2020

Statistical methods & algorithms for autonomous immunoglobulin repertoire analysis

BOSTON UNIVERSITY

GRADUATE SCHOOL OF ARTS AND SCIENCES

AND

COLLEGE OF ENGINEERING

Dissertation

STATISTICAL METHODS & ALGORITHMS FOR AUTONOMOUS IMMUNOGLOBULIN REPERTOIRE ANALYSIS

by

KATHERINE FRANCES NORWOOD

B.S., University of New Hampshire, 2014
M.S., Boston University, 2018

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2020

Approved by

First Reader _____

Thomas B. Kepler, Ph.D.
Professor of Microbiology
Professor of Mathematics & Statistics

Second Reader _____

Trevor Siggers, Ph.D.
Professor of Biology

Third Reader _____

Evan Johnson, Ph.D.
Professor of Medicine & Biostatistics

DEDICATION

"If I have seen further than others, then it is by standing on the shoulders of giants."

- Isaac Newton

This work is dedicated to all of my teachers, who opened my eyes to see at all and together raised me onto their giant shoulders.

ACKNOWLEDGMENTS

I would like to thank my mentor and primary adviser Dr. Tom Kepler, for teaching me how to instill the rigors of statistical thinking into daily practice, for his patience in answering my unending questions, and for reminding me when to concentrate on the bigger picture.

Thank you also to the rest of the Kepler Lab, both current and former members for their varied support over the past six years: Thank you to Dr. Akshaya Ramesh, Dr. Stephanie Pavlovich, and Dr. Sila Ataca for being such amazing role models, and your guidance in the day-to-day ups and downs of early years in grad school. Thank you to Dr. Kate Sawatzki, fellow buuny mom and DND gamer for her endless enthusiasm and excitement. Thank you to Fumiaki Aihara for his many welcome distractions from around nerd culture. Thank you to Dr. Axin Hua for being the lab's resident software and Visual Studio wizard, Dr. Grace Kepler for her humor and mathematical insights.

I'd also like to acknowledge the managers of the wet lab, whose tireless work has been the source of most of my data: Dr. Feng Feng, Dr. Yu Mei Wang. Thank you also to the lab's newest students Moises, Luciana, Yentl, Michelle, and Jack for their part in keeping me grounded.

I'd like to thank my thesis committee for their time and irreplaceable feedback over the entire progression of my project: Dr. Trevor Siggers, Dr. Luis Carvalho, Dr. Lee Wetzler, and Dr. Evan Johnson. Thanks also to Dr. Barbara Nikolajczyk for serving on my qualifying exam committee before her departure to the University of Kentucky.

I would like to especially thank my dissertation writing group: Beth Becker, Tyler Faits, Mike Quintin, Will Hackett, Chris Mancuso, and Dr. Daniel Lancour, for being peer mentors and accountability buddies, and for their continued mental and emotional support. Without you folks, the final three chapters of this dissertation may never have been finished.

I want to thank the rest of my friends from the BU, UNH, board gaming and role playing communities who have helped keep me sane over the past six years. Extra special thanks go to Beth Becker, Rachael Ivison, Tyler Faits, Dr. Mikayla Balch, Ricky DiCillo, Tim Schroeder, Patrick and Stephanie Green, Amanda and David Neider, and Jeremy Petravicz.

I want to thank my rabbit Phred, named for DNA sequencing quality scores, for being a continual source of joy, and for reminding me when to just slow down and eat my vegetables.

I want to thank my parents, Gordon and Lisa, my sister Jessica, my grandparents and the rest of my extended family, without whose unfailing love and unconditional support I could never have come this far.

Finally, I want to thank my fiancé Jon Wurtz, for keeping me grounded, and for always being there when I need you most.

STATISTICAL METHODS & ALGORITHMS FOR

AUTONOMOUS IMMUNOGLOBULIN REPERTOIRE ANALYSIS

KATHERINE FRANCES NORWOOD

Boston University Graduate School of Arts and Sciences

and College of Engineering, 2020

Major Professor: Thomas B. Kepler Ph.D., Professor of Microbiology and Professor of Mathematics and Statistics

ABSTRACT

Investigating the immunoglobulin repertoire is a means of understanding the adaptive immune response to infectious disease or vaccine challenge. The data examined are typically generated using high-throughput sequencing on samples of immunoglobulin variable-region genes present in blood or tissue collected from human or animal subjects. The analysis of these large, diverse collections provides a means of gaining insight into the specific molecular mechanisms involved in generating and maintaining a protective immune response. It involves the characterization of distinct clonal populations, specifically through the inference of founding alleles for germline gene segment recombination, as well as the lineage of accumulated mutations acquired during the development of each clone.

Germline gene segment inference is currently performed by aligning immunoglobulin sequencing reads against an external reference database and assigning each read to the entry that provides the best score according to the metric used. The problem with this approach is that allelic diversity is greater than can be usefully accommodated in a static database. The absence of the alleles used from the database often leads to the misclassification of single-nucleotide polymorphisms as somatic

mutations acquired during affinity maturation. This trend is especially evident with the rhesus macaque, but also affects the comparatively well-catalogued human databases, whose collections are biased towards samples from individuals of European descent.

Our project presents novel statistical methods for immunoglobulin repertoire analysis which allow for the de novo inference of germline gene segment libraries directly from next-generation sequencing data, without the need for external reference databases. These methods follow a Bayesian paradigm, which uses an information-theoretic modelling approach to iteratively improve upon internal candidate gene segment libraries. Both candidate libraries and trial analyses given those libraries are incorporated as components of the machine learning evaluation procedure, allowing for the simultaneous optimization of model accuracy and simplicity. Finally, the proposed methods are evaluated using synthetic data designed to mimic known mechanisms for repertoire generation, with pre-designated parameters. We also apply these methods to known biological sources with unknown repertoire generation parameters, and conclude with a discussion on how this method can be used to identify potential novel alleles.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ILLUSTRATIONS

LIST OF ABBREVIATIONS

AVA ..... Anthrax Vaccine Adsorbed

Avg. ..... Average

BCR ..... B Cell Receptor

e.g. ..... exempli gratia

HIV ..... Human Immunodeficiency Virus

i.e. ..... id est

IG ..... Immunoglobulin

IGHV ..... Immunoglobulin Heavy Chain V Gene

IGVRG ..... Immunoglobulin Variable Region Gene

IMGT ..... IMmunoGeneTicsr

mRNA ..... Messenger RiboNucleic Acid

PMF ..... Probability Mass Function

SNP ..... Single Nucleotide Polymorphism

UCA ..... Unmutated Common Ancestor

V(D)J ..... Variable, Diversity, Joining

VRG ..... Variable Region Gene

CHAPTER ONE: INTRODUCTION

Role of Statistical Modeling in Modern Vaccine Development[1–3]

Vaccine technology has saved countless lives by harnessing the hallmark feature of an adaptive immune system: ‘immunological memory’. This evolutionary marvel is what allows an organism to recognize repeated encounters with pathogens, and to launch a stronger, more coordinated immune response. Over the past two decades, vaccine research and development has seen significant advances in both genetic sequencing technology and greater access to computational resources.

