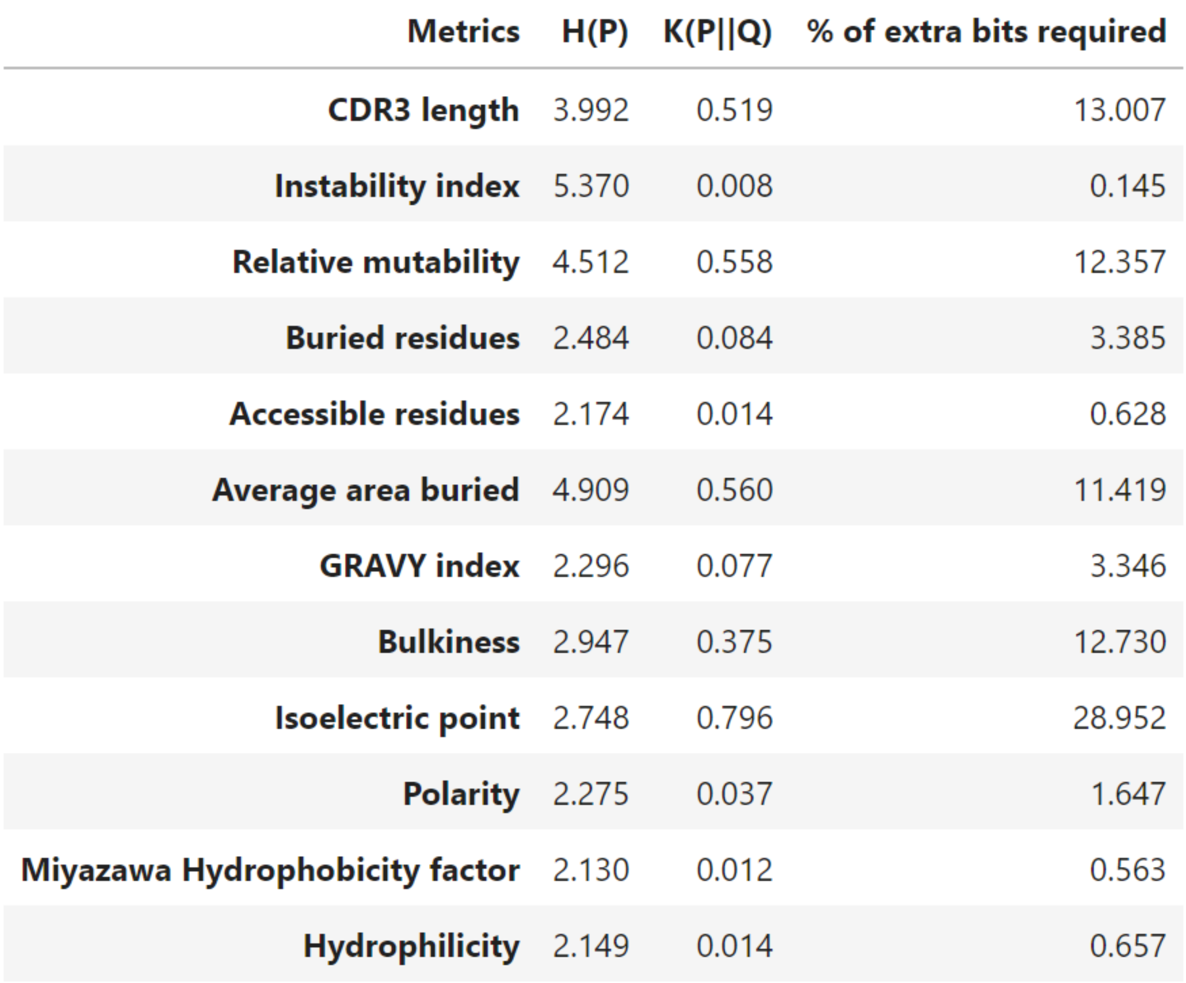


Figure 3-30 PCA projection of Sumrep feature statistics. *PCA was performed over the pooled multi-dimensional summary statistics (of both CDR3 and full-length sequences), and the principal components were plotted, with each datapoint annotated by their DoC, to investigate any potential clustering of the data.*

Finally, as part of the undersampling algorithm, we calculated 12 summary statistics for the undersampled data to determine whether, and by how much, the undersampled data differs from the population distribution data, and if so, to tune the undersampling strategy. We calculated the probability distributions for the summary statistics for both undersampled and population distribution datasets to calculate Shannon's entropy for the population distribution, here treated as the true distribution, the KL divergence of the undersampled data from the true distribution, and finally the percentage of extra bits required for the undersampled dataset, with respect to every summary statistic, to mimic the true distribution. Note that these extra required percentages are the percentage of the size of the undersampled dataset. The results in *Table 3-7 show the e*xtra bits required for the undersampled summary statistics in most cases (with the exception of the isoelectric point) are small enough for an optimised undersampling strategy that balances the tradeoff between minimising the class imbalance and minimising information loss. In conclusion, the undersampling strategy successfully retains most of the information and features (at least those considered within the scope of this study) provided by the complete dataset.



Metrics	H(P)	K(P||Q)	% of extra bits required
CDR3 length	3.992	0.519	13.007
Instability index	5.370	0.008	0.145
Relative mutability	4.512	0.558	12.357
Buried residues	2.484	0.084	3.385
Accessible residues	2.174	0.014	0.628
Average area buried	4.909	0.560	11.419
GRAVY index	2.296	0.077	3.346
Bulkiness	2.947	0.375	12.730
Isoelectric point	2.748	0.796	28.952
Polarity	2.275	0.037	1.647
Miyazawa Hydrophobicity factor	2.130	0.012	0.563
Hydrophilicity	2.149	0.014	0.657

Table 3-7 KL divergence of the undersampled data from the full-dataset. *Probability distributions of every summary statistic, irrespective of the DoC, was calculated (using KDE with bandwidth parameter set to 1.0) for the undersampled and full-dataset (see Equation 3-19). These probability distributions of the full-dataset's summary statistics were used for calculating the Shannon's entropy of the corresponding summary statistics. The Shannon's entropies, together with their*

corresponding probability distributions of the undersampled-dataset's summary statistics, were used to calculated the KL divergence between the undersampled and full-dataset (see Equation 3-19) w.r.t. every summary statistic. Finally, the extra information needed (in terms of the percentage of bits) by the undersampled dataset to fully describe the distribution of each summary statistic were calculated by Equation 3-18.

The percentage of extra bits is calculated by

$$f(x) = \frac{K(P||Q)}{H(P)} \times 100 \tag{3-16}$$

Where $H(P)$ is the true distribution's Shannon's entropy and $K(P\|Q)$ is the KL divergence of the undersampled distribution from the true distribution. The Shannon's entropy is calculated by

$$H(P) = -\sum(p_i \log(p_i)) \tag{3-17}$$

where p is the calculated probability distribution of the summary statistics of the true distribution. The KL divergence is calculated by

$$K(P||Q) = \sum(p_i \log(\frac{p_i}{q_i})) \tag{3-18}$$

3.3.9 RIIM: Relative Immunoglobulin Incidence Measure

Throughout this research, we have followed the established definition of commonality, a metric that discards vast amounts of granular information embedded within the diverse distributions of Convergent Clusters/clonotypes, even within the same DoC class. Information that could be crucial in addressing the possible asymmetric label noise introduced by the inadequacy of sequencing depth (even in the largest available datasets used here) to counter-balance the extreme distribution of clonotype frequencies and the limited size of cohorts in such studies that cumulatively contributes to the noise. Besides these practical concerns, a far more compelling discussion is raised by the question of whether this measure of commonality is adequate, or even realistic. As discussed, we know that there is a universal bias in V, D and J gene usages across individuals of a population, additional to the downstream gene recombination biases; phenomena with special importance for studying the convergence of immunoglobulin molecules in a population. By collapsing sequences into a small number of discrete classes, we effectively ignore all the nuances provided by these, and other biases involved in the latent generative processes underlying immunoglobulin molecule creation and, consequently, population-wide convergence.

As we have shown in the results in section 3.3.3, supported by previous research[236], clonotype frequency is not informative enough as a sole predictor for DoC of a clonotype, largely thanks to the extreme inequalities we observe in various frequency distributions in immune repertoires. Nonetheless, the same results demonstrate extreme trends that somewhat resemble a scale-free structure in the frequency distributions over DoC, e.g. the observed Shifted-Gumbel distribution, suggestive of complex underlying processes that likely have an influence on the observed convergence. While such general trends are not independently useful for our point-predictions, we cannot dismiss their potential supplementary roles in convergence. For instance, though we have argued that some of the public clonotypes may represent attractor solutions on which immune systems converge, a lot of clonotypes could be convergent responses to currently-circulating epitopes, or convergent memory responses to recurring epitopes. We have no idea how the convergent-clonotype frequencies are affected by the reason behind their convergence; however, it is safe to assume that, at least in some cases, the complex underlying generative processes would be independent of each other, and so would be the resulting observed frequencies. Dismissal of the differences in the immunological underpinnings of different sources of convergence within each DoC class could potentially oversimplify the ordinal relationships among public clonotypes of varying DoC values into a linear ordinal relationship, where the ordinal relationships may not be strictly linear in the respective feature space, resulting in a manifold learning that is prone to overfitting. The same reasoning applies to clonotypes with the same DoC class, though, we can make an exception for the private clonotypes, as in this research we are not concerned with modelling the "degree of rarity". In other words, private clonotypes are by definition non-convergent, at least as far as our empirical observations are concerned; therefore, their frequency distribution should have no bearing on their degree of convergence. Nevertheless, it is highly likely that some these private clonotypes could be public provided additional data, be it with increasing the cohort size or the sequencing depth of the current subjects, however, as it stands, we have no reliable priors to enable robust extrapolation of the frequency distribution of these hypothetical cases beyond the single subjects in which they are observed.

In conclusion, we postulate that, while the observed frequency of a Convergent Cluster in a single individual may have little value in making inferences about convergence, there is some structure in the relationship between Convergent Cluster frequencies and the degree of convergence. To investigate the validity of this postulate, first, we massively expanded the depth, and extended the dimensionality, of the analysis carried out in section 3.3.3, namely the extreme inequalities in the distributions of Convergent Cluster/clonotype over DoC (see figures *3-7 and 3-8*). The results of this analysis are demonstrated in *Figure 3-31.*

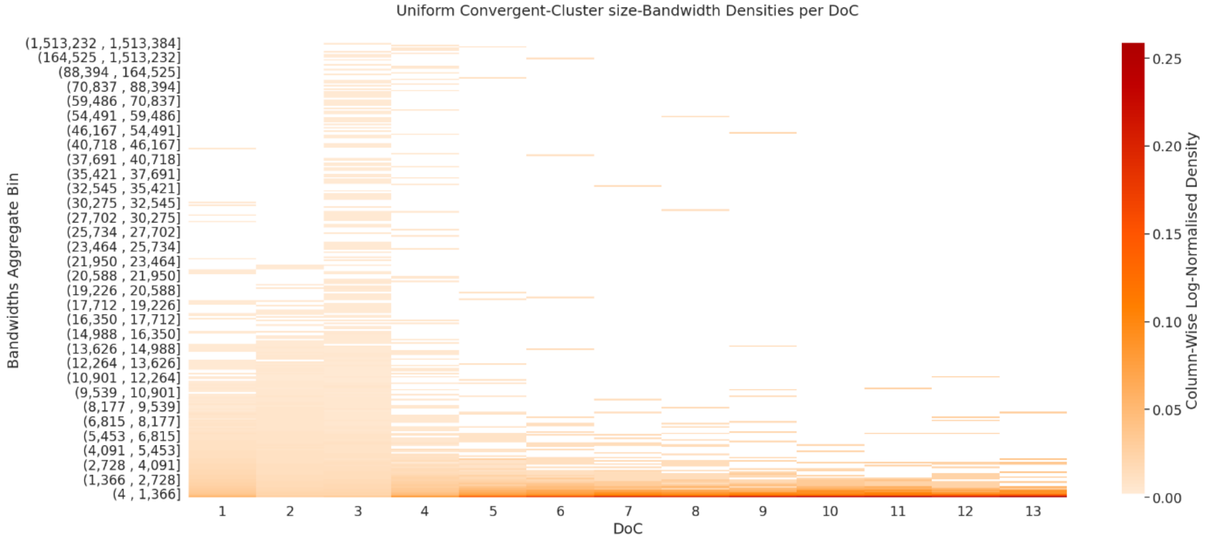


Figure 3-31 Convergent Cluster per DoC density map. *Top %10 largest Convergent Clusters/clonotypes and binned the cluster/clonotype sizes into equal-sized frequency bandwidths and plotted it as a heatmap, whereby the X-axis is the DoC values, and the Y-axis is the range of values the bandwidths represents; in this case, almost all bandwidths have the range (y_i,y_i+152]. Finally, I performed a column-wise log normalisation of the cell values, reflected by the temperature, to capture the density of frequency-bandwidths and provide a higher resolution view of the frequency densities. Note that, for the sake of readable visualisation, the Y-axis ticks are the eqaul-range aggregates of the bandwidths.*

Interestingly, the shape of the distribution arising in this heatmap looks remarkably similar to the distributions in figures *3-7 and 3-8. T*his is not very surprising given the fractal nature of the underlying distributions, i.e. clonotype frequency in individual repertoires[8,198] and clonotype frequency over DoC (see section 3.3.3), and that Gompertzian dynamics, such the observed shifted Gumbel distributions, emerges as a result of the fractal-stochastic dualism[237,238]. Such a fractal-stochastic dualism is theorised to arise from the non-linear and stochastic coupling of probabilities of at least two antagonistic processes. Here, I postulate that these two distributions are the clonotype frequency distribution in individual repertoires[8,198] and the clonotype frequency distribution over DoC. A possible interpretation of this is that large numbers of the clonotypes in the tail of the per-repertoire frequency distribution are private, which collectively, make up the vast majority of the total share of the sequences.

*Figure 3-31 provides a*dditional evidence for such a fractal-stochastic dualism, by providing a greater sampling depth and demonstrating a high-resolution density (the temperature) of varying range of frequencies per DoC. Moreover, it shows that the highest temperature (0.27, the maximum value on the temperature scale) in the heatmap is the rows of cells, corresponding to the frequency bandwidth with values ranging [4, 156] over the DoC range of [5, 13]. Moreover, the rows corresponding to the bandwidths with the approximate aggregate range [4, 1300] also have significantly high temperatures ranging between [0.1, 0.27], positively correlated with DoC.

In *Figure 3-31, the size* of a Convergent Cluster is the sum of its frequencies across the individuals, which might result in skewed results for public clonotypes with high frequencies across most individuals. Furthermore, the asymmetrically declining probabilities of DoC may cause misrepresentation of frequency per DoC. To this end, we calculated the Summed Proportional Geometric Densities (SPGD), which is the sum of the set of geometric means of frequencies per cluster in the set of Convergent Clusters of a DoC, normalised by the product of the number of Convergent Clusters and the number of sequences of the respective DoC:

$$\frac{\sum_{j=1}^{|\overrightarrow{c_d}|}\prod_{i=1}^{d}\overrightarrow{x_{j_i}}^{\frac{1}{d}}}{|\overrightarrow{c_d}||\vec{s}|}$$

(3-19)

Where d is the DoC value, s is the vector of the sequences of the DoC, $\overrightarrow{c_d}$ is the vector of the Convergent Clusters of the DoC which components are represented by x. We also calculated the Median of Proportional Geometric Densities (MPGD), which is the median of the set of geometric means of frequencies per cluster in the set of Convergent Clusters of a DoC, normalised by the product of the number of Convergent Clusters and the number of sequences of the respective DoC:

$$\frac{M_{j=1}^{|\overrightarrow{c_d}|}\prod_{i=1}^{d}\overrightarrow{x_{j_i}}^{\frac{1}{d}}}{|\overrightarrow{c_d}||\vec{s}|}$$

(3-20)

Where d is the DoC value, s is the vector of the sequences of the DoC, $\overrightarrow{c_d}$ is the vector of the Convergent Clusters of the DoC which components are represented by x, and M is the median operator.

As figures *3-32 and 3-33 d*emonstrate, SPGD and MPGD have a polynomial relationship with DoC and that a normalised frequency measure has a positive relationship with the degree of convergence as a polynomial function, even if it is not independently useful in making point-predictions.

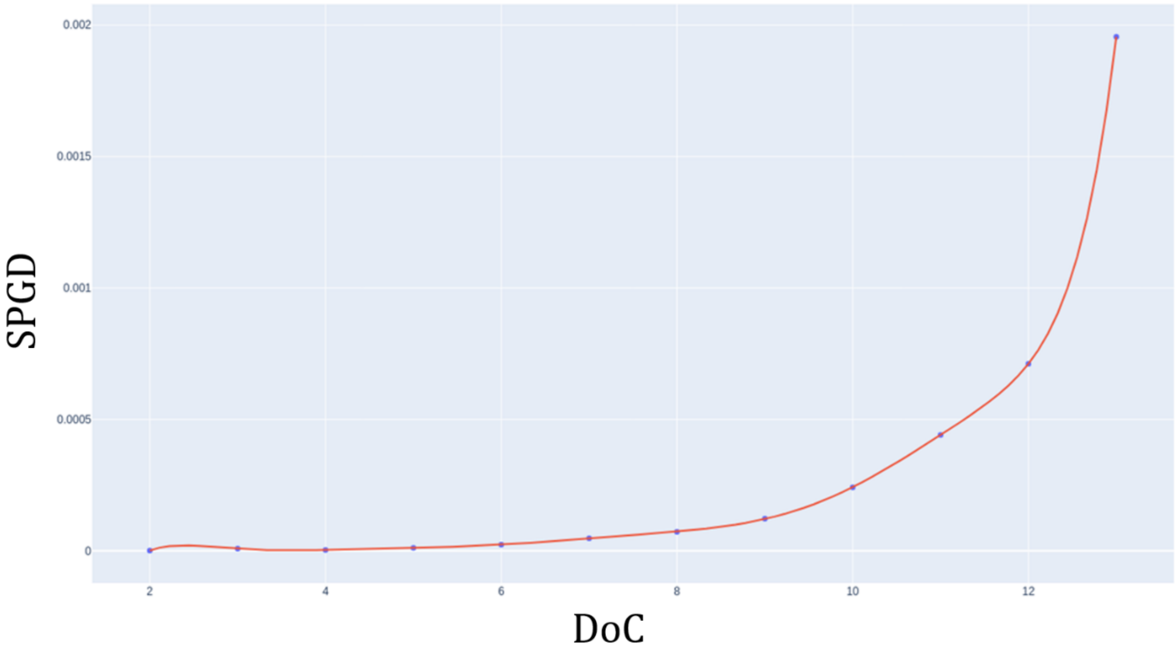


Figure 3-32 ***Sum of convergent cluster-size normalised geometric-means across DoCs.*** *The normalised gemetric mean of the per-subject frequency for every convergent cluster was calculated. The normalisation was simply the dividion of the geometric means by the total number of sequence multiplied by the total number of convergent clusters of the convergent cluster's corresponding DoC. The normalised geometric means of every DoC were summed and plotted, were a Chebyshev polynomial of degree 9 was used to fit the observed trend.*

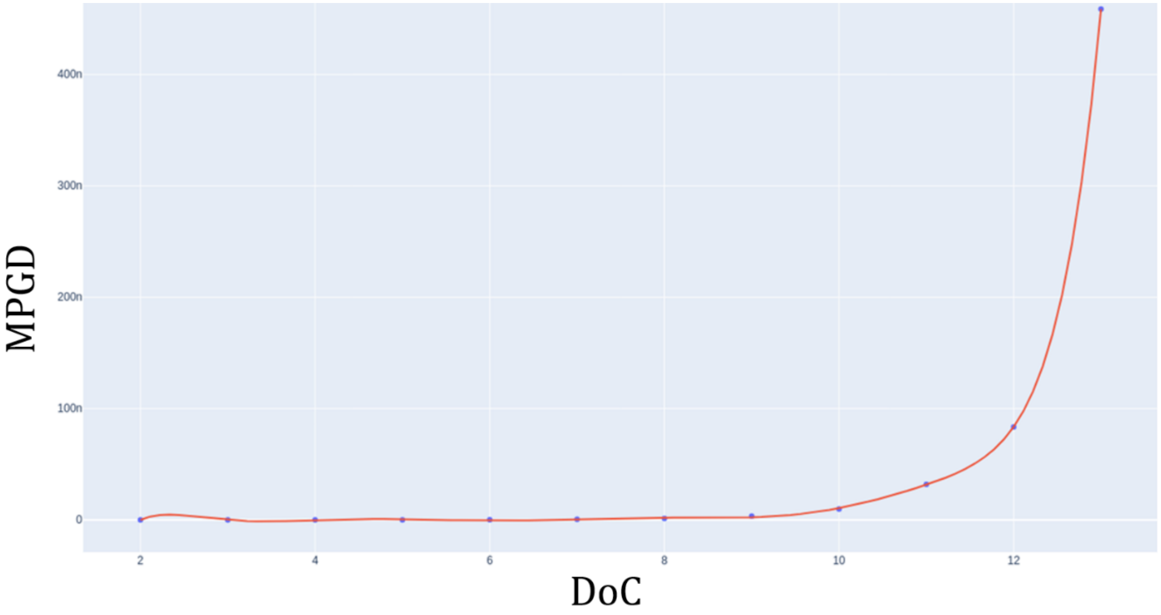
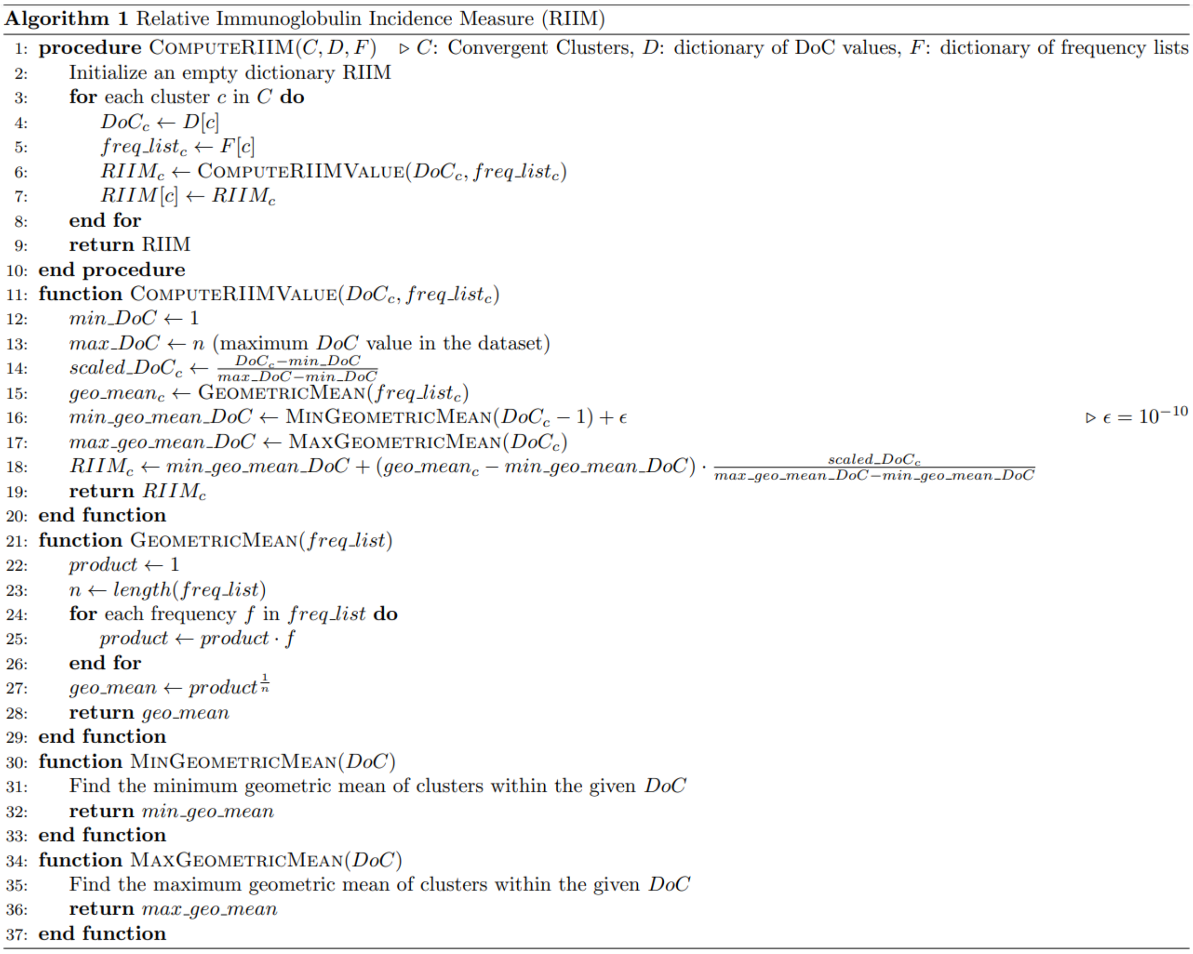


Figure 3-33 ***Median of convergent cluster-size normalised geometric-means across DoCs.*** *The normalised geometric mean of the per-subject frequency for every convergent cluster was calculated. The normalisation was simply the division of the geometric means by the total number of sequences multiplied by the total number of convergent clusters of the convergent cluster's corresponding DoC. The median of the normalised geometric means of every DoC were plotted, were a Chebyshev polynomial of degree 9 was used to fit the observed trend.*

Following the observations about the potential relationship between Convergent Cluster frequency and the overall trends in convergence, we introduce a new measure of convergence, namely the Relative Immunoglobulin Incidence Measure (RIIM). RIIM combines the geometric mean of a clonotypes' frequencies and the empirical knowledge of its DoC into a population-wide density measure of immunoglobulins that somewhat resembles the related concept of epidemiological Incidence Rate. The definition and implementation of this metric are thoroughly described in *Algorithm 3*, *Equation 3-19* and *Figure 3-34*. Notably, we see in *Figures 3-34* and *3-35* that by using RIIM values make for much denser regression targets, which should significantly improve the smoothing of the learnt manifold and therefore generalisability of the models. Nevertheless, the primary function of this metric is to capture the nuanced immunological underpinnings of convergence.

As an example, consider two clonotypes of DoC 3, where one is shared at frequencies 100, 1, 1 and the other is shared at frequencies 70, 50 and 40, the geometric means for these two are 4.64 and 51.9

respectively. If the minimum and maximum geometric means for DoC three are 1 and 60 respectively, and the label for $\mathrm{DoC}_2 + \varepsilon$ and DoC_3 are 0.0833 and 0.167 respectively, the labels for these two convergent clusters would be 0.0885 and 0.156 respectively, as opposed to both receiving 0.167 as target their values when MinMax scaling DoC to range [0, 1]. In the next section, we will use the methodology and findings of this section in the implementation of our deep regression models.



```
Algorithm 1 Relative Immunoglobulin Incidence Measure (RIIM)
procedure ComputeRIIM($C, D, F$)    ▷ $C$: Convergent Clusters, $D$: dictionary of DoC values, $F$: dictionary of frequency lists
    Initialize an empty dictionary RIIM
    for each cluster $c$ in $C$ do
        $DoC_c \leftarrow D[c]$
        $freq\_list_c \leftarrow F[c]$
        $RIIM_c \leftarrow$ ComputeRIIMValue($DoC_c, freq\_list_c$)
        $RIIM[c] \leftarrow RIIM_c$
    end for
    return RIIM
end procedure
function ComputeRIIMValue($DoC_c, freq\_list_c$)
    $min\_DoC \leftarrow 1$
    $max\_DoC \leftarrow n$ (maximum $DoC$ value in the dataset)
    $scaled\_DoC_c \leftarrow \frac{DoC_c - min\_DoC}{max\_DoC - min\_DoC}$
    $geo\_mean_c \leftarrow$ GeometricMean($freq\_list_c$)
    $min\_geo\_mean\_DoC \leftarrow$ MinGeometricMean($DoC_c - 1$) $+ \epsilon$    ▷ $\epsilon = 10^{-10}$
    $max\_geo\_mean\_DoC \leftarrow$ MaxGeometricMean($DoC_c$)
    $RIIM_c \leftarrow min\_geo\_mean\_DoC + (geo\_mean_c - min\_geo\_mean\_DoC) \cdot \frac{scaled\_DoC_c}{max\_geo\_mean\_DoC - min\_geo\_mean\_DoC}$
    return $RIIM_c$
end function
function GeometricMean($freq\_list$)
    $product \leftarrow 1$
    $n \leftarrow length(freq\_list)$
    for each frequency $f$ in $freq\_list$ do
        $product \leftarrow product \cdot f$
    end for
    $geo\_mean \leftarrow product^{\frac{1}{n}}$
    return $geo\_mean$
end function
function MinGeometricMean($DoC$)
    Find the minimum geometric mean of clusters within the given $DoC$
    return $min\_geo\_mean$
end function
function MaxGeometricMean($DoC$)
    Find the maximum geometric mean of clusters within the given $DoC$
    return $max\_geo\_mean$
end function
```

Algorithm 3 RIIM algorithm. *This algorithm encapsulates the RIIM, a new and alternative measure of immunoglobulin convergence we introduce in this chapter. It consists of several functions and procedures that work together to compute the RIIM value for each Convergent Cluster. The main procedure, `ComputeRIIM`, takes three inputs: `C`: A collection of convergent clusters, `D`: A dictionary containing Degree of Convergence (DoC) values for the clusters and `F`: A dictionary containing frequency lists for each cluster. The procedure first initializes an empty dictionary called `RIIM`. It then iterates through each cluster in `C`. For each cluster, it retrieves the corresponding `DoC` value and frequency list from the dictionaries `D` and `F`, respectively. The RIIM value for the current cluster is then calculated using the `ComputeRIIMValue` function, and this value is stored in the `RIIM` dictionary with the cluster as the key. The final `RIIM dictionary` is returned when the procedure finishes. The `ComputeRIIMValue` function calculates the `RIIM` value for a specific cluster using the following steps:1) It first scales the `DoC` value for the cluster by subtracting the minimum `DoC`*

value (1) and dividing by the range of `DoC` values. 2) It then calculates the geometric mean of the cluster's frequency list using the `GeometricMean` function. 3) The minimum and maximum geometric means for the given `DoC` value are computed using the `MinGeometricMean` and `MaxGeometricMean` functions, respectively, with a small constant `epsilon` (10^{-10}) added to the minimum geometric mean. 4) The RIIM value is calculated using a linear interpolation between the minimum and maximum geometric means, with the scaled DoC value as the interpolation factor. The `GeometricMean` function computes the geometric mean of a given frequency list by multiplying all the frequencies together and then taking the nth root, where n is the length of the list. The `MinGeometricMean` and `MaxGeometricMean` functions find the minimum and maximum geometric means, respectively, of the clusters within the specified DoC value.

$$RIIM(x) = x \begin{cases} 0 & DoC = 0 \\ MinMax(G, label_{DoC_{i-1}} + \varepsilon, label_{DoC_i})) & DoC > 0 \end{cases}$$

$$G = MinMax(Gmean(x_{gs}) \ \ , \ \ Min(DoC_{Gmean}), \ \ Max(DoC_{Gmean}))$$

(3-21)