One active area within vaccine development is immunoglobulin (IG) repertoire analysis, a field dedicated to analyzing the genetics of the immune cells responsible for producing immunoglobulin proteins in order to better understand the fundamental mechanisms of the adaptive immune system in response to natural infectious disease (or its simulation by vaccine). As the methodology of the field continues to shift towards more quantitative approaches, there is a critical need for novel statistical methods and sophisticated algorithms which can overcome the inherent challenges associated with analyzing immunoglobulin repertoire data robustly and accurately.

In this introductory chapter, we review the foundational biology of adaptive immune system as it pertains to the development of the immunoglobulin repertoire, and the challenges this poses to quantitative data analysis. In particular, we concentrate on the mechanisms behind the production of immunoglobulin proteins, with an emphasis on the different sources of their molecular diversity. We conclude with a brief discussion on

existing algorithms available for quantitative immunoglobulin repertoire analysis, along a review of their existing limitations and avenues for further development.

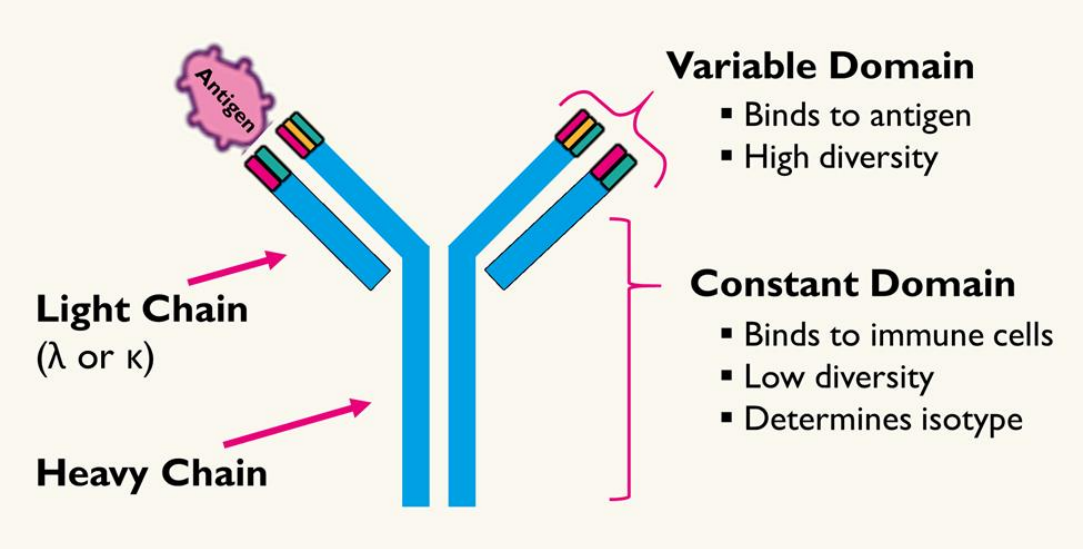
Structure & Function of Antibody Immunoglobulin Protein[1,4]

The adaptive immune cells responsible for secreting immunoglobulin proteins are a subclass of white blood cells known as B cells. In early stages of B cell development, immunoglobulin proteins are found on the surface membrane of B cells, where they are often referred to as B cell receptors (BCRs). In fully-differentiated B cells, or plasmablasts, these immunoglobulin proteins are secreted *en masse* into the interstitial fluid as antibodies. The primary function of antibodies is to bind to foreign material called antigens, which are components of an invading pathogen or its toxic byproducts, and come from a variety of molecular sources (lipids, polysaccharide, glycoprotein etc.). Antibody binding to antigens is what allows for the direct neutralization of pathogens and their associated toxins. Antibody binding also facilitates the identification and destruction of pathogens by other circulating immune cells, like macrophages, as well as the activation of the complement branch of the innate immune system.

The structure of the immunoglobulin protein is well-suited to carry out these intended functions. Each secreted antibody molecule has separate domains for binding host immune cells and foreign antigens. Figure 1 contains an illustration of these distinct binding domains, in relation to the overall immunoglobulin structure. Each antibody is composed of two heavy chains and two light chains, each of which contain separate ‘variable’ and ‘constant’ region binding domains. Despite its name, the ‘constant’ region domain does exhibit some molecular diversity. Variation in heavy chain constant region

allows for changes in quaternary protein conformation, called isotypes, which influences function by allowing for selective detection by different immune cells. For example, antibodies with isotype IgM exist as pentamers, and expressed primarily by early-stage B cells (naïve B cells), whereas IgG isotype antibodies are secreted as monomers by later-stage B cells.

Figure 1: Illustration of Antibody Structure



In contrast, the variable region domain is responsible for binding to the antigen, and exhibits an extremely high level of molecular diversity. This is due to the significant molecular challenge associated with providing the unique specificity required for high binding affinity in the face of a potentially limitless space of antigenic binding surfaces. The challenge is further compounded at the genetic level, where there is an additional constraint on the proportion of genome space that can be allocated for encoding immunoglobulin proteins. However, evolution has selected for several solutions that maintain efficient storage of genetic information while also incorporating high degrees of

molecular diversity. We review two of these mechanisms, V(D)J recombination and clonal evolution, in the following sections.

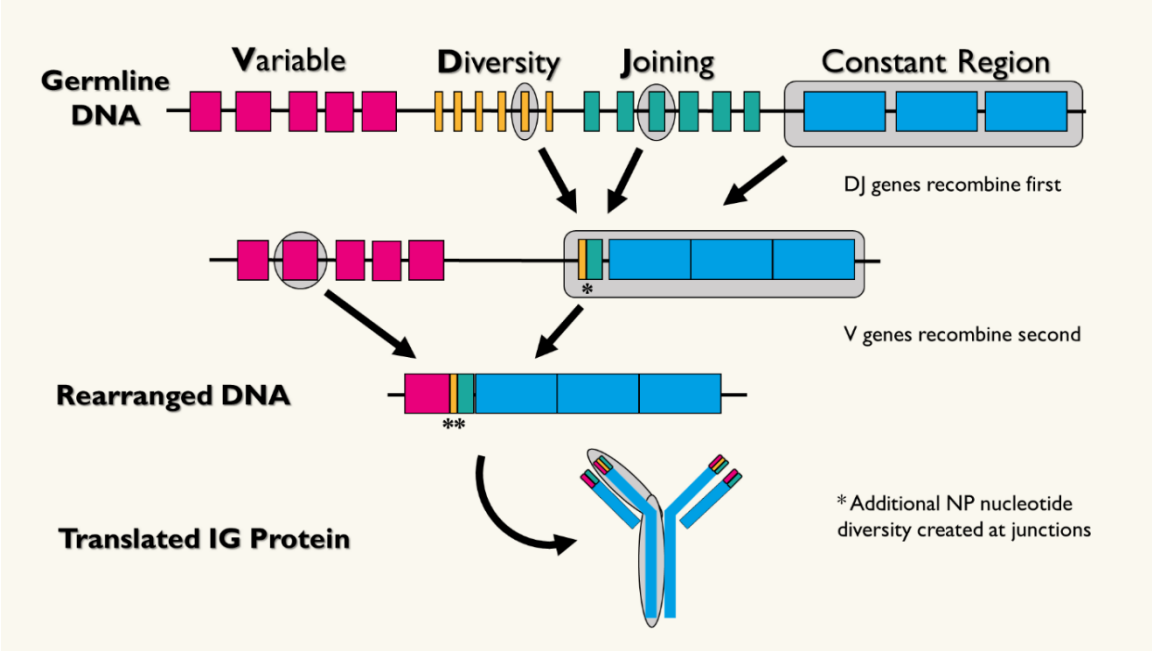
V(D)J Recombination[1,5]

V(D)J recombination is a process of stochastic rearrangement of germline immunoglobulin gene segments which occurs during the early stages of B cell development. For humans, the genetic loci which encode for these gene segments are located on different chromosomes, with heavy chain gene segments being on chromosome 14 and kappa (κ) and lambda (λ) light chain gene segments being located on chromosomes 2 and 22 respectively. Across all loci, these gene segments exhibit substantial diversity both in terms of their length and their overall information complexity. For heavy chains, the locus is divided into three categories of gene segment, labeled (V)ariable, (D)iversity, and (J)oining gene segments respectively, whereas both light chain loci contain only V and J gene segments.

V(D)J recombination entails the rearrangement of these diverse gene segments such that one segment from each of the classes joins together to form a unique gene combination. Heavy chain recombination occurs prior to light chain recombination, with the DJ join occurring first, followed by the V-DJ join. The mechanism by which this occurs also incorporates additional molecular diversity at the junctional sites of rearrangement in the form of NP nucleotide addition and deletion. Because of this stochastic rearrangement process, there is a high likelihood of introducing frameshift mutations that compromise the folding integrity of the translated immunoglobulin protein. As a result, light chain recombination only occurs following a productive heavy

chain rearrangement, and proceeds using the κ chain gene segments by default. λ light chain gene segments are only rearranged and incorporated into the immunoglobulin when κ chain rearrangements fail to produce a productive light chain from both parental chromosomes.

Figure 2: Simplified Illustration of Heavy Chain V(D)J Recombination



The combinatorial possibilities from V(D)J gene segment rearrangements and heavy/light chain pairings account for a significant fraction of the molecular diversity required to challenge an effectively limitless space of potential antigenic binding surfaces. Beyond the broad VDJ classes, germline-level variation can further hierarchically subdivided into separate gene families, segments, and alleles. Germline gene name notation reflects this hierarchical organization; for example, the gene name IGHV3-23*01 indicates the first allelic variant of the twenty-third gene segment in the V3 family of heavy chain variable region immunoglobulin genes. Note that while any

given individual will only contain at most two allelic variants per germline gene segment (one on each parental chromosome), the population for a species as a whole will exhibit greater allelic variation per gene segment. Table 1 below summarizes the number of known functional human variable region gene segments, as catalogued by the international ImMunoGeneneTics (IMGT) information system in February 2019, along with an approximate average length for each category of gene segment. As discussed in the section as the conclusion of this chapter, these genes likely do not represent the full breadth of human allelic diversity, but can serve as a rough guideline for understanding.

Table 1: Summary of V(D)J Alleles for Human Heavy & Light Chains

CHAIN	CLASS	CODE	APPROX. LENGTH	# UNIQUE SEGMENTS	# ALLELES[1]
HEAVY	Variable	IGHV	~300 nt	55	267
	Diversity	IGHD	~15 nt	22	30
	Joining	IGHJ	~50 nt	6	13
KAPPA (κ)	Variable	IGKV	~290 nt	41	66
	Joining	IGKJ	~40 nt	5	9
LAMBDA (λ)	Variable	IGLV	~300 nt	33	70
	Joining	IGKJ	~40 nt	5	7

Clonal Evolution and Immunoglobulin Affinity Maturation[1,6]

In later stages of B cell development, B cells migrate to germinal centers within secondary lymphatic tissue (e.g. lymph nodes, spleen) where they enter a microcosm of

[1]Pulled from IMGT's database for IG variable region genes, human, functional (pseudogenes and ORFs excluded; gene segments with multiple functionality codes were included as long as they contained at least one functional allele). Date of accession: 2-26-2019

evolution by natural selection inside germinal centers of secondary lymphatic tissues. Each progenitor B cell, having selected its own unique V(D)J gene segment rearrangement, will display a unique immunoglobulin protein on its surface membrane as a B cell receptor (BCR). BCRs which are capable of binding with material provided by antigen-presenting cells and are activated by other immune cells within the germinal center are stimulated to proliferate. Thus, every descendent B cell within a shared lineage of its founding progenitor cell is a member of a B cell clone. During these successive rounds of proliferation, the rearranged genetic loci responsible for encoding for the BCRs will be subjected to a course of intentional somatic hypermutation. The rate of polymorphisms introduced at these loci is significantly higher than the natural background rate, which confers extra molecular diversity. Many of these acquired mutations will have a deleterious effect on BCR/antigen binding, resulting in the eventual extinction of the clone through negative selection. However, some of the polymorphisms will provide a net positive selective advantage on binding, allowing for the expansion of the clonal population and a dramatic increase in immunoglobulin binding affinity.

Figure 3: Simplified Illustration of Clonal Selection & Expansion

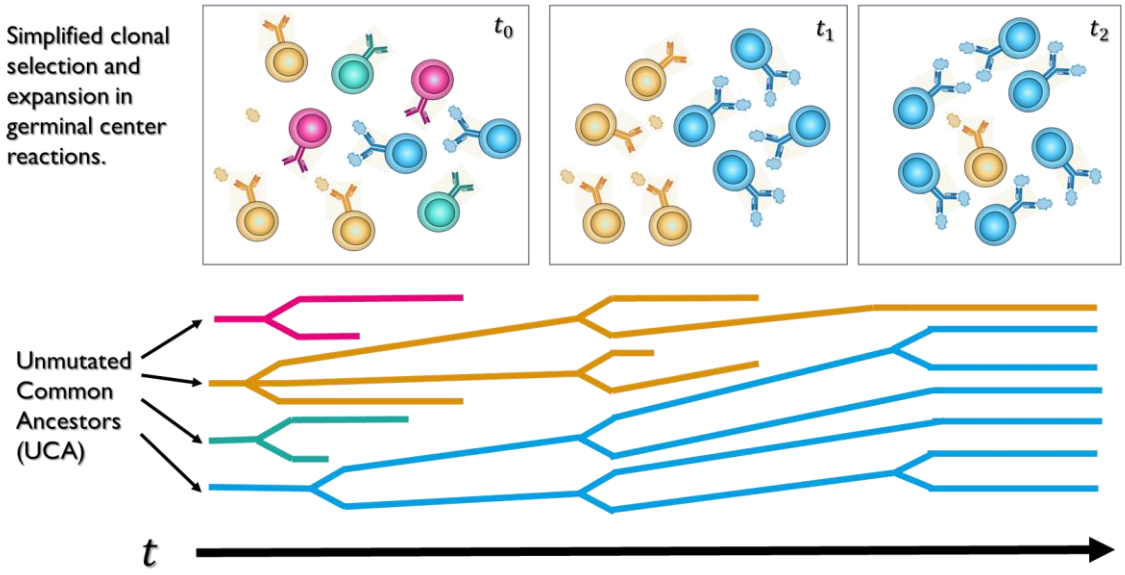


Figure 3: t = time after clonal founder, UCA = unmutated common ancestor, clonal founder or progenitor