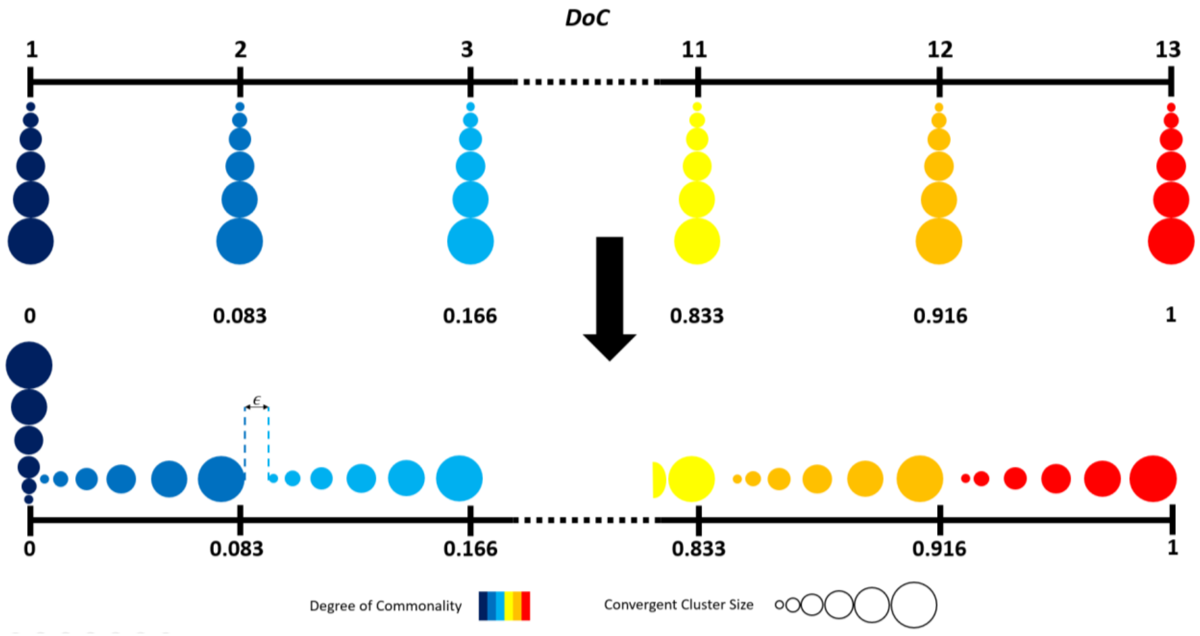


Figure 3-34 RIIM transforms sparse targets into a smoothly distributed target range. When converted to regression labels, i.e. between zero and one, the labels are dilated between the regression values coresponding to the DoC-1 and DoC of the input. The convergent clusters are ranked based on the geometric mean of their sibject-wide frequencies, and finally using min-max scalling are assigned a label.

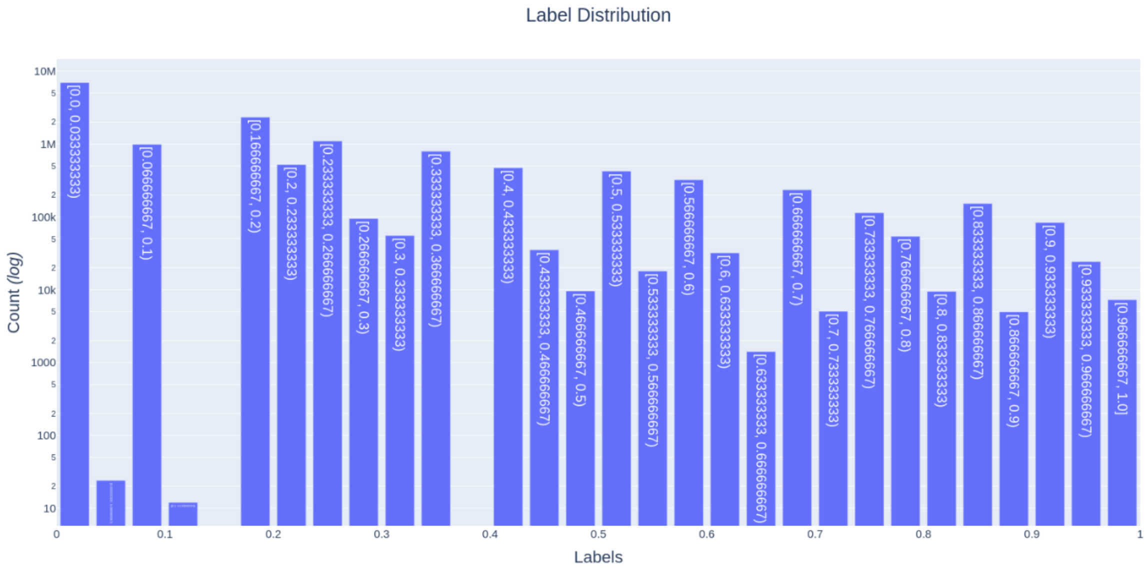


Figure 3-35 ***Log-Scaled RIIM Histogram.*** *The resulting distribution of the label dilation is plotted with a log-scaled Y-axis, which shows that the label dilation protocol successfully projects the otherwise discretely interspersed labels onto a truly continuous scale.*

3.3.10 RIIM Predictions Using SE-TCN and Transformer Architectures

In section 3.3.9 we performed a series of analyses to investigate the relationship between Convergent Cluster frequencies and the degree of convergence. Following the findings, we proposed RIIM as a new definition for the degree of convergence and implemented regression labels according to this metric. Here we will use the neural network architecture used in previous sections and compare its performance to the performance of our implementation of the transformer architecture, which is the current state-of-the-art deep learning approach for modelling genomics data.

The same SE-TCN model used in previous experiments was applied to predict the scaled labels but with modifications. Instead of one-hot-encoding the input sequences at the amino acid level, the sequences were sliced with a sliding window of size three and stride one, the resulting overlapping 3-mers were one-hot-encoded, and these were used as inputs to the model. Furthermore, these inputs were passed into an embedding layer of 32 dimensions before being passed into the remainder of the model (which was otherwise unchanged). In addition, we implemented a transformer architecture as a state-of-the-art approach for comparison.

Ever since the invention of the attention mechanism, the transformer architecture has become the state-of-the-art technique for sequential data and has several components which distinguish it from other types of architecture. The central component of this architecture is the attention layers, which allow the model to focus more on the features that are important for a given task.

Although the attention mechanism is not exclusive to the transformer architecture, together with other unique components in the transformer architecture, it grants the model advantages over other types of architecture. One such component is the positional embedding, which (similar to the standard embedding layer) makes the model invariant to input size/length by computing a mask to ignore zero-paddings, used for equalising the sequences to a fixed length required for embeddings. Though different techniques exist, here the positional embedding learns the order/position of the features in the same manner as the standard embedding layer learns to embed feature token indices. This positional information allows the model to learn long-distance relationship/dependencies within the data as well as the role that the order plays within these relationships. Furthermore, the model includes a compression-projection module, similar to the SE-TCN, which facilitates the learning of abstract relationships among features within data further. Another notable component is the layer normalisation, which plays a similar regularisation role to batch normalisation. Finally, the most specific part of the architecture is the way attention mechanism is used, i.e. multi-head attention, which is simply stacking multiple attention layers in parallel, with each identifying different important features to focus on, and finally concatenating their outputs. For more specific details see *Figure 3-36.* In addition to the SE-TCN and transformer models, we also implemented GBT using the Mlib library of the Apache Spark platform[232–235] for prediction of RIIM values based on the summary statistics features. This was trained and validated on different splits and the inferences made on a test set were plotted using the Plotly Python graphing library, as shown in *Figure 3-37*, with the MSE plotted over the DoC distribution. Furthermore, HPC-sumrep was used in two ways: to investigate the degree to which repertoire features may be useful for predicting the DoC; and to evaluate the impact (if any) of the chosen undersampling strategy on the characteristics of the dataset. It is worth noting that this evaluation was only possible because a relevant subset of the sumrep features were implemented on top of an Apache Spark engine, thereby reducing the time required to run these calculations by several orders of magnitude.

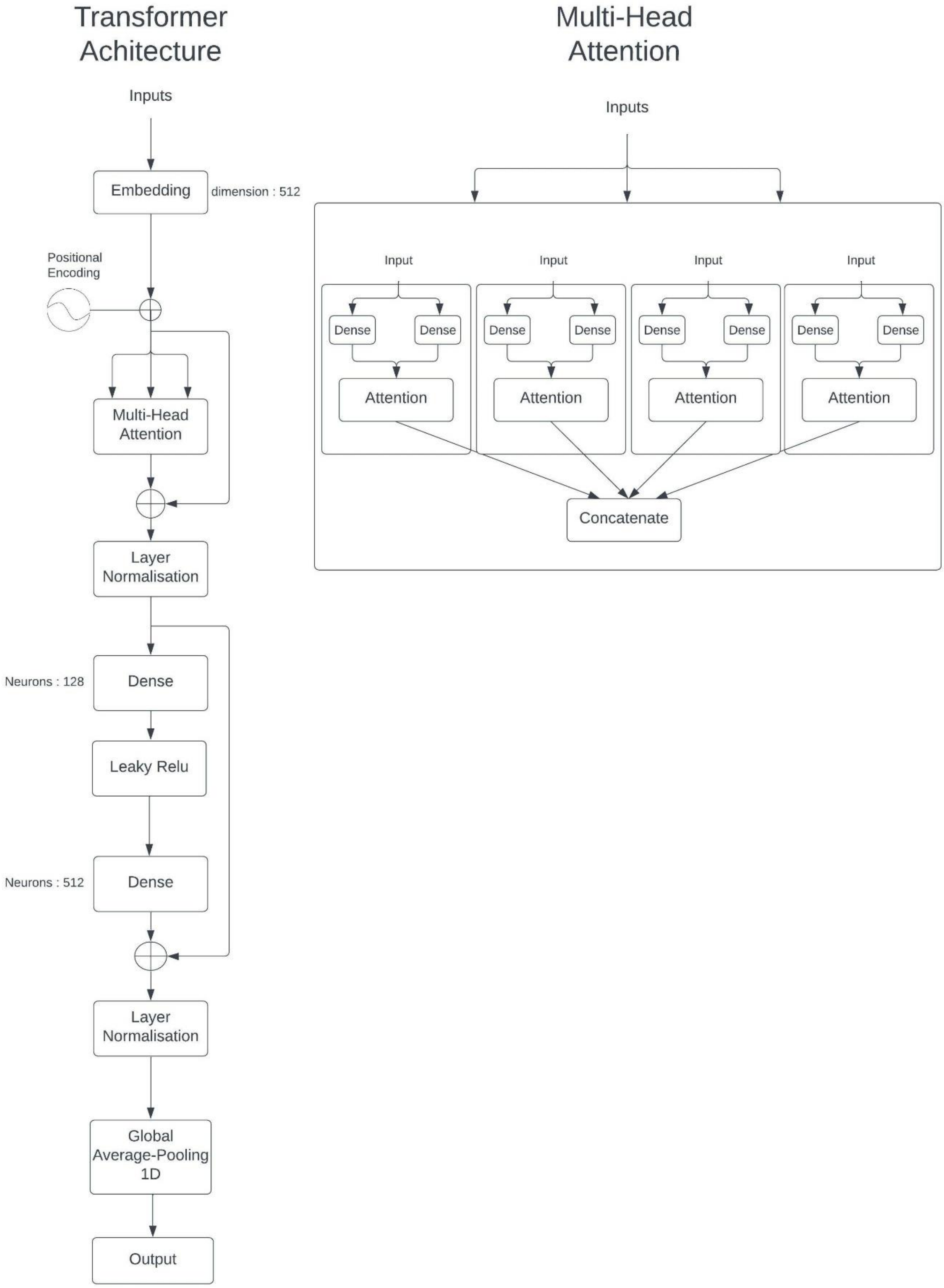


Figure 3-36 The overview of Transformer architecture.

Figure 3-37 evaluates (at a DoC-specific level of granularity) the performance of all three models in terms of MSE. It demonstrates that, not only, a model trained on our summary statistics is not nearly as good as models trained on genomics data, our summary statistics are not at all expressive enough for this task. This is unsurprising: as we saw in the section 3.3.8, there appears to be (at most) weak differentiation between DoCs associated with sumrep summary statistics.

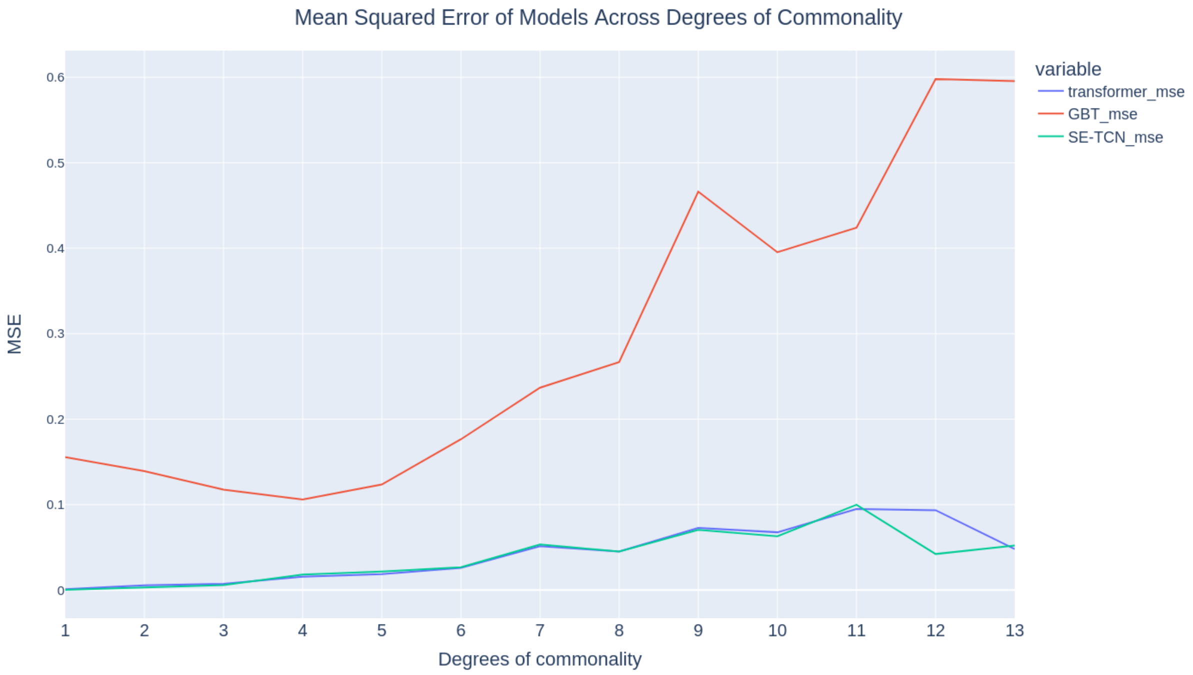


Figure 3-37 ***MSE performance of the regression models.*** The performance of the SE-TCN, Transformer and GBT (trained on summary statistic features) models *for predicting the scaled continuous labels* was evaluated by the MSE *of the models for every DoC*, showing that there is not enough signal in summary statistic features to predict DoC well *overall* and that the *deep* neural network *models* trained on genomic features perform similarly *very well across all DoCs*.

Interestingly, and perhaps surprisingly, the SE-TCN model marginally outperforms the transformer model. This is further evident in *Table 3-8, which sum*marises the performance of the two models in terms of both MSE and MAE averaged across all DoCs.