Current Approaches to Immunoglobulin Repertoire Analysis[2,3,7–12]

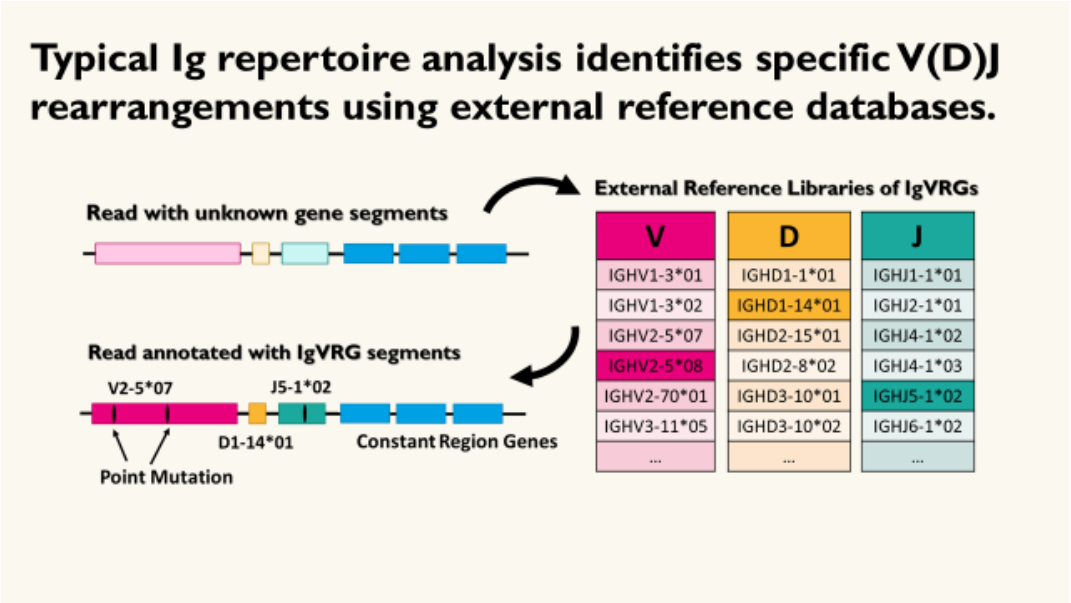
The unique features of the adaptive immune system which enable high levels of diversity in the immunoglobulin repertoire are also the ones which make it an interesting challenge for statistical modelling. This kind of information is often sought after by groups pursuing 'rational vaccine design': a modern approach to vaccine development for viruses which have proven difficult to develop effective vaccines for, like HIV and influenza. By collecting samples of the adaptive immune system during an active response to a pathogen (whether through a natural infection, or one simulated by vaccination), these groups can shine a light onto the specific features which confer immune protection. In particular, computational biologists who aim to characterize immunoglobulin repertoires are typically interested in identifying the specific gene

segment rearrangements and acquired mutations of particular antibodies, and how these events influenced antibody binding affinity to a particular antigen.

Genetic material from these immune repertoire samples can be isolated through several next-generation sequencing platforms, including bulk transcriptomics, single-cell sequencing, and immunoglobulin-specific sequencing. Sequencing pipelines can be customized to suit the needs of an individual study, but typically include filtering and normalization steps to ensure read quality, annotation of reads with V(D)J rearrangements, partitioning into distinct clonal lineages, and inferences on the mutations acquired during clonal evolution.

Immunoglobulin gene segment annotations are typically assigned to reads following their alignment against a reference database, or ‘library’, of known allelic variants, like the ones maintained by IMGT. Positions within sequencing reads that differ from those sequences found in the reference databases are usually marked as acquired polymorphisms, after controlling for the inherent sources of technical error with sample preparation and sequencing. Together, both gene segment assignment and mutational patterns inform the statistical methods for clonal lineage inference and classification.

Figure 4: VRG Gene Annotation with Reference Library



To date, only two non-alignment based methods for immunoglobulin repertoire analysis have been recently published: IgGraph and IgDiscover. However, both of these methods still retain a logical dependency on an external reference database of immunoglobulin allelic variants. IgGraph is an innovative de Bruijn graph-based algorithm, which incorporates IMGT reference segments into their antibody graphs as 'colored' reads. IgDiscover is a clustering method designed to detect novel alleles, and also requires an initial input starting database of reference alleles, which is updated iteratively over the course of the algorithm's execution.

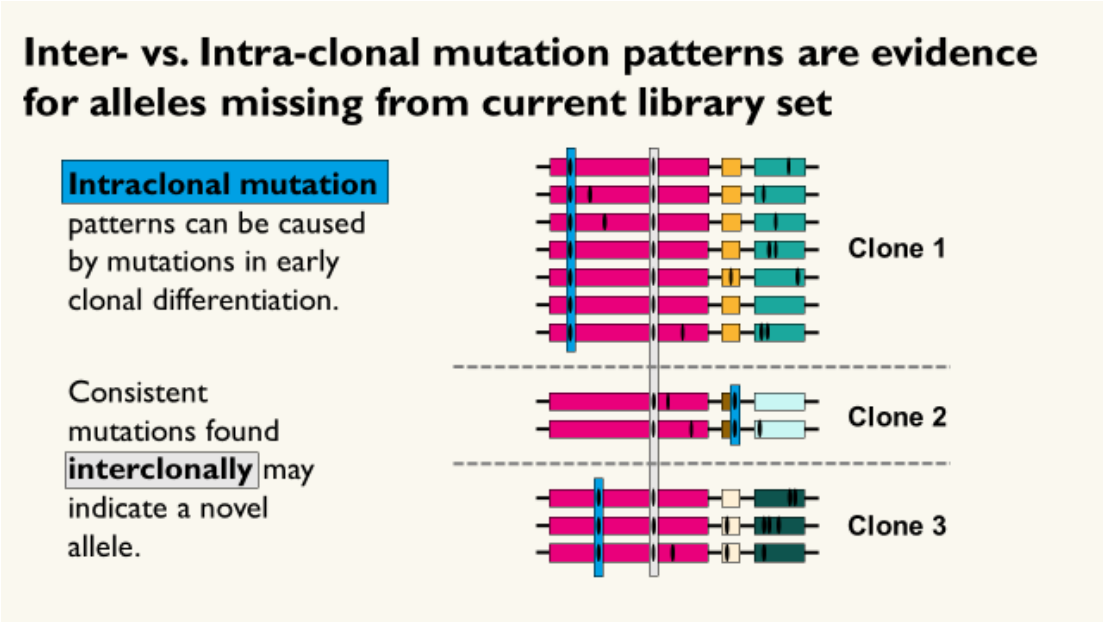
Limitations to Current Methods for Immunoglobulin Repertoire Analysis[13–17]