	MSE	*MAE*
Transformer Model	0.0163	0.0864
SE-TCN Model	0.0157	0.0820

Table 3-8 The summary of the deep neural network models' overall performance, irrespective of the DoCs.

To take a closer look, we evaluated these models by MAE at a DoC-specific level. These results, summarised in *Figure 3-38, show that* the SE-TCN model for majority of the labels is on par with the transformer model and outperforms it for some labels.

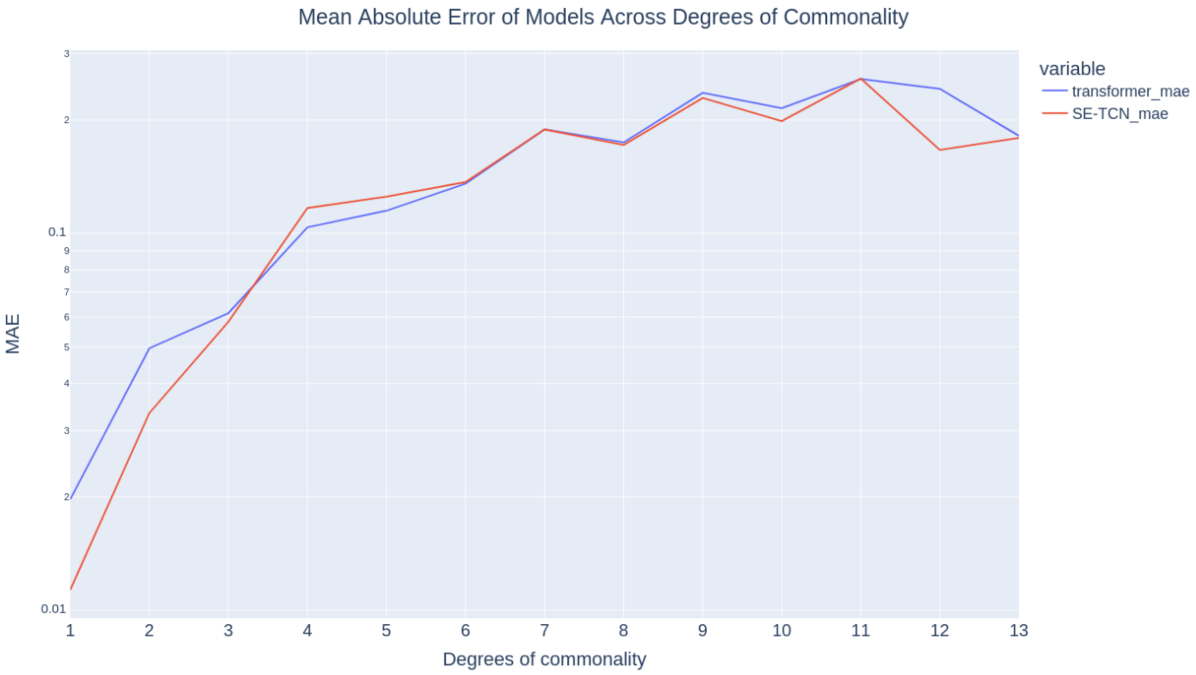


Figure 3-38 MAE performance of the regression models. *The performances of the SE-TCN and Transformer models for predicting the scaled continuous labels were evaluated by the MAE of the models for every DoC, showing that the SE-TCN model slightly outperforms the Transformer model.*

To explore these results with finer granularity, the error distributions (i.e. the difference between the predictions and the labels) were calculated for both models at every DoC. These results are plotted in *Figure 3-39, which inc*ludes both outliers and inliers in addition to the error distributions. As we can see in this figure, the SE-TCN model, for the most part, has a smaller variance in error distributions, with outliers closer to the mean, and peaks closer to zero error. It should be noted that, though the size of outliers and inliers seems deceptively large on the plot, they are only a small fraction of the total data per label. This is further supported by *Figure 3*-40, where we can see that even for classes of higher DoC, which are more prone to errors, the interquartile range and the standard deviation remain bounded within very good margins of error, while means and medians of the error remains very close to zero. It should be noted that an error of 0.0833 roughly corresponds to a misprediction by one DoC. Hence these models are, on average, accurate to within one DoC and more accurate than that for the "less public" data. When predictions are incorrect about the higher DoC labels, this tends to be an underestimation of the DoC, whereas lower DoC labels tend to be overestimated (see *Figures 3-39* and *3-40*). Overall, this behaviour is to be expected, owing to the issue of large label imbalances, which (unlike with a classification model) cannot be accounted for straightforwardly by adjusting the loss function.

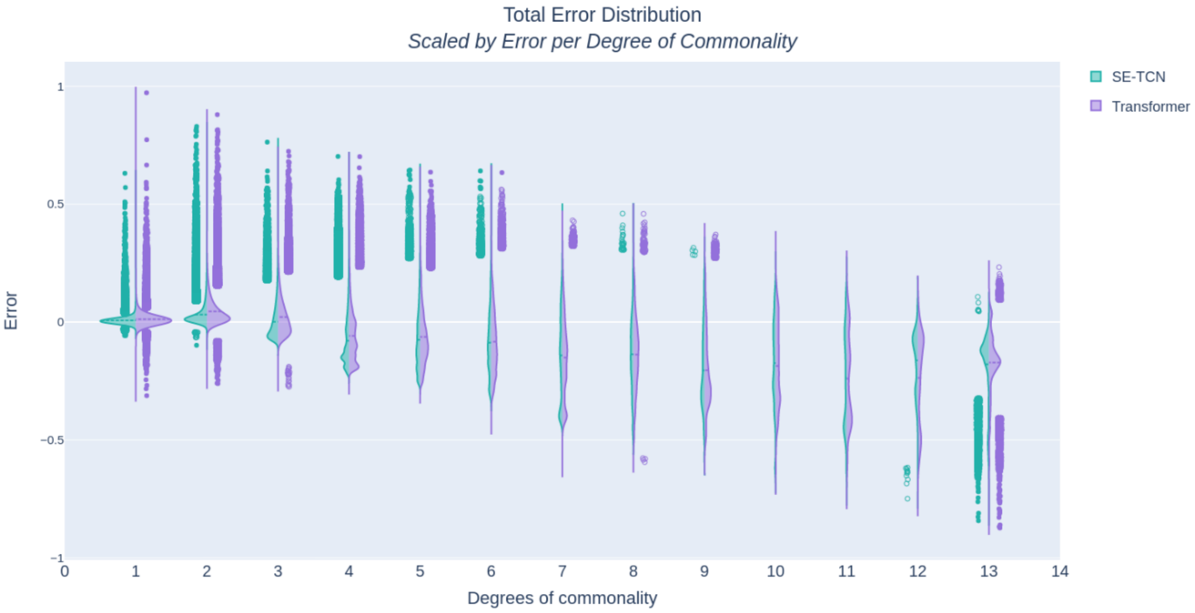


***Figure 3-39 The Error distribution of the SE-TCN and Transformer models per DoC.** The error for every input was calculated simply by the difference between the predicted value and the ground truth label. The distribution of the errors for each model for every DoC was calculated and plotted with the mean (small line inside the distribution orthagonal to its spread), outliers (solid balls) and suspected outliers (empty balls). The distribution of the errors expands with DoC; however, it remains within a very good range. While the outliers visually seem large in numbers, they are small in number in comparison to the total data per class (see Figure 3-39 for more details).*

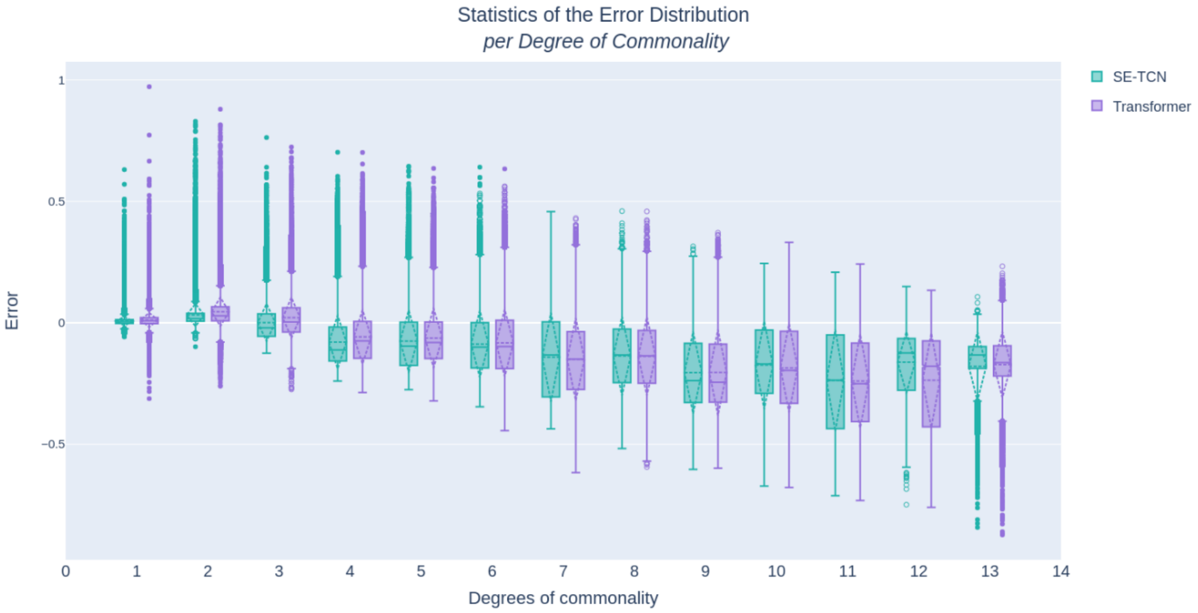


Figure 3-40 Statistics of the Error distributions of the SE-TCN and Transformer models per DoC. *Following Figure 3-39 we calculated several statistical measures to elucidate the distributions of error further and visualised the results by box-whisker plots. These statistics include interquartile range (boxes), median (the solid line dividing the two boxes), mean (the dashed line parallel to the median), standard deviation (the dashed triangles), dispersion and skewness beyond the interquartile range (whiskers), outliers (empty balls) and suspected outliers (solid balls). Evidently, the errors of the model, even for the smaller minority-classes (relatively high error due to insufficient data) are still within a very good range given the standard deviation and the interquartile range.*

The observed power law distribution of DoC suggests that the immune system's architecture is fractal and inherently complex, likely involving various generative processes underlying the population-wide convergence of B-cell clonotypes. This is expected from the self-organisation dynamics that arise from bidirectional bottom-up (individual-repertoire evolutionary dynamics) and top-down (population-level evolutionary dynamics) governing dynamics of immunity. As a result, we expect the systemic statistical differences we observe in clonotype frequency across commonality to reflect the underlying generative dynamics and have nuanced relevance in a systems immunology view of convergence, and consequently, the simpler view of convergence, as DoC does represent the full picture of the complexities. Subsequently, we proposed RIIM, a novel measure of immunoglobulin population density as an alternative to commonality, which incorporates the geometric mean of clonotype frequencies in all individuals they occur in and provides a more comprehensive and realistic measure of shared clonotypes. By incorporating frequency, this method resembles the well-established concept of incidence, commonly used in epidemiology, and captures the relative incidence of shared clonotypes across individuals to provide a more meaningful and granular representation of shared clonotypes in the immune system. It is essential to note that the utility of this measure is purely to provide a more accurate measure of immunological convergence and not necessarily for optimisation of machine learning results, though, that may or may not be an indirect consequence. Although we present the model performance on predicting RIIM within the context of DoC values, this mainly intended for consistency with prior work and our own resuts, as well as a domain-specific way of model error assesment. However, direct comparison of models which predict DoC against models which predict RIMM may be ill-advised, as these measures represent equivarient views of immunilogical process, and by extention convergence, and as such, should result in learning of different feature spaces by deep learning models.

3.4 Conclusion

The work undertaken here combines the deepest AIRR-seq data available to date, carries out a series of transformations of the data and the labels and finally utilises deep neural networks for the prediction of genomic convergence across a human population in a way that can be extended to making predictions about the state of immunoglobulin molecules in the broader public. These results demonstrate that it is possible to predict the degree of genomics convergence of antibody clonotypes using machine learning with a high degree of granularity. To the best of our knowledge, these models achieve state-of-the-art performance both in terms of the granularity of predictions and levels of accuracy but also highlight several issues and requirements that should be addressed in future research. The most immediate requirement is more data for the minority classes, which, as ultra-deep sequencing results become more abundant, should gradually become less of an issue. There is also a need for the development of techniques that could address the problem of imbalanced datasets.

Finally, our work demonstrates that the analysis of immune repertoire data, particularly in the context of machine learning, can be a quite nuanced process, and vigilance must be undertaken along with a more granular interpretation of the results. For instance, see section 3.3.7, where the leak-free undersampling strategy was developed to avoid misleadingly positive machine learning results and the higher resolution predictions of the machine learning models (section 3.3.10) to provide a more detailed and fair presentation of model performance.

4 Discussion

In this research, we apply and combine various statistical and ML techniques for a systems understanding of the convergence among immunoglobulin repertoires. We demonstrate that statistical summarisation of immune repertoires in combination with the application of deep learning methods to analyse raw sequencing data provides a powerful pipeline for making systemic predictions about immune repertoires.

In chapter two, we demonstrate that using a plethora of summary statistics could enable better interpretability of immune repertoire analysis, and although it can sometimes be difficult to gain insights from, summary statistics could be used in conjunction with ML methods for making informed decisions about various stages during the development of the ML pipeline, particularly in data processing and maybe even for the interpretability of results.

In chapter three, we combined genomics datasets from ultra-deep AIRR-seq studies, performed a series of data transformation techniques and finally developed deep neural network models capable of successfully predicting the DoC of immunoglobulin molecules, on a scale of zero to one, with a very low degree of error. The final results provided in section 3.3.10, particularly *Figures 3-39* and *3-40,* demonstrate the success of our deep learning approach in modelling genomic convergence of immunoglobulins. We processed the data from the 13 subjects within our datasets, such that, all antibodies falling under the same “V3J” clonotype definition are clustered into the independent convergent clusters. First, each convergent cluster’s DoC value is determined (by how many subjects share the cluster) and continualised to equidistant values between zero and one (cDoC) by min-max normalisation. Then, for every DoC class, we calculated the geometric mean of the subject-specific frequencies of every convergent cluster, and then min-max normalised these means between the cDoC corresponding to the DoC class and the cDoC of the DoC one degree below, as the respective lower and upper bounds. Given the large class imbalance in our data, we devised an undersampling protocol, which maintains the integrity of the data in a variety of ways. Primarily, this algorithm samples at least one sequence from each subject in every convergent cluster, thereby respecting the variance in the data. Furthermore, it minimises the KL divergence of the various antibody sequence summary statistics of the sample and population distributions. Whilst this approach samples every convergent cluster in all public classes, it increasingly increases the number of, non-identical, but redundant, samples per convergent cluster in line with the decreasing size of the DoCs. These redundant samples in theory act as adversarial examples, which should help with improving model performance, particularly for the minority classes. We divided this undersampled data in a "leak-free" manner into 10 cross-validation folds, to prevent leakage of data from training and validation sets into the test set, which would have resulted in a flawed evaluation of the models. Furthermore, we ensured these leak-free sets contained comparable sample sizes for every DoC label. Finally, we developed deep neural network models trained on this dataset, which achieved, to the best of our knowledge, state-of-the-art results. Furthermore, the deep learning models trained on the genomics data clearly outperformed the GBT models trained on summary statistics, though interestingly, our bespoke SE-TCN model slightly outperformed the transformer model.

Despite achieving a high level of performance, the models' error variance, i.e. uncertainty, remains larger for higher DoCs and increases with the DoC label. This is likely to be due to two factors; namely, the large class imbalance of our data increasing with the DoC and the potential mislabelling of data as a result of the small cohort size of 13 individuals.

The class imbalance is the easier of the two problems to tackle. However, there has not been much development in the research for deep regression imbalance[239], although, recently, Yang *et al* developed a technique for addressing imbalanced labels, which are normally distributed[239]. We plan to extend this technique for addressing this issue for labels sampled from non-normal distributions, particularly the heavy-tailed distributions such as the distribution of our data's labels.

The second problem, namely the mislabelling problem, is an inherent issue that will more or less persist as long as machine learning is a sensible approach to this task. In other words, unless one samples the whole population of the earth, or sufficiently close to it, this problem stands, though obviously, the likelihood of mislabelling decreases to insignificant levels with samples of possibly only a few orders of magnitude larger than ones available to use, if the sampling is done adequately. Specifically, if one acquires ultra-deep AIRR-seq samples from reasonable number of individuals across a reasonable breadth of subpopulations, defined by various factors, e.g., geographical, genetic etc., one should be able practically to eliminate the mislabelling problem, as far as machine learning is concerned.

Currently, as such data is out of reach, the utility of machine learning may prove to be useful to tackle this problem, despite mislabelling posing issues for the learning process. For instance, if there are enough correctly-labelled data from the minority classes, deep learning models, in theory, could learn the feature manifold correlating with the degree of immunoglobulin convergence across the true population. Therefore, it could be possible that the models could predict theoretical labels, perhaps, more accurately than the labels empirically determined from data coming from small cohorts. This, of course, must be tested and indeed can be tested. We plan to increase our current data with other large datasets and update the labels accordingly. We can then test this hypothesis by evaluating the results of our models trained on the current dataset, particularly for the samples which have positive value errors, against the updated labels of the hypothetical aggregate dataset. However, this probably requires the models first to be retrained and calibrated by the deep-regression-imbalance technique(s) we are planning to develop. This is because the error distributions are currently increasingly biased towards the negative values of error for the higher DoC label, whilst the lower DoC labels have error distributions biased towards positive values (see *Figures 3-39* and *3-40*), which is most likely to be caused by the data imbalance.

Of course, the broader problem is that the distribution of immunoglobulin molecules in a population, whilst in part due to varying generation probabilities based on genomic biases, is likely to be more greatly affected by their functional impact. For instance, some antibodies may bind to reucrring "public epitopes"[240], some may respond to recurrent or endemic pathogens and some may be polyspecific to various epitopes[241] and have a high "Long-term immunopotentiation" – the ability to develop high-affinity mature antibodies against various antigens, all of which could have an impact on the probability distribution. Therefore, to quantify, predict and understand convergence, one ultimately needs to do this in the context of the function, which remains elusive due to the lack of ultra-deep paired-chain single-cell AIRR-seq and high-throughput accurate structure prediction. Even if these technologies were available, one could not deterministically predict functional convergence due to the nature of degeneracy in biological structures and function[19], yet, deep learning models prove to be useful in that area, too[149]. Nonetheless, understanding genomic convergence has its own somewhat independent basis and immense value in understanding basic biology as well as industrial applications.

One of the important future directions one can take is in cross-validating different definitions of convergence. As we have seen, we have chosen a strict definition of convergent clusters, which may or may not reflect the reality of convergence, though, similar arguments can be made for or against all other definitions, which inevitably results in mislabelling of some of the instances. Therefore, similar instances with differential labels should result in high model uncertainty. And particular to genomics data, it is also possible that many correctly labelled instances would have high feature similarities, adding further necessary uncertainty to the model. Moreover, as we have seen, DoC and clonal frequency distributions are extremely long-tailed, whereby clonotypes with mid-to-high DoC values often fall on the extreme lower end of the long-tailed clonal frequency distribution, at least in some of the repertoires across which they are shared. Despite using the deepest data available to date, it is still very likely that some convergent clusters in our data are missing representative instances from some individuals due to a lack of sequencing depth, resulting in mislabelling. Similarly, sequencing bias could result in better coverage of some convergent clusters over others of equal DoC, resulting in another form of adding to model confusion by mislabelling. we only have a maximum of 13 individuals, As such, many sequences would be inherently mislabelled demanding extreme sequencing depths to capture or because of not having a large enough population.

In this thesis, we also introduce the “Blind Mapmaker” hypothesis to provide a novel theoretical framework for answering the question of why public clonotypes, other than those resulting from immune responses to endemic, common or circulating antigens, might exist. While we conduct some analysis for testing the basis of this hypothesis and its underlying assumptions, a great deal of future work is required for testing this hypothesis. In particular, in section 3.3.3 we show that the public clonotypes are not simply those with the closest sequence similarity to germline sequences (see *Figure 3-3*). Section 3.3.3 also provides some statistical evidence alluding to the existence of a complex relationship between clonotype frequency and DoC, akin to statistical observations made in complex systems that exhibit self-organisation and fractal geometry. We scale up these analyses in section 3.3.9 and argue the observed Gompertzian dynamics are likely to point to the existence of a fractal-stochastic dualism between individual repertoire's clonotype frequency distributions and population-wide clonotype frequency distribution over DoC. These results provide some evidence for the Blind Mapmaker hypothesis, in that, most of the public clonotypes are found in small numbers because they are the starting points for developing immune responses which diverge in similarity to the public clonotypes and grow in numbers through the affinity maturation process, resulting in private clonotypes found in large numbers in each individual. Finally, while the deep learning models do not provide direct evidence for this hypothesis, their success, in the face of the failure of the simpler approaches which do not utilise genomic sequence data (see section 3.3.10), suggests that the complex genomic features might define the commonality of immunoglobulins.

It is important to consider the function of any complex, particularly multi-cellular, biological system in multiple levels of the organisation. For instance, we can think of the earth's entire ecosystem as the top level of organisation followed by other lower levels in the hierarchy, such as local ecosystems, a population of a species in that local ecosystem, an individual with that population and lastly (in this analogy), the cells within that multi-cellular individual. We could go deeper by considering the "Selfish Gene" view of life or even deeper when considering self-organisation at lower levels, such as dynamic intracellular patterning of molecules and proteins vital for biological function[242] or even protein folding.

Similarly, a systems immunology view of immunity follows a hierarchical level that operates at all levels with feedback loops across all levels of the organisation. For instance, consider the following simplified and limited description of B-cell-specific aspects of immunity. At the lowest level of organisation, we have the biased stochastic processes of VDJ recombination and insertion/deletion events introduced by AID. We can consider B cells as the lowest-level agents in this hierarchy, which dynamically respond to their environment by evolution and adaptation. We can also consider the resulting lineage of B-cell clonotypes as a higher-level organisation of the agents. Next, we can consider each individual's repertoire, a collection of the clusters of clonotypes, as an agent in the population biology context and each local population as an immune agent in the context of the survival of a species. As with other biological systems, in this context, all levels of the organisation and/or agents have a unified global objective of their own, which, notably, has a loose definition, i.e. survival and growth, which introduces adversarial and cooperative game theoretical dynamics among the agents. Agents' survival is often constrained by, or coupled with, satisfying some objective functions necessary for the survival and growth of the other levels of the organisation, i.e. the system as a whole.

At the lower levels of organisation, there is a demand for agents, such as the AID enzyme and other elements involved, to generate enough diversity so the system can have a sufficiently diverse repertoire of agents to result in a repertoire that is not only robust, but also anti-fragile[243] against the vast and chaotic space of pathogenic epitopes. At a higher level, agents (B cells or clonotypes) must satisfy several objectives, e.g. non-autoreactivity and some degree of affinity or avidity for antigens, to survive. The higher the repertoire of epitopes a B cell can bind, the higher the likelihood of its growth; in other words, the higher its diversity of response, the higher its chances of survival and expansion. At the next level, the survival of an individual is determined by the diversity of pathogens and diseases it can survive and, by extension, the diversity of the repertoires it can produce. Finally, the survival of a species depends on the diversity of the pathogens it can withstand across space and time.

Here, agents' behaviour at all levels of the organisation directly or indirectly affects lower and higher levels of organisation, but notably, encodes information within the system as a consequence of its actions or changes in its state. A prominent example of this is the development of high-affinity B cells into memory cells. Though, such examples need not always be so specific; for instance, the general trends in the selection of B cells can affect the bias in the VDJ usage and recombinations, which can be fixed into the genomes with a population, or even the species, as we observe a universal pattern in VDJ usage biases. As part of the Blind Mapmaker hypothesis, I argue that the evolutionary and population dynamics, acting for the global objective of species' survival, regulate self-organisation at the different levels, which results in the emergence of the diversity of clonotypes with the complex and fractal economics that we observe. Such fractal distributions/structures which arise in economics are the hallmarks of self-organisation in the underlying complex systems[244], and similarly, numerously observed in the organisation of complex biological systems[245].

Such a decentralised dynamic without an explicit control mechanism or a global design blueprint is the hallmark of self-organisation in biological systems and, recently, in successful applications of robotics, for creating artificially intelligent agents which exhibit self-guided emergent behaviour and intelligence that resembles natural intelligence[246]. Perhaps the most famous examples come from reinforcement learning, with the notable examples of AlphaZero[247] and AlphaStar[248], where we observe agents exhibiting game-theoretical dynamics among other emergent phenomena observed at different levels of organisation. It is important to note that survival acting as the global objective is not synonymous with a task-specific global objective or control mechanism, which we can argue, for this specific case of the immune system would be neutralising the maximum number of epitopes possible as effectively and rapidly as possible. As a result, by operating through decentralised complex evolutionary dynamics, the immune system may reach specific attractors as an emergent phenomenon[246], optimising species' survival. I argue that such attractors facilitate the mapping of the epitope space effectively enough to grant the efficient and time-critical immune response required for survival.

For instance, we know how limited the empirical diversity of our B cells is compared to the theoretical space of possible epitopes, most of which, admittedly, may not ever materialise for a variety of reasons (e.g. thermodynamic unfeasibility) and most of the remaining space may not be in circulation at any one point in time. Nonetheless, the immune system has no extrinsic knowledge of this space that it has to navigate for survival, which would be fraught with events known as "black swans" - a term describing highly unpredictable and rare occurrences with significant consequences - much like most non-linear dynamical systems[249]. In fact, the long-tail distributions we observed in the clonotypes frequency distributions in and across repertoires display the existence of black swan responses both in individual and population immunity. Furthermore, it is crucial to survival that immune responses are swift enough to counteract the rapid, often exponential, growth of pathogens and diseases. Indeed we know that primary immune responses are almost always fast enough but, most significantly, consistent. Lastly, we must consider the peril many organisms face by expanding the repertoire of their pathogens through migration to different niches. The success of adaptive immunity in managing all such risks and challenges raises the question; "how can such material and temporal efficiency be achieved in the face of so much adversity?”. As an answer, I proposed the Blind Mapmaker hypothesis, which in a nutshell, suggests that the observed convergence across repertoires is an emergent phenomenon of dynamics discussed here, which grants an immunity capable of greater coverage than could be provided solely by bottom-up mechanics, such as the clonal selection principles or top-down regulation.

Inherently, the immune system has to tackle a multi-objective optimisation problem, and it is difficult to ignore the relevance of its self-organising mechanisms to artificial algorithms and intelligence systems, particularly to the recent fields in optimisation theory such as EDMO, which employ dynamic evolutionary mechanisms for solving dynamical multi-objective problems. Furthermore, the short- and long-term explore-exploit tendencies in generating the necessary diversity of adaptive immunity are commonplace in multi-objective optimisation. Similarly, the fractal organisation of the clonotypes with and across repertoires of a population resembles the principles, such as Pareto-efficiency, used to describe the dynamics behind economic inequality. Curiously, there is a close relationship between Pareto-domination and multi-objective optimisation problems where no global optima can satisfy all possible objectives without harming others, whereby emergent attractors often arise as a consequence of self-organised agents, as discussed. Instead, a set of feasible solutions function as attractors, which can be argued cooperatively to solve a global objective, resembling the coverage of the epitope space provided by the attractor clonotypes proposed in the Blind Mapmaker hypothesis. Subsequently, we see an intricate distribution of “a wealth of agents” with adaptive behaviour that collectively contribute to immunity across levels of organisation.

As discussed in the results, it is important to note that this hypothesis only aims to describe what might underpin convergence in adaptive immune repertoires and is only partially supported by the results of the original research in this thesis. Moreover, the convergence we observe in our results could come from many sources, e.g. immune memory or response to circulating pathogenic epitopes, in addition to the so-called attractor Convergent Cluster introduced in section 3.1.2. Rigorous future research is required to continue what is proposed and supported in this work. However, it is important to note that the current methodologies may not be sufficient for testing the existence, and analysis, of these attractor clonotypes, which I proposed to be multi-specific to maximise their utility and, as a result, survival. For instance, recent simulations show that “tug-of-war” dynamics may be present among B cells competing for antigen binding. This could result in different affinity optimisation dynamics rather than maximising affinity, which questions the current methodologies only measuring dissociation equilibrium[207] to examine immunoglobulin responses. One alternative way of testing may be provided through longitudinal repertoire studies, whereby we examine the persistence of high DoC/RIIM clonotypes across time, which, if positive, could provide one of the pieces of evidence necessary for examining this hypothesis.

In this research, we have provided an extensive statistical summarisation of repertoires, particularly in the context of the degree of commonality of immunoglobulin molecules within a human population. We demonstrate that the frequency distribution of clonotypes over the degree of commonality follows complicated patterns, pointing to complex underlying evolutionary and immunological dynamics that shape such patterns. We demonstrate that the collective summary statistics and frequency distribution are not sufficient for accurate prediction of the DoC. However, by incorporating the knowledge about the frequency of the clonotypes with their DoC, we introduce RIIM as a new and more granular measure of DoC, which we believe could bring researchers in this field a step closer to unravelling the underlying dynamics that shape the complexity of DoC. We used this knowledge in informing our main research focus, namely developing DNNs for predicting DoC. We designed a variety of DNN models for predicting DoC in various formats, i.e. as classification and regression tasks, to predict DoC when only trained on BCR amino acid sequence data. Our best models tolerate the extreme long-tailed imbalance and asymmetric noise across the labels/targets and successfully predict DoC (see section 3.3). Particularly, *Figures 3-39* and *3-40*, demonstrate our models' effectiveness in predicting RIIM (MAE: 0.082), providing an argument for the existence of a possible relationship between BCR sequences and RIIM, in addition to demonstrating the effectiveness of deep learning models for this task, as most errors are within a reasonable distance from the ground truth DoC. These figures show that the error distributions for predicting RIIM are homoscedastic – where the variability of residuals stays the same across the DoC - as evidence for consistency in the models' learning, generalisability and possible usefulness for detecting mislabeled data. Additionally, the fact that two different neural network architectures display similar homoscedastic error distribution patterns may be further evidence that some of the models' errors are inaccuracies in the context of our limited training dataset with 13 subjects, rather than the wider human population. Nonetheless, further investigations are required to assess whether all errors are genuine, for instance through deliberate, but careful, mislabelling of a small number of test cases or updating the test set with more subjects. Such a study, if shown to support this hypothesis, could demonstrate the power and high impact of our approach, when a large sample size (i.e. concerning the number of subjects) is such a scarcity in BCR repertoire datasets.

Bibliography

1. Abbas, A. K. & Lichtman, A. Cellular and molecular immunology. Preprint at (2005).

2. Murphy, K. & Weaver, C. *Janeway's Immunobiology*.

3. Cobey, S., Wilson, P. & Matsen, F. A. The evolution within us. *Philosophical Transactions of the Royal Society B: Biological Sciences* Preprint at https://doi.org/10.1098/rstb.2014.0235 (2015).