The primary issue facing all current methods and algorithms for immunoglobulin repertoire analysis is their dependency upon a complete and accurate reference database of germline gene segment alleles. However, most organisms have no available reference

databases, which limits their potential as vaccine development models or for comparative immunology studies. Furthermore, those organisms which do have available reference databases are systematically undersampled in regards to the overall allelic diversity present within the species as a whole. This is especially true for the rhesus macaque, a nonhuman primate frequently used as a model organism for early vaccine trials. However, even the comparatively well-catalogued human reference databases are incomplete, as evidenced by the discovery of multiple novel alleles within systematically underrepresented populations. Using incomplete reference databases in repertoire analysis poses significant problems to the overall accuracy of downstream results. The schematic in Figure 5 demonstrates how the validity of the interpretation of observed genetic variation can be called into question due to an incomplete reference database.

Interclonal vs. Intraclonal Mutation Patterns

Figure 5: Interclonal vs. intraclonal mutation patterns



The schematic above contrasts two types of commonly observed mutation patterns: interclonal and intraclonal patterns. An intraclonal mutation pattern is unique to the members of a given clone, whereas an interclonal mutation pattern can be observed across members of multiple clones that share common V(D)J rearrangements. In the illustrated example, the clones share a common Variable gene assignment, but have different Diversity and Joining gene assignments. Intraclonal mutation patterns are more likely to indicate shared mutations which were acquired during somatic hypermutation and clonal evolution, while interclonal mutation patterns are more likely to be indicative of a reference database with missing alleles. This is due to the extremely low probability of observing the same mutation at the same nucleotide position across multiple clones.

Outline of Proposed Methods for Autonomous Repertoire Analysis

My research project is on the development of novel statistical methods and algorithms for autonomous immunoglobulin repertoire analysis. In this context, ‘autonomous’ refers to the inference of germline gene segment assignments directly from high-throughput immunoglobulin sequencing data, without reliance on external databases of reference libraries. This project can be divided into four specific aims:

1. *De novo* construction of candidate germline gene segment libraries for internal modelling, given only the information available from processed immunoglobulin sequencing reads.
2. Iterative improvement of candidate libraries using information obtained from standard library-based analysis methods.

3. Evaluation of algorithm performance through synthetically generated repertoire data, designed to mimic diversity of true immunoglobulin repertoires.
4. Application of statistical methods to actual biological data collected from human subjects as part of an earlier immunoglobulin repertoire study.

The first two aims are achieved with a machine learning procedure outlined below, in four separate phases. The first aim is accomplished via the initialization procedure of Phase 1, which uses a clustering algorithm based on the Dirichlet process to group processed reads based on the likelihood that they share a common ancestral VRG allele. When cluster membership is finalized, each inferred ancestral allele is submitted as a tentative entry into the germline gene segment libraries. The details of this procedure are discussed in the following chapter.

Phase 2 represents a traditional repertoire analysis pipeline which relies on the use of reference libraries for analysis. However, instead of using a potentially incomplete external reference database, these methods incorporate the internal germline gene segment libraries constructed in Phase 1. The second major project aim is accomplished during Phase 3 of overall machine learning procedure, using information from both the constructed internal libraries, as well as the results of a standard Ig repertoire analysis pipeline in Phase 2. Since Phases 2 and 3 are connected in an iterative loop, Phase 4 represents the criteria for termination of the loop, as well as the general conditions for evaluation of the learning process as a whole. These methods are discussed in Chapter 3.

The results of the third aim are discussed in Chapter 4. It details the series of experiments used in empirical selection of model hyperparameters, as well as a variety of

trials on synthetic data of the entire learning procedure. Overall algorithm performance and potential areas for further improvement are also discussed in this chapter.

The fourth aim is addressed in Chapter 5, which discusses the results of applying our algorithm to human immunoglobulin heavy chain repertoires. It also compares the results of our analysis with those of three reference database-dependent approaches: Cloanalyst, IgBlast, and IMGT's High-VQUEST. We explore the possibility of potential discovery of a novel allele, review the strengths and limitations of our approach, and conclude with a discussion on remaining open questions and potential future directions.

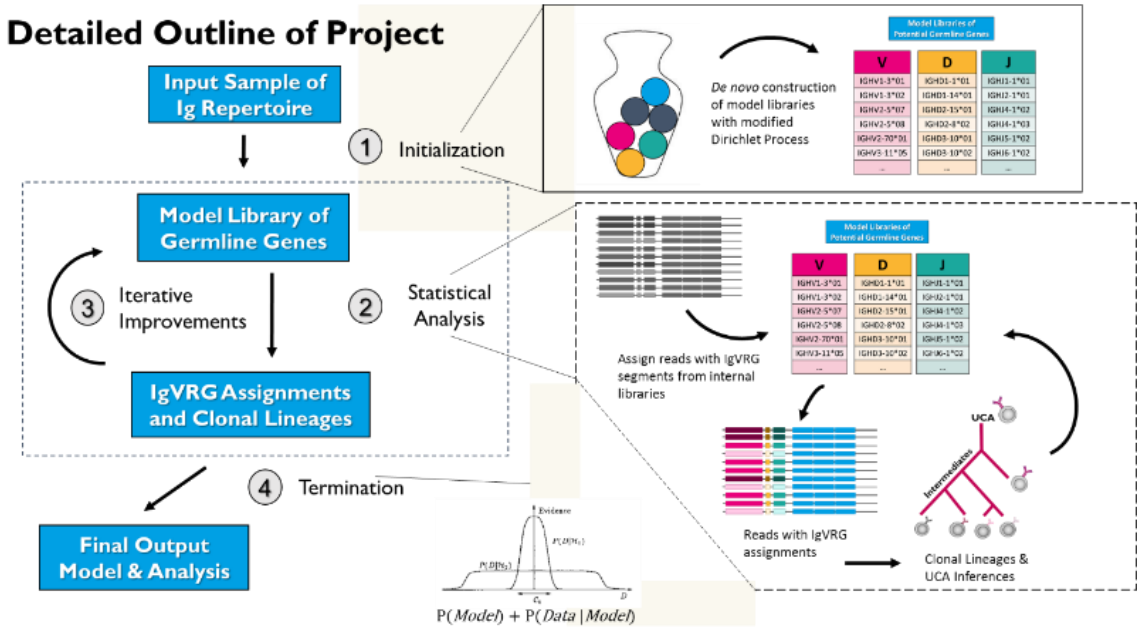
CHAPTER TWO: *DE NOVO* INFERENCE OF GERMLINE GENE LIBRARIES FROM IMMUNOGLOBULIN REPERTOIRE DATA

Introduction

Analyzing the immunoglobulin repertoire with data collected from high-throughput sequencing comes with its own set of unique challenges. The principle challenge facing existing methods for immunoglobulin repertoire analysis is in identifying the biological and technical sources of observed read variation with a high degree of accuracy and precision. While much work has been done to address observed variation caused by the numerous technical challenges of read quality control, there still exists a need to develop statistical methods which can robustly differentiate between the opposed biological sources of germline allelic variation and acquired somatic mutation.