4. Sarkizova, S. & Hacohen, N. Systems Immunology : Learning the Rules of the Immune System. doi:10.1146/annurev-immunol-042617-053035.

5. Kovaltsuk, A. *et al.* How B-Cell Receptor Repertoire Sequencing Can Be Enriched with Structural Antibody Data. *Front. Immunol.* **8**, 1753 (2017).

6. Sheng, Z. *et al.* Effects of Darwinian Selection and Mutability on Rate of Broadly Neutralizing Antibody Evolution during HIV-1 Infection. *PLoS Comput. Biol.* **12**, e1004940 (2016).

7. Greiff, V. *et al.* A bioinformatic framework for immune repertoire diversity profiling enables detection of immunological status. *Genome Med.* (2015) doi:10.1186/s13073-015-0169-8.

8. Elhanati, Y. *et al.* Inferring processes underlying B-cell repertoire diversity. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* (2015) doi:10.1098/rstb.2014.0243.

9. Sepúlveda, N., Paulino, C. D. & Carneiro, J. Estimation of T-cell repertoire diversity and clonal size distribution by Poisson abundance models. *J. Immunol. Methods* **353**, 124–137 (2010).

10. Gibson, K. L. *et al.* B-cell diversity decreases in old age and is correlated with poor health status. *Aging Cell* **8**, 18–25 (2009).

11. Giudicelli, V. & Lefranc, M.-P. IMGT/JunctionAnalysis: IMGT Standardized Analysis of the V-J and V-D-J Junctions of the Rearranged Immunoglobulins (IG) and T Cell Receptors (TR). *Cold Spring Harb. Protoc.* **2011**, db.prot5634-pdb.prot5634 (2011).

12. Wagner, G. P. & Altenberg, L. Perspective: Complex adaptations and the evolution of evolvability. *Evolution* **50**, 967 (1996).

13. Wagner, A. Robustness and evolvability in living systems. (2013).

14. Miho, E., Roškar, R., Greiff, V. & Reddy, S. T. Large-scale network analysis reveals the sequence space architecture of antibody repertoires. *Nat. Commun.* **10**, 1321 (2019).

15. Adams, R. M., Kinney, J. B., Walczak, A. M. & Mora, T. Epistasis in a fitness landscape defined by antibody-antigen binding free energy. *Cell Syst.* **8**, 86-93.e3 (2019).

16. Briney, B., Inderbitzin, A., Joyce, C. & Burton, D. R. Commonality despite exceptional diversity in the baseline human antibody repertoire. *Nature* **566**, 393–397 (2019).

17. Soto, C. *et al.* High frequency of shared clonotypes in human. *Nature* (2019).

18. Greiff, V. *et al.* Learning the High-Dimensional Immunogenomic Features That Predict Public and Private Antibody Repertoires. *The Journal of Immunology* **199**, 2985–2997 (2017).

19. Edelman, G. M. & Gally, J. A. Degeneracy and complexity in biological systems. *Proc. Natl. Acad. Sci. U. S. A.* **98**, 13763–13768 (2001).

20. Reuter, J. A., Spacek, D. V. & Snyder, M. P. High-throughput sequencing technologies. *Mol. Cell* **58**, 586–597 (2015).

21. Park, S.-J., Saito-Adachi, M., Komiyama, Y. & Nakai, K. Advances, practice, and clinical perspectives in high-throughput sequencing. *Oral Dis.* **22**, 353–364 (2016).

22. Pervaiz, T. *et al.* High Throughput Sequencing Advances and Future Challenges. *Journal of Plant Biochemistry & Physiology* **05**, (2017).

23. Morey, M. *et al.* A glimpse into past, present, and future DNA sequencing. *Molecular Genetics and Metabolism* **110**, 3–24 (2013).

24. Chiu, R. W. K. *et al.* Non-invasive prenatal assessment of trisomy 21 by multiplexed maternal plasma DNA sequencing: large scale validity study. *BMJ* **342**, c7401 (2011).

25. Goodwin, S., McPherson, J. D. & McCombie, W. R. Coming of age: ten years of next-generation sequencing technologies. *Nat. Rev. Genet.* **17**, 333–351 (2016).

26. Metzker, M. L. Sequencing technologies the next generation. *Nat. Rev. Genet.* **11**, 31–46 (2010).

27. Buermans, H. P. J. & den Dunnen, J. T. Next generation sequencing technology: Advances and applications. *Biochimica et Biophysica Acta - Molecular Basis of Disease* **1842**, 1932–1941 (2014).

28. McGillivray, P. *et al.* Network Analysis as a Grand Unifier in Biomedical Data Science. *Annual Review of Biomedical Data Science* **1**, annurev-biodatasci--080917--013444 (2018).

29. Feng, Y., Zhang, Y., Ying, C., Wang, D. & Du, C. Nanopore-based fourth-generation DNA sequencing technology. *Genomics Proteomics Bioinformatics* **13**, 4–16 (2015).

30. Katze, M. G., Korth, M. J. & Law, G. L. *From Basics to Systems Biology Viral Pathogenesis*.

31. Laserson, U. *et al.* High-resolution antibody dynamics of vaccine-induced immune responses. *Proc. Natl. Acad. Sci. U. S. A.* **111**, 4928–4933 (2014).

32. Yermanos, A. *et al.* Comparison of methods for phylogenetic B-cell lineage inference using time-resolved antibody repertoire simulations (AbSim). *Bioinformatics* **33**, 3938–3946 (2017).

33. Vollmers, C., Sit, R. V., Weinstein, J. A., Dekker, C. L. & Quake, S. R. Genetic measurement of memory B-cell recall using antibody repertoire sequencing. *Proc. Natl. Acad. Sci. U. S. A.* **110**, 13463–13468 (2013).

34. Boyd, S. D. *et al.* Measurement and clinical monitoring of human lymphocyte clonality by massively parallel V-D-J pyrosequencing. *Sci. Transl. Med.* **1**, (2009).

35. Chaudhary, N. & Wesemann, D. R. Analyzing Immunoglobulin Repertoires. *Front. Immunol.* **9**, (2018).

36. Liu, H. *et al.* The methods and advances of adaptive immune receptors repertoire sequencing. *Theranostics* **11**, 8945–8963 (2021).

37. Rizzo, J. M. & Buck, M. J. Key principles and clinical applications of "next-generation" DNA sequencing. *Cancer Prev. Res.* **5**, 887–900 (2012).

38. Lau, D., Bobe, A. M. & Khan, A. A. RNA Sequencing of the Tumor Microenvironment in Precision Cancer Immunotherapy. *Trends Cancer Res.* **5**, 149–156 (2019).

39. Georgiou, G. *et al.* The promise and challenge of high-throughput sequencing of the antibody repertoire. *Nat. Biotechnol.* **32**, 158–168 (2014).

40. He, L. *et al.* Toward a more accurate view of human B-cell repertoire by next-generation sequencing, unbiased repertoire capture and single-molecule barcoding. *Sci. Rep.* **4**, 6778 (2014).

41. Khan, T. A. *et al.* Accurate and predictive antibody repertoire profiling by molecular amplification fingerprinting. *Science Advances* **2**, e1501371 (2016).

42. Davidsen, K. & Matsen, F. Benchmarking tree and ancestral sequence inference for B cell receptor sequences. *bioRxiv* 307736 (2018).

43. Chovanec, P. *et al.* Unbiased quantification of immunoglobulin diversity at the DNA level with VDJ-seq. *Nat. Protoc.* **13**, 1232–1252 (2018).

44. Friedensohn, S., Khan, T. A. & Reddy, S. T. Advanced Methodologies in High-Throughput Sequencing of Immune Repertoires. *Trends Biotechnol.* **35**, 203–214 (2017).

45. Lebrigand, K., Magnone, V., Barbry, P. & Waldmann, R. High throughput error corrected Nanopore single cell transcriptome sequencing. *Nat. Commun.* **11**, 1–8 (2020).

46. Lowden, M. J. & Henry, K. A. Oxford nanopore sequencing enables rapid discovery of single-domain antibodies from phage display libraries. *Biotechniques* **65**, 351–356 (2018).

47. Christley, S. *et al.* VDJServer: A Cloud-Based Analysis Portal and Data Commons for Immune Repertoire Sequences and Rearrangements. *Front. Immunol.* **9**, (2018).

48. Vander Heiden, J. A. *et al.* PRESTO: A toolkit for processing high-throughput sequencing raw reads of lymphocyte receptor repertoires. *Bioinformatics* **30**, 1930–1932 (2014).

49. Gupta, N. T. *et al.* Change-O: A toolkit for analyzing large-scale B cell immunoglobulin repertoire sequencing data. *Bioinformatics* **31**, 3356–3358 (2015).

50. Pommi, C., Levadoux, S., Sabatier, R., Lefranc, G. & Lefranc, M.-P. IMGT standardized criteria for statistical analysis of immunoglobulin V-REGION amino acid properties. *Journal of molecular recognition : JMR* **17**, 17–32.

51. Lefranc, M.-P. *et al.* IMGT®, the international ImMunoGeneTics information system® 25 years on. *Nucleic Acids Res.* **43**, D413-22 (2015).

52. Ralph, D. K. & Matsen, F. A. Consistency of VDJ Rearrangement and Substitution Parameters Enables Accurate B Cell Receptor Sequence Annotation. *PLoS Comput. Biol.* (2016) doi:10.1371/journal.pcbi.1004409.

53. Rubelt, F. *et al.* Adaptive Immune Receptor Repertoire Community recommendations for sharing immune-repertoire sequencing data. *Nat. Immunol.* **18**, 1274–1278 (2017).

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

55. Miho, E. *et al.* Computational strategies for dissecting the high-dimensional complexity of adaptive immune repertoires. *Front. Immunol.* (2018) doi:10.3389/fimmu.2018.00224.

56. Miho, E., Greiff, V., Roškar, R. & Reddy, S. T. The fundamental principles of antibody repertoire architecture revealed by 1 large-scale network analysis 2. doi:10.1101/124578.

57. Bashford-Rogers, R. J. M. *et al.* Network properties derived from deep sequencing of human b-cell receptor repertoires delineate b-cell populations. *Genome Res.* (2013) doi:10.1101/gr.154815.113.

58. Ben-Hamo, R. & Efroni, S. The whole-organism heavy chain B cell repertoire from Zebrafish self-organizes into distinct network features. *BMC Syst. Biol.* (2011) doi:10.1186/1752-0509-5-27.

59. Chang, Y.-H. *et al.* Network Signatures of IgG Immune Repertoires in Hepatitis B Associated Chronic Infection and Vaccination Responses. *Sci. Rep.* **6**, 26556 (2016).

60. Madi, A. *et al.* T cell receptor repertoires of mice and humans are clustered in similarity networks around conserved public CDR3 sequences. *Elife* (2017) doi:10.7554/eLife.22057.

61. Csurdi, G. & Nepusz, T. The igraph software package for complex network research.

62. Bastian, M., Heymann, S. & Jacomy, M. Gephi: An Open Source Software for Exploring and Manipulating Networks. *Third International AAAI Conference on Weblogs and Social Media* 361–362 (2009).

63. Schult, D. A. LA-UR- Exploring network structure, dynamics, and function using NetworkX.

64. Shannon, P. *et al.* Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* **13**, 2498–2504 (2003).

65. Yokota, R., Kaminaga, Y. & Kobayashi, T. J. Quantification of inter-sample differences in T cell receptor sequences. *bioRxiv* 128025 (2017).

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

67. Dash, P. *et al.* Quantifiable predictive features define epitope-specific T cell receptor repertoires. *Nature* (2017) doi:10.1038/nature22383.

68. Baldi, P. Deep Learning in Biomedical Data Science. 181–205 (2018).

69. Min, S., Lee, B. & Yoon, S. Deep learning in bioinformatics. *Brief. Bioinform.* bbw068 (2016).

70. Ching, T. *et al.* Opportunities And Obstacles For Deep Learning In Biology And Medicine. *J. R. Soc. Interface* (2017) doi:10.1101/142760.

71. Rampasek, L. & Goldenberg, A. TensorFlow: Biology's Gateway to Deep Learning? *Cell Systems* **2**, 12–14 (2016).

72. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **61**, 85–117 (2015).

73. Mamoshina, P., Vieira, A., Putin, E. & Zhavoronkov, A. Applications of Deep Learning in Biomedicine. *Mol. Pharm.* **13**, 1445–1454 (2016).

74. Introduction to artificial neural networks. *Proceedings Electronic Technology Directions to the Year 2000 ETD-95* 36–62 Preprint at https://doi.org/10.1109/ETD.1995.403491 (1995).

75. Buduma, N. & Locascio, N. *Deep Learning*. 775 (2017).

76. Bengio, Y. & Lee, H. Editorial introduction to the Neural Networks special issue on Deep Learning of Representations. *Neural Netw.* **64**, 1–3 (2015).

77. Deng, L. Deep Learning: Methods and Applications. *Found. Signal. Process. Commun. Netw.* **7**, 197–387 (2014).

78. Serra, A., Galdi, P. & Tagliaferri, R. Machine learning for bioinformatics and neuroimaging. *Wiley Interdiscip. Rev. Data Min. Knowl. Discov.* 1–33 (2018).

79. Baldi, P. F. & Hornik, K. Learning in linear neural networks: a survey. *IEEE Trans. Neural Netw.* **6**, 837–858 (1995).

80. Lecun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* vol. 521 436–444 Preprint at https://doi.org/10.1038/nature14539 (2015).

81. Thomas, N. *et al.* Tracking global changes induced in the CD4 T-cell receptor repertoire by immunization with a complex antigen using short stretches of CDR3 protein sequence. *Bioinformatics* **30**, 3181–3188 (2014).

82. Jaderberg, M. *et al.* Reinforcement Learning with Unsupervised Auxiliary Tasks. 1–14 (2016).

83. Denas, O. & Taylor, J. Deep modeling of gene expression regulation in an Erythropoiesis model. *ICML Workshop on Representation Learning* (2013).

84. Telenti, A., Lippert, C., Chang, P.-C. & DePristo, M. Deep learning of genomic variation and regulatory network data. *Hum. Mol. Genet.* **27**, R63–R71 (2018).

85. P. Murphy, K. *Machine Learning: A Probabilistic Perspective*. (1991).

86. Kaneko, T. Generative adversarial networks: Foundations and applications. **3**, 189–197 (2018).

87. Le, Q. V. *et al.* Building high-level features using large scale unsupervised learning. *arXiv preprint arXiv:1112. 6209* 81–88 (2011).

88. Hecht-Nielsen, R. *Neurocomputing*. 433 (Addison-Wesley Pub. Co, 1990).

89. Abadi, M. *et al.* TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems. (2016).

90. Stephens, Z. D. *et al.* Big Data: Astronomical or Genomical? *PLoS Biol.* **13**, e1002195 (2015).

91. Jain, A. K., Mao, J. & Mohiuddin, K. M. Artificial neural networks: A tutorial. Preprint at (1996).

92. Muller, B., Reinhardt, J. (joachim) & Strickland, M. T. (michael T. *Neural networks : an introduction*. (Springer, 1995).

93. Bishop, C. M. Curvature-driven smoothing: a learning algorithm for feedforward networks. *IEEE Trans. Neural Netw.* **4**, 882–884 (1993).

94. Baldi, P. Gradient descent learning algorithm overview: a general dynamical systems perspective. *IEEE Trans. Neural Netw.* **6**, 182–195 (1995).

95. Salakhutdinov, R. & Hinton, G. An efficient learning procedure for deep Boltzmann machines. *Neural Comput.* **24**, 1967–2006 (2012).

96. Hoang, Q., Nguyen, T. D., Le, T. & Phung, D. Multi-Generator Generative Adversarial Nets. 1–23 (2017).

97. Poernomo, A. & Kang, D.-K. Biased Dropout and Crossmap Dropout: Learning towards effective dropout regularization in convolutional neural network. *Neural Netw.* (2018) doi:10.1016/j.neunet.2018.03.016.

98. Widrow, B. & Lehr, M. A. 30 years of adaptive neural networks: perceptron, Madaline, and backpropagation. *Proc. IEEE* **78**, 1415–1442 (1990).

99. Qian, Q., Jin, R., Yi, J., Zhang, L. & Zhu, S. Efficient distance metric learning by adaptive sampling and mini-batch stochastic gradient descent (SGD). *Mach. Learn.* **99**, 353–372 (2014).

100. Ioffe, S. & Szegedy, C. Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. (2015).

101. Masters, D. & Luschi, C. Revisiting Small Batch Training for Deep Neural Networks. 1–18 (2018).

102. Hassanzadeh, H. R. & Wang, M. D. DeeperBind: Enhancing prediction of sequence specificities of DNA binding proteins. in *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)* 178–183 (IEEE, 2016).

103. Egmont-Petersen, M., de Ridder, D. & Handels, H. Image processing with neural networks—a review. *Pattern Recognit.* **35**, 2279–2301 (2002).

104. Angermueller, C., PU+00e4rnamaa, T., Parts, L. & Stegle, O. Deep learning for computational biology. *Molecular systems biology* **12**, 878 (2016).

105. Baker, B., Gupta, O., Naik, N. & Raskar, R. Designing Neural Network Architectures Using Reinforcement Learning. *Proc. of the 5th International Conference on Learning Representations* 1–18 (2017).

106. Goodfellow, I. J., Bulatov, Y., Ibarz, J., Arnoud, S. & Shet, V. Multi-digit Number Recognition from Street View Imagery using Deep Convolutional Neural Networks. (2013).

107. Connor, J. T., Martin, R. D. & Atlas, L. E. Recurrent neural networks and robust time series prediction. *IEEE Trans. Neural Netw.* **5**, 240–254 (1994).

108. Atiya, A. F. & Parlos, A. G. New results on recurrent network training: unifying the algorithms and accelerating convergence. *IEEE Trans. Neural Netw.* **11**, 697–709 (2000).

109. Pham, V., Bluche, T., Kermorvant, C. & Louradour, J. Dropout improves Recurrent Neural Networks for Handwriting Recognition. (2013).

110. Lipton, Z. C., Berkowitz, J. & Elkan, C. A Critical Review of Recurrent Neural Networks for Sequence Learning. (2015).

111. Lipton, Z. C., Kale, D. C., Elkan, C. & Wetzel, R. Learning to Diagnose with LSTM Recurrent Neural Networks. (2015).

112. Williams, R. J. & Zipser, D. A Learning Algorithm for Continually Running Fully Recurrent Neural Networks. *Neural Comput.* **1**, 270–280 (1989).

113. Sutskever, I., Vinyals, O. & Le, Q. V. Sequence to Sequence Learning with Neural Networks. (2014).

114. KoutnU+00edk, J., Greff, K., Gomez, F. & Schmidhuber, J. A Clockwork RNN. (2014).

115. Hochreiter, S. & Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **9**, 1735–1780 (1997).

116. Beringer, N., Graves, A., Schiel, F. & Schmidhuber, J. Classifying Unprompted Speech by Retraining LSTM Nets. in 575–581 (2005).

117. Chollet, F. *Deep Learning with Python, Second Edition*. (Simon and Schuster, 2021).

118. Goodfellow, I. J. *et al.* Generative Adversarial Networks. (2014).

119. Nadkarni, P. M., Ohno-Machado, L. & Chapman, W. W. Natural language processing: an introduction. *J. Am. Med. Inform. Assoc.* **18**, 544–551 (2011).

120. Young, T., Hazarika, D., Poria, S. & Cambria, E. Recent Trends in Deep Learning Based Natural Language Processing. *arXiv [cs.CL]* (2017).

121. Jumper, J. *et al.* Applying and improving AlphaFold at CASP14. *Proteins* (2021) doi:10.1002/prot.26257.

122. Jumper, J. *et al.* Highly accurate protein structure prediction with AlphaFold. *Nature* **596**, 583–589 (2021).

123. Tunyasuvunakool, K. *et al.* Highly accurate protein structure prediction for the human proteome. *Nature* **596**, 590–596 (2021).

124. Mikolov, T., Chen, K., Corrado, G. & Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv [cs.CL]* (2013).

125. Joulin, A., Grave, E., Bojanowski, P. & Mikolov, T. Bag of Tricks for Efficient Text Classification. *arXiv [cs.CL]* (2016).

126. Bojanowski, P., Grave, E., Joulin, A. & Mikolov, T. Enriching Word Vectors with Subword Information. *arXiv [cs.CL]* (2016).

127. Palangi, H. *et al.* Deep Sentence Embedding Using Long Short-Term Memory Networks: Analysis and Application to Information Retrieval. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* **24**, 694–707 (2016).

128. Melamud, O., Goldberger, J. & Dagan, I. context2vec: Learning Generic Context Embedding with Bidirectional LSTM. in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning* 51–61 (Association for Computational Linguistics, 2016).

129. Zuo, Y. *et al.* Short text classification based on bidirectional TCN and attention mechanism. *J. Phys. Conf. Ser.* **1693**, 012067 (2020).

130. Huang, J., Lu, C., Ping, G., Sun, L. & Ye, X. TCN-ATT: A Non-recurrent Model for Sequence-Based Malware Detection. in *Advances in Knowledge Discovery and Data Mining* 178–190 (Springer International Publishing, 2020).

131. Choromanski, K. *et al.* Rethinking Attention with Performers. *arXiv [cs.LG]* (2020).

132. Vaswani, A. *et al.* Attention Is All You Need. *arXiv [cs.CL]* (2017).

133. Letarte, G., Paradis, F., Giguère, P. & Laviolette, F. Importance of Self-Attention for Sentiment Analysis. in *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP* 267–275 (Association for Computational Linguistics, 2018).

134. Shaw, P., Uszkoreit, J. & Vaswani, A. Self-Attention with Relative Position Representations. *arXiv [cs.CL]* (2018).

135. Alipanahi, B., Delong, A., Weirauch, M. T. & Frey, B. J. Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning. *Nat. Biotechnol.* **33**, 831–838 (2015).

136. Vincent, P., Larochelle, H., Bengio, Y. & Manzagol, P.-A. Extracting and composing robust features with denoising autoencoders. in *Proceedings of the 25th international conference on Machine learning - ICML '08* 1096–1103 (ACM Press, 2008).

137. Way, G. P. & Greene, C. S. Evaluating deep variational autoencoders trained on pan-cancer gene expression. (2017).

138. Hinton, G. E. & Salakhutdinov, R. R. Reducing the dimensionality of data with neural networks. *Science* **313**, 504–507 (2006).

139. Baldi, P. Autoencoders, Unsupervised Learning, and Deep Architectures. *ICML Unsupervised and Transfer Learning* 37–50 (2012).

140. Chen, L., Cai, C., Chen, V. & Lu, X. Learning a hierarchical representation of the yeast transcriptomic machinery using an autoencoder model. *BMC Bioinformatics* **17**, S9 (2016).

141. Che, Z., Purushotham, S., Khemani, R. & Liu, Y. Distilling Knowledge from Deep Networks with Applications to Healthcare Domain. (2015).

142. Yi, H. C. *et al.* A Deep Learning Framework for Robust and Accurate Prediction of ncRNA-Protein Interactions Using Evolutionary Information. *Molecular Therapy - Nucleic Acids* **11**, 337–344 (2018).

143. Ghahramani, A., Watt, F. M. & Luscombe, N. M. Generative adversarial networks uncover epidermal regulators and predict single cell perturbations. *bioRxiv* 262501 (2018).

144. Clauwaert, J., Menschaert, G. & Waegeman, W. Explainability in transformer models for functional genomics. *Brief. Bioinform.* **22**, (2021).

145. Song, Z. *et al.* Attention-based multi-label neural networks for integrated prediction and interpretation of twelve widely occurring RNA modifications. *Nat. Commun.* **12**, 4011 (2021).

146. Widrich, M. *et al.* DeepRC: Immune repertoire classification with attention-based deep massive multiple instance learning. *bioRxiv* 2020.04.12.038158 (2020) doi:10.1101/2020.04.12.038158.

147. Ramsauer, H. *et al.* HOPFIELD NETWORKS IS ALL YOU NEED. *arXiv [cs.NE]* (2020).

148. Elnaggar, A. *et al.* ProtTrans: Towards cracking the language of lifes code through self-supervised deep learning and high performance computing. *IEEE Trans. Pattern Anal. Mach. Intell.* **PP**, 1–1 (2021).

149. Wu, K. E. *et al.* TCR-BERT: learning the grammar of T-cell receptors for flexible antigen-binding analyses. *bioRxiv* 2021.11.18.469186 (2021) doi:10.1101/2021.11.18.469186.

150. Olsen, T. H., Moal, I. H. & Deane, C. M. AbLang: An antibody language model for completing antibody sequences. *bioRxiv* 2022.01.20.477061 (2022) doi:10.1101/2022.01.20.477061.

151. Recht, B., Roelofs, R., Schmidt, L. & Shankar, V. Do CIFAR-10 Classifiers Generalize to CIFAR-10? (2018).

152. Olson, B. J. *et al.* sumrep: A Summary Statistic Framework for Immune Receptor Repertoire Comparison and Model Validation. *Front. Immunol.* **10**, 2533 (2019).

153. Fu, X. *et al.* High-throughput sequencing of the expressed Torafugu (Takifugu rubripes) antibody sequences distinguishes IgM and IgT repertoires and reveals evidence of convergent evolution. *Front. Immunol.* **9**, (2018).

154. Hong, B. *et al.* In-Depth Analysis of Human Neonatal and Adult IgM Antibody Repertoires. *Front. Immunol.* **9**, 128 (2018).

155. Galson, J. D. *et al.* In-Depth Assessment of Within-Individual and Inter-Individual Variation in the B Cell Receptor Repertoire. *Front. Immunol.* **6**, 531 (2015).

156. Mroczek, E. S. *et al.* Differences in the composition of the human antibody repertoire by b cell subsets in the blood. *Front. Immunol.* **5**, (2014).

157. Miqueu, P. *et al.* Statistical analysis of CDR3 length distributions for the assessment of T and B cell repertoire biases. *Mol. Immunol.* **44**, 1057–1064 (2007).

158. Larimore, K., McCormick, M. W., Robins, H. S. & Greenberg, P. D. Shaping of Human Germline IgH Repertoires Revealed by Deep Sequencing. *The Journal of Immunology* **189**, 3221–3230 (2012).

159. Ostmeyer, J. *et al.* Statistical classifiers for diagnosing disease from immune repertoires: a case study using multiple sclerosis. *BMC Bioinformatics* **18**, 401 (2017).

160. Cinelli, M. *et al.* Feature selection using a one dimensional naïve Bayes’ classifier increases the accuracy of support vector machine classification of CDR3 repertoires. *Bioinformatics* **33**, 951–955.

161. Nouri, N. & Kleinstein, S. H. Performance-optimized partitioning of clonotypes from high-throughput immunoglobulin repertoire sequencing data. *bioRxiv* 175315 (2017) doi:10.1101/175315.

162. Nouri, N. & Kleinstein, S. H. Somatic hypermutation analysis for improved identification of B cell clonal families from next-generation sequencing data. *PLoS Comput. Biol.* **16**, e1007977 (2020).

163. Vander Heiden, J. A. Computation Methods for the Analysis of B Cell Repertoires and Applications to Human Autoimmunity. (Yale UniversityProQuest Dissertations Publishing, 2017).

164. Gupta, N. T. *et al.* Hierarchical Clustering Can Identify B Cell Clones with High Confidence in Ig Repertoire Sequencing Data. *The Journal of Immunology* **198**, 2489–2499 (2017).

165. Wu, Y.-C. B., Kipling, D. & Dunn-Walters, D. K. Age-Related Changes in Human Peripheral Blood IGH Repertoire Following Vaccination. *Front. Immunol.* **3**, (2012).

166. Levin, M. *et al.* Persistence and evolution of allergen-specific IgE repertoires during subcutaneous specific immunotherapy. *J. Allergy Clin. Immunol.* **137**, 1535–1544 (2016).

167. Kovaltsuk, A. *et al.* Observed Antibody Space: A Resource for Data Mining Next-Generation Sequencing of Antibody Repertoires. *J. Immunol.* **201**, 2502–2509 (2018).

168. Ralph, D. K. & Matsen, F. A., 4th. Per-sample immunoglobulin germline inference from B cell receptor deep sequencing data. *PLoS Comput. Biol.* **15**, e1007133 (2019).

169. Ralph, D. K. & Matsen, F. A., 4th. Likelihood-Based Inference of B Cell Clonal Families. *PLoS Comput. Biol.* **12**, e1005086 (2016).

170. Sturges, H. A. The Choice of a Class Interval. *J. Am. Stat. Assoc.* **21**, 65–66 (1926).

171. Freedman, D. & Diaconis, P. On the histogram as a density estimator:L 2 theory. *Zeitschrift fur Wahrscheinlichkeitstheorie und Verwandte Gebiete* **57**, 453–476 (1981).

172. Scargle, J. D., Norris, J. P., Jackson, B. & Chiang, J. Studies in astronomical time series analysis. VI. Bayesian block representations. *Astrophys. J.* **764**, (2013).

173. Knuth, K. H. Optimal Data-Based Binning for Histograms. *arXiv [physics]* (2006).

174. The Astropy Collaboration *et al.* The Astropy Project: Building an inclusive, open-science project and status of the v2.0 core package. *arXiv [astro-ph.IM]* (2018) doi:10.3847/1538-3881/aabc4f.

175. Pedregosa, F. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).

176. Atchley, W. R., Zhao, J., Fernandes, A. D. & Drüke, T. Solving the protein sequence metric problem. *Proc. Natl. Acad. Sci. U. S. A.* **102**, 6395–6400 (2005).

177. Kidera, A., Konishi, Y., Oka, M., Ooi, T. & Scheraga, H. A. Statistical analysis of the physical properties of the 20 naturally occurring amino acids. *J. Protein Chem.* **4**, 23–55 (1985).

178. Ikai, A. Thermostability and aliphatic index of globular proteins. *J. Biochem.* **88**, 1895–1898 (1980).

179. Kyte, J. & Doolittle, R. F. A simple method for displaying the hydropathic character of a protein. *J. Mol. Biol.* **157**, 105–132 (1982).

180. Radzicka, A. & Wolfenden, R. Comparing the polarities of the amino acids: side-chain distribution coefficients between the vapor phase, cyclohexane, 1-octanol, and neutral aqueous solution. *Biochemistry* **27**, 1664–1670 (1988).

181. Goodfellow, I., Bengio, Y. & Courville, A. *Deep Learning*. (MIT Press, 2016).

182. Tian, T., Wan, J., Song, Q. & Wei, Z. Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nature Machine Intelligence* **1**, 191–198 (2019).

183. Graves, J. *et al.* A Review of Deep Learning Methods for Antibodies. *Antibodies (Basel)* **9**, (2020).

184. Davidsen, K. *et al.* Deep generative models for T cell receptor protein sequences. *Elife* **8**, (2019).

185. Collins, A. M. & Jackson, K. J. L. On being the right size: antibody repertoire formation in the mouse and human. *Immunogenetics* **70**, 143–158 (2018).

186. Schramm, C. A. & Douek, D. C. Beyond Hot Spots: Biases in Antibody Somatic Hypermutation and Implications for Vaccine Design. *Front. Immunol.* **9**, 1876 (2018).