In this chapter, we present a machine learning model which aims to disentangle these two sources of variation by autonomously inferring libraries of germline alleles de novo, using only information available within the high-throughput sequencing data itself, and iteratively improving upon those libraries using insights collected from reference-based repertoire analysis methods. Figure 6 outlines this overall model, with each of the four main components of the schematic discussed in their corresponding sections below.

Figure 6: Project Outline Schematic[18]

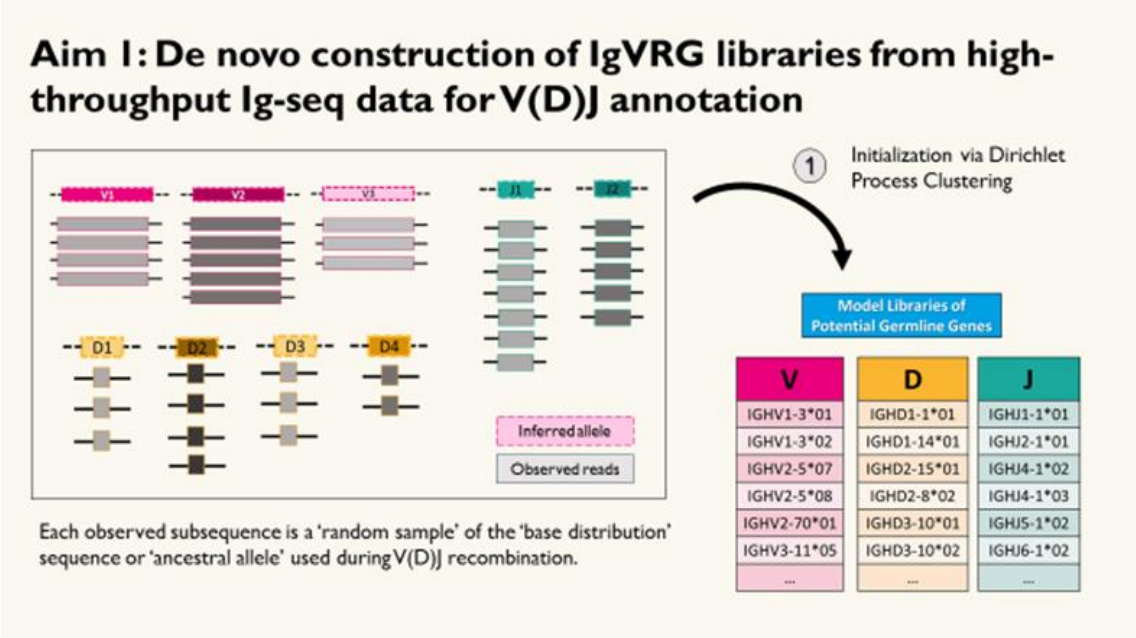


Initialization of autonomous machine learning procedure

Alignment-based methods for immunoglobulin repertoire analysis all rely upon an external reference database of germline genes as an integral part of their approach. This becomes problematic in cases when these reference databases are either unavailable, as is the case for many organisms of potential research interest or are incomplete due to undersampling a species' allelic variation. Our methods overcome this essential limitation by initially inferring a set of internal reference libraries of germline gene alleles directly from high throughput sequencing data. This *de novo* inference of allele libraries is done through a clustering procedure based on the Dirichlet process, and is discussed in detail in the following chapter. In essence, sequences are grouped according to a likelihood function which accounts for their overall shared similarity. The features which are shared across a cluster as a whole are used to infer the most likely candidate

for a potential germline allele, while features which only exist in a subset of cluster members are attributed to individual variation arising from acquired somatic mutation.

Figure 7: Summary of Project Aim 1

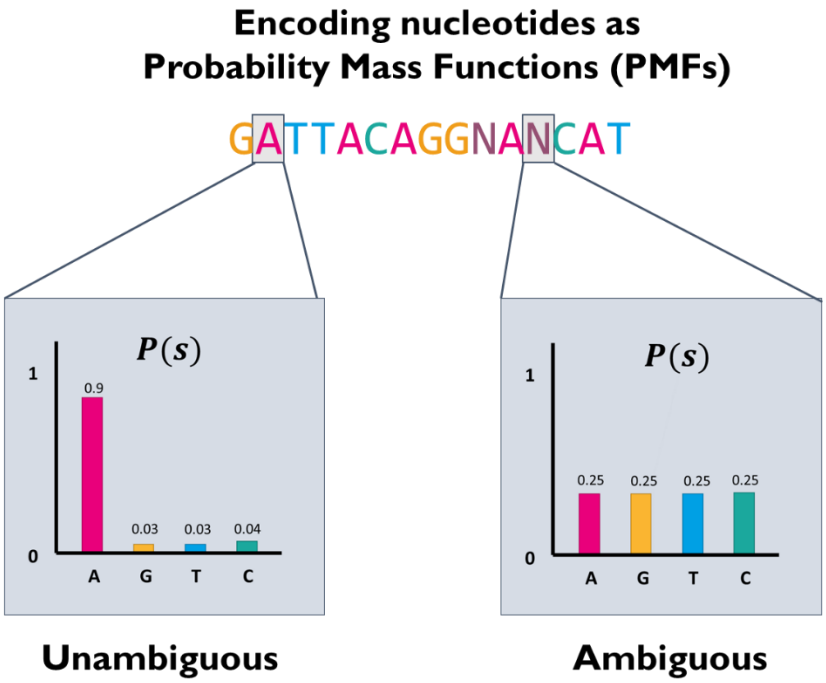


There are several features of the Dirichlet process clustering approach that we find particularly suitable for our purposes in this project. First, the Dirichlet process clustering approach is ideally suited for cases when the true number of clusters is unknown, as it is here with an unknown number of true germline alleles. This gives it an advantage over less sophisticated algorithms such as K-means, where user-designated parameters fix the total number of clusters prior to analysis. Second, as the number of new observations increases, the expected number of total clusters converges to some finite number, but the probability of detecting a new cluster always remains nonzero. Third, this relative probability of detecting novel clusters is highest in early stages of the

Dirichlet process, when there have been relatively few observations, but decays as the total observations accumulate.

We argue that these properties are well suited to our machine learning model because they mimic the natural process associated with scientific discovery. Namely, starting from a position of relative ignorance as to the true state of total number of clusters, or analogously germline gene segment alleles, the relative probability of detecting a novel cluster or previously unobserved allele is high. Greater amounts of evidence and experience allow us to update and refine our existing models. The process always maintains a capacity to overturn the existing model, but any means to do so must pass a higher burden of proof with each new successive observation. Similarly, we wish to always have the capacity to detect novel germline gene alleles, as long as that detection is mitigated by the overall accumulated evidence surrounding known gene segments. We further argue that this approach, which mimics the natural discovery process, is more statistically sound than methods which use metrics for measuring and scoring mismatches, insertions and deletions. This is because our methods fundamentally encode several of the uncertainties of the system in question into the evaluation framework itself. For example, the likelihood of observing a particular nucleotide variation at a given position within a sequence is explicitly modeled as a probability mass function, which allows us to directly quantify our uncertainty.

Figure 8: Sequence Reads Modeled as Probability Mass Functions



Statistical analysis of immunoglobulin sequencing reads using standard methods[19,20]

The purpose of the initialization phase of the project is to derive a model of internal libraries of candidate germline gene segment alleles with a Dirichlet process clustering procedure. We discuss the mathematical underpinnings of this procedure in the following chapter. In contrast, the purpose of the second phase of this project is to analyze the original immunoglobulin sequencing data with previously developed alignment-based methods, but replacing the external reference database of alleles with the internal model libraries constructed from the initialization phase.

In this project, the traditional statistical analysis of immunoglobulin sequencing reads takes on three unique forms. First, sequencing reads are annotated with unique

V(D)J combinations using V segment ‘alleles’ derived from our inferred libraries, and DJ segment alleles derived from external reference libraries. Reads are assigned with the allelic candidates using a maximum likelihood based scoring function, with any observed variation from proposed germline categorized as acquired somatic mutation, which also allows each read sequence to be annotated with an associated mutation frequency. These sets of unique V(D)J combinations allow the reads to be partitioned into distinct clones.

Second, a maximum likelihood tree is inferred for each clone, under nucleotide substitution evolutionary models. The evolutionary models we use in our tree inference work are those derived from Kimura80 and Jukes & Cantor ’69. An example of one these inferred maximum likelihood trees is given in Figure 9 below. The principal difference between the two models is that Kimura gives separate rate parameters for nucleotide transitions and transversions, whereas Jukes-Cantor allows only a single rate parameter for non-self nucleotide substitution. For our work, we predominantly favor the Jukes-Cantor model over the Kimura model for alignment with an exception for the alignments used for identifying the conserved cysteine codon used in parsing V gene segments from the rest of the immunoglobulin read.

Figure 9: Sample Clonal Maximum Likelihood Tree and Member Sequences

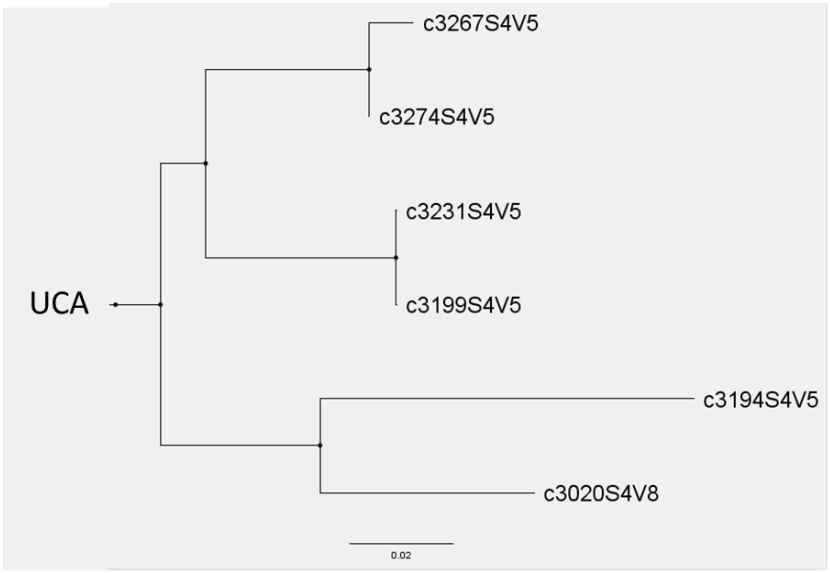


Figure 9: Dots indicate an identical nucleotide as reference at index position; individual letters refer to nucleotides which differ from the reference sequence.

Third, for each clonal lineage tree, we can derive a likelihood function which is representative of the entire clone; this function would take the individual nucleotide sequences which make up members of a clone as input, and then output a single real number value which expresses the information cost associated with grouping these sequences under a shared clone. Here, our encoding of nucleotides as probability mass functions becomes particularly important in determining how much weight to assign ambiguous positions within a given member sequence. This method also allows us to account for the extreme variation we often see in clone size as it is not unusual for a small number of clones to take up a large proportion of the total sequence population. Using a single likelihood function representative of the clone as a whole allows for us to control for this 'jackpot effect' while also simultaneously comparing clones independently of one another.

We wish to reiterate that the methods in this second phase are part of the standard repertoire analysis pipeline, and that the innovation introduced from our approach stems primarily from (a) the direct inference of the allelic libraries used in read annotation, rather than an external source and (b) the integration of both our novel allelic inference and standard repertoire methods into a cohesive, iterative learning procedure.

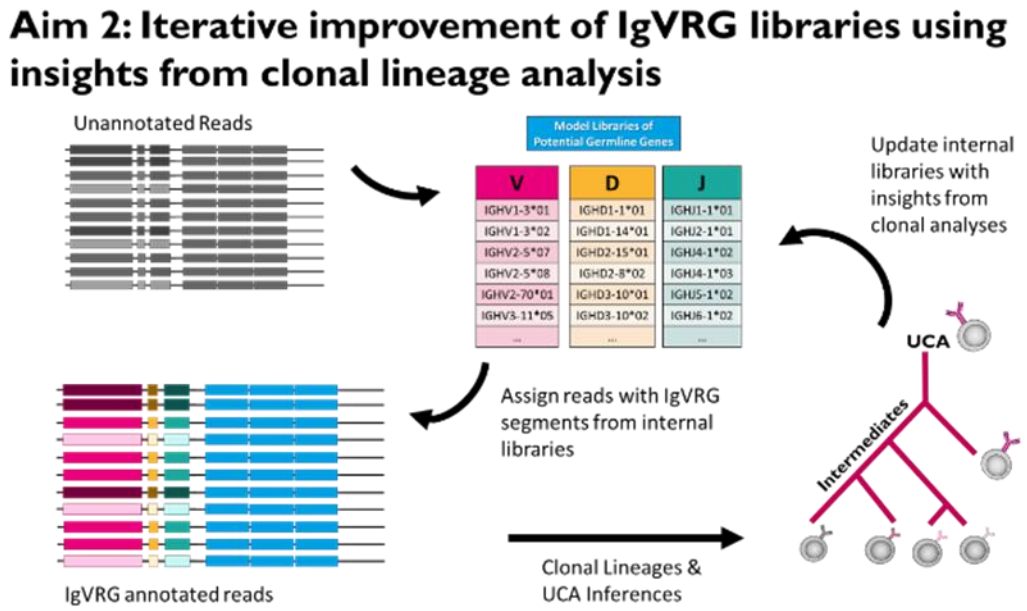
Iterative improvement of internal libraries for candidate germline alleles