187. Marcou, Q., Mora, T. & Walczak, A. M. High-throughput immune repertoire analysis with IGoR. *Nat. Commun.* **9**, 561 (2018).

188. Sethna, Z., Elhanati, Y., Callan, 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).

189. Crowe, J. E., Jr & Koff, W. C. Deciphering the human immunome. *Expert Rev. Vaccines* **14**, 1421–1425 (2015).

190. Setliff, I. *et al.* Multi-Donor Longitudinal Antibody Repertoire Sequencing Reveals the Existence of Public Antibody Clonotypes in HIV-1 Infection. *Cell Host Microbe* **23**, 845-854.e6 (2018).

191. Schmitz, A. J. *et al.* A vaccine-induced public antibody protects against SARS-CoV-2 and emerging variants. *Immunity* **54**, 2159-2166.e6 (2021).

192. Angeletti, D. & Yewdell, J. W. Understanding and Manipulating Viral Immunity: Antibody Immunodominance Enters Center Stage. *Trends Immunol.* **39**, 549–561 (2018).

193. Guthmiller, J. J. *et al.* A public broadly neutralizing antibody class targets a membrane-proximal anchor epitope of influenza virus hemagglutinin. *bioRxiv* 2021.02.25.432905 (2021) doi:10.1101/2021.02.25.432905.

194. Greiff, V. *et al.* Systems Analysis Reveals High Genetic and Antigen-Driven Predetermination of Antibody Repertoires throughout B Cell Development. *Cell Rep.* **19**, 1467–1478 (2017).

195. Raybould, M. I. J. *et al.* Public Baseline and shared response structures support the theory of antibody repertoire functional commonality. *PLoS Comput. Biol.* **17**, e1008781 (2021).

196. Xu, J. L. & Davis, M. M. Diversity in the CDR3 Region of VH Is Sufficient for Most Antibody Specificities. *Immunity* **13**, 37–45 (2000).

197. Desponds, J., Mora, T. & Walczak, A. M. Fluctuating fitness shapes the clone-size distribution of immune repertoires. *Proc. Natl. Acad. Sci. U. S. A.* **113**, 274–279 (2016).

198. Mora, T. & Walczak, A. M. How many different clonotypes do immune repertoires contain? *Current Opinion in Systems Biology* **18**, 104–110 (2019).

199. He, K., Zhang, X., Ren, S. & Sun, J. Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. *arXiv [cs.CV]* (2015).

200. Liu, L. *et al.* On the Variance of the Adaptive Learning Rate and Beyond. *arXiv [cs.LG]* (2019).

201. Zhang, M. R., Lucas, J., Hinton, G. & Ba, J. Lookahead Optimizer: k steps forward, 1 step back. *arXiv [cs.LG]* (2019).

202. Hastie, T., Friedman, J. & Tibshirani, R. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. (Springer, New York, NY, 2001).

203. Bishop, C. M. & Nasrabadi, N. M. Pattern Recognition and Machine Learning. (*SpringerLink*, 2006).

204. James, G., Witten, D., Hastie, T. & Tibshirani, R. *An Introduction to Statistical Learning: with Applications in R*. (Springer, New York, NY, 2021).

205. Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, Piotr Dollar. Focal Loss for Dense Object Detection.

206. Quigley, M. F. *et al.* Convergent recombination shapes the clonotypic landscape of the naïve T-cell repertoire. *Proceedings of the National Academy of Sciences* **107**, 19414–19419 (2010).

207. Jiang, H. & Wang, S. Molecular tug of war reveals adaptive potential of an immune cell repertoire. *arXiv [physics.bio-ph]* (2022).

208. Alstott, J., Bullmore, E. & Plenz, D. Powerlaw: a Python package for analysis of heavy-tailed distributions. *PLoS One* **9**, e85777 (2014).

209. van den Oord, A. *et al.* WaveNet: A Generative Model for Raw Audio. *arXiv [cs.SD]* (2016).

210. Hu, J., Shen, L., Albanie, S., Sun, G. & Wu, E. Squeeze-and-Excitation Networks. *arXiv [cs.CV]* (2017).

211. Yuan, L., Tay, F. E. H., Li, G., Wang, T. & Feng, J. Revisiting Knowledge Distillation via Label Smoothing Regularization. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* Preprint at https://doi.org/10.1109/cvpr42600.2020.00396 (2020).

212. Müller, R., Kornblith, S. & Hinton, G. When Does Label Smoothing Help? *arXiv [cs.LG]* (2019).

213. de la Torre, J., Puig, D. & Valls, A. Weighted kappa loss function for multi-class classification of ordinal data in deep learning. *Pattern Recognit. Lett.* **105**, 144–154 (2018).

214. Lemaitre, G. Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning.

215. Dutta, A., Dubey, T., Singh, K. K. & Anand, A. SpliceVec: Distributed feature representations for splice junction prediction. *Comput. Biol. Chem.* **74**, 434–441 (2018).

216. Schubach, M., Re, M., Robinson, P. N. & Valentini, G. Imbalance-Aware Machine Learning for Predicting Rare and Common Disease-Associated Non-Coding Variants. *Sci. Rep.* **7**, 2959 (2017).

217. Yoon, K. & Kwek, S. A data reduction approach for resolving the imbalanced data issue in functional genomics. *Neural Comput. Appl.* **16**, 295–306 (2007).

218. Mikołajczyk, A. & Grochowski, M. Data augmentation for improving deep learning in image classification problem. in *2018 International Interdisciplinary PhD Workshop (IIPhDW)* 117–122 (ieeexplore.ieee.org, 2018).

219. Perez, L. & Wang, J. The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv [cs.CV]* (2017).

220. Shorten, C. & Khoshgoftaar, T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data* **6**, 1–48 (2019).

221. Feng, S. Y. *et al.* A Survey of Data Augmentation Approaches for NLP. *arXiv [cs.CL]* (2021).

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

223. Gasteiger, E. *et al.* Protein identification and analysis tools on the ExPASy server. in *The Proteomics Protocols Handbook* 571–607 (Humana Press, 2005).

224. Janin, J. Surface and inside volumes in globular proteins. *Nature* **277**, 491–492 (1979).

225. Rose, G. D., Geselowitz, A. R., Lesser, G. J., Lee, R. H. & Zehfus, M. H. Hydrophobicity of amino acid residues in globular proteins. *Science* **229**, 834–838 (1985).

226. M. O. Dayhoff, R. M. S. Chapter 22: A model of evolutionary change in proteins. in *in Atlas of Protein Sequence and Structure* (1978).

227. Guruprasad, K., Reddy, B. V. & Pandit, M. W. Correlation between stability of a protein and its dipeptide composition: a novel approach for predicting in vivo stability of a protein from its primary sequence. *Protein Eng.* **4**, 155–161 (1990).

228. Miyazawa, S. & Jernigan, R. L. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules* **18**, 534–552 (1985).

229. Hopp, T. P. & Woods, K. R. Prediction of protein antigenic determinants from amino acid sequences. *Proc. Natl. Acad. Sci. U. S. A.* **78**, 3824–3828 (1981).

230. Zimmerman, J. M., Eliezer, N. & Simha, R. The characterization of amino acid sequences in proteins by statistical methods. *J. Theor. Biol.* **21**, 170–201 (1968).

231. Tabb, D. L. An algorithm for isoelectric point estimation. http://fields.scripps.edu/DTASelect/20010710-pI-Algorithm.pdf.

232. Meng, X. *et al.* MLlib: Machine Learning in Apache Spark. *arXiv [cs.LG]* (2015).

233. Zaharia, M. *et al.* Apache Spark: a unified engine for big data processing. *Commun. ACM* **59**, 56–65 (2016).

234. Zaharia, M., Chowdhury, M., Franklin, M. J. & Shenker, S. Spark: Cluster computing with working sets. https://www.usenix.org/legacy/event/hotcloud10/tech/full_papers/Zaharia.pdf.

235. Zaharia, M. *et al.* Resilient Distributed Datasets: A fault-tolerant abstraction for in-memory cluster computing. https://www.usenix.org/system/files/conference/nsdi12/nsdi12-final138.pdf.

236. Elhanati, Y., Sethna, Z., Callan, C. G., Jr, Mora, T. & Walczak, A. M. Predicting the spectrum of TCR repertoire sharing with a data-driven model of recombination. *Immunol. Rev.* **284**, 167–179 (2018).

237. Cabrales, L. E. B. *et al.* Modified Gompertz equation for electrotherapy murine tumor growth kinetics: predictions and new hypotheses. *BMC Cancer* **10**, 589 (2010).

238. Waliszewski, P. & Konarski, J. A Mystery of the Gompertz Function. in *Fractals in Biology and Medicine* 277–286 (unknown, 2005).

239. Yang, Y., Zha, K., Chen, Y.-C., Wang, H. & Katabi, D. Delving into Deep Imbalanced Regression. *arXiv [cs.LG]* (2021).

240. Shrock, E. L. *et al.* Germline-encoded amino acid-binding motifs drive immunodominant public antibody responses. *Science* **380**, eadc9498 (2023).

241. Willis, J. R., Briney, B. S., DeLuca, S. L., Crowe, J. E., Jr & Meiler, J. Human germline antibody gene segments encode polyspecific antibodies. *PLoS Comput. Biol.* **9**, e1003045 (2013).

242. Halatek, J., Brauns, F. & Frey, E. Self-organization principles of intracellular pattern formation. *Philos. Trans. R. Soc. Lond. B Biol. Sci.* **373**, (2018).

243. Taleb, N. N. & Douady, R. Mathematical Definition, Mapping, and Detection of (Anti)Fragility. *arXiv [q-fin.RM]* (2012).

244. Max, T., Cole, E., Obront, Z. & Hoffman, E. STATISTICAL CONSEQUENCES OF FAT TAILS.

245. Cottineau, C. West G., 2017, Scale. The universal laws of growth, innovation, sustainability, and the pace of life in organisms, cities, economies, and companies. *Cybergeo* (2017) doi:10.4000/cybergeo.28543.

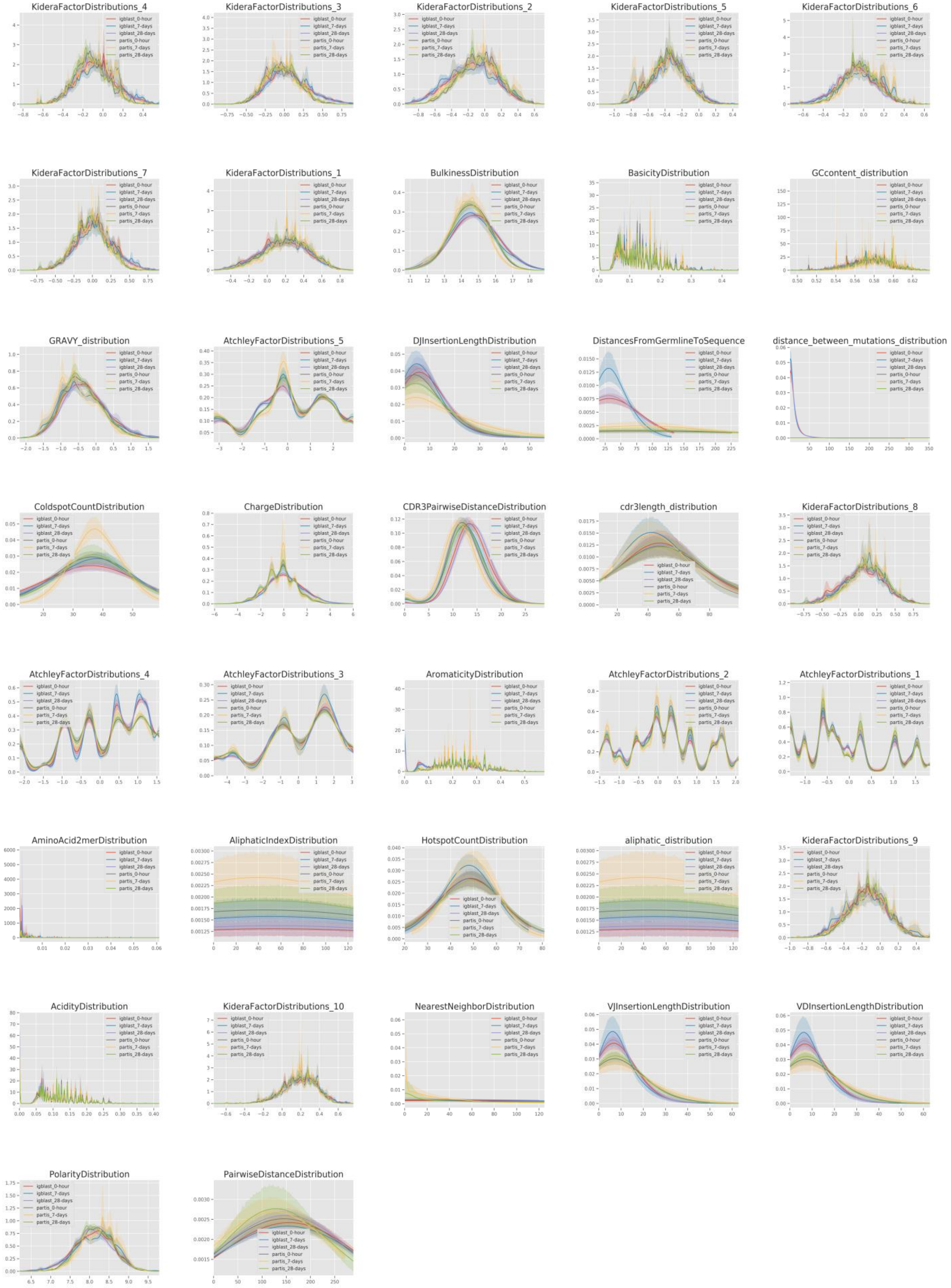
246. Der, R. & Martius, G. Novel plasticity rule can explain the development of sensorimotor intelligence. *Proceedings of the National Academy of Sciences* **112**, E6224–E6232 (2015).

247. Silver, D. *et al.* A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science* **362**, 1140–1144 (2018).

248. Vinyals, O. *et al.* Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature* **575**, 350–354 (2019).

249. Taleb, N. N. The black swan: The impact of the highly improbable. https://answerbook.ir/wp-content/uploads/2019/02/Black-Swan-Summary-Nassim-Taleb-Economist.pdf (2007).

Supplementary Figures


KideraFactorDistributions_4
KideraFactorDistributions_3
KideraFactorDistributions_2
KideraFactorDistributions_5
KideraFactorDistributions_6
KideraFactorDistributions_7
KideraFactorDistributions_1
BulkinessDistribution
BasicityDistribution
GCcontent_distribution
GRAVY_distribution
AtchleyFactorDistributions_5
DJInsertionLengthDistribution
DistancesFromGermlineToSequence
distance_between_mutations_distribution
ColdspotCountDistribution
ChargeDistribution
CDR3PairwiseDistanceDistribution
cdr3length_distribution
KideraFactorDistributions_8
AtchleyFactorDistributions_4
AtchleyFactorDistributions_3
AromaticityDistribution
AtchleyFactorDistributions_2
AtchleyFactorDistributions_1
AminoAcid2merDistribution
AliphaticIndexDistribution
HotspotCountDistribution
aliphatic_distribution
KideraFactorDistributions_9
AcidityDistribution
KideraFactorDistributions_10
NearestNeighborDistribution
VJInsertionLengthDistribution
VDInsertionLengthDistribution
PolarityDistribution
PairwiseDistanceDistribution
igblast_0-hour
igblast_7-days
igblast_28-days
partis_0-hour
partis_7-days
partis_28-days

Supplementary Figure 1. Distribution plot aggregates across the two frameworks at each single timepoint of *DDW et al* study.