The third phase of the project satisfies the second aim of the project, wherein we improve our initial internal model libraries for germline gene segments alleles. We achieve this using by using the information contained within the clonal likelihood functions derived from the second phase of the project. By interpreting the variation

within the clonal likelihood functions as if they were a single representative sequence, we can better examine the interclonal variation indicative of a shared ancestor allele. The distinction between phase two and three is as follows: In phase two, we use the intraclonal variation to inform our understanding of clonal history of somatic mutations acquired during immunoglobulin affinity maturation. In phase three, we use the interclonal variation to inform and improve our understanding of the underlying allelic variation amongst clones which share common gene segments.

This is done by performing additional rounds of the same Dirichlet process clustering procedure used in phase one. However, in this phase of the project, instead of clustering on the immunoglobulin sequencing reads themselves, the procedure clusters on the representative clonal likelihood functions. In just the same manner as the previous two phases, we produce candidate libraries of germline alleles, which are then used to derive new clonal lineage inferences. We repeat these two phases in an iterative fashion, as diagrammed in Figure 10 below.

Figure 10: Summary of Project Aim 2



Phase 4: Loop Termination & Library Evaluation

Since we have modeled the clonal lineages as distinct likelihood functions, we have a consistent and convenient framework for measuring both the accuracy of our model and its overall complexity. This is because the valuation of the likelihood function also serves as a measure of the cost associated with storing information about competing clonal models.

In general, we have found that this global likelihood scoring function converges rapidly after only a few iterations of the phase two / phase three loop, with the system seeing the largest degree of improvement within the first three iterations and typically reaching a steady state around 5-7 iterations. By default, we set a fixed termination criteria at 7 iterations to maintain protocol consistency for our synthetic and biological trials discussed in chapters 4 & 5. Future upgrades to the software could make this

criteria setting adjustable under an ‘advanced settings’ interface, but is currently restricted since any additional iterations would require a significant increase to overall runtime for marginal model improvements.

CHAPTER THREE: ADAPTATION OF THE DIRICHLET PROCESS FOR CLUSTERING IMMUNOGLOBULIN SEQUENCES

The primary aim of this chapter is to explain in detail the Bayesian statistical methods and algorithms used for clustering our immunoglobulin sequencing reads, in order to arrive at a putative set of internal libraries of germline gene segment alleles. The Dirichlet process plays an integral role in both library construction and improvement, and is defined both formally in the context of infinite mixture models as a nonparametric Bayesian approach to clustering, and demonstrated informally through a Pólya urn illustration. We discuss how both Bayesian components of likelihood function and prior are calculated and applied to our clustering context, as well as describe how each of these calculations fits into the Gibbs sampler machine learning algorithm.

Role of the Dirichlet Process in Larger Machine Learning Model[21–31]

In the first iteration of our larger machine learning algorithm, we construct libraries of germline gene segment alleles de novo from immunoglobulin sequencing reads. In later iterations, we construct these libraries using representative sequences from maximum likelihood trees derived during the previous iteration's clonal lineage analysis. We arrive at these libraries using a Gibbs sampling clustering procedure, based on the Dirichlet Process infinite mixture model. It is important to note that this clustering procedure does not occur on intact immunoglobulin gene sequences, but rather on parsed subsequences which have been separated into their V and DJ components. The details of this parsing procedure (and other sequence pre-processing steps) are included in Chapter 4. The methods described in this chapter were developed solely for the clustering and

inference on V gene segment alleles, but in principle could be also be applied to D and J gene segment alleles. However, since D and J gene segments contain significantly less information than V gene segments due to reduced overall length, practical implementation of appropriate clustering methods has been reserved for future algorithm upgrades.

The Dirichlet Process infinite mixture model is a type of Bayesian nonparametric method. Mixture model methods aim to characterize subpopulations (or clusters) within a larger population, when the membership of individual observations into the potential subpopulations is unknown. The classical example is the Gaussian mixture model, whose full population distribution can be described as the sum of K Gaussian distributions $f(x|\theta_i)$, proportional to their corresponding weight parameters p_i.

Eqns. 1.1 – 1.2

$$f\left(x|\vec{p},\vec{\theta}\right) = \sum_{i=1}^{K} p_i f(x|\theta_i)$$
$$\theta_i = \{\mu_i, \sigma_i\}$$

θ_i represents the parameter set for the individual distributions of the mixture model, which are the mean (μ_i) and variance (σ_i) parameters of the corresponding Gaussian distributions. p_i is the coefficient which determines the relative weight of each individual distribution to the overall mixture, such that $\sum_i p_i = 1$ and $p_i > 0, \forall\, i$. Finite mixture models, like the one in Eqn. 1.1, are unsuitable for our application due to the need for the statistician to predesignate the total number of clusters K prior to analysis. Nonparametric methods, like the Dirichlet process, allow for generalization to a potentially infinite number of distributions that make up the mixture model.

They allow us to simultaneously estimate the posterior distributions on the clustering structure of our observed data under marginalized likelihood functions, while also estimating the prior probabilities on the parameters of those density functions. The nature of this simultaneous prediction is what makes it especially useful for clustering applications when the true number of subpopulations is unknown.

There are two primary components to the Dirichlet Process implementation of a Bayesian infinite mixture model: a 'base distribution' H, and a scaling or concentration parameter α. H represents the family of distributions that each of the defining mixture subpopulation distributions inherits from, while α represents how much to weigh the proportions with which each of the subpopulations are mixed. In our previous example, H would represent the Gaussian family of distributions, where each of the individual mixture components $f(x|\mu_i, \sigma_i)$, are random sample distributions from H. Similarly, α would be associated with the proportional weight parameters p_i. In this sense, samples from the Dirichlet Process can be thought of as modeling a 'distribution of distributions'.

Eqns. 2.1 – 2.4

$$f(x|\theta_i) \sim H$$
$$\theta_i \sim G = Gaussian(\mu_i, \sigma_i)$$
$$G \sim DP(\alpha, H)$$
$$f\left(x|\vec{p}, \vec{\theta}\right) = \int_1^{\infty} p_i f(x|\theta_i)$$

Illustration of Dirichlet Process through Pólya Urns

We can illustrate how to apply the Dirichlet Process to a classification problem using a variation of the classical Pólya urn model. (Note that this variation is not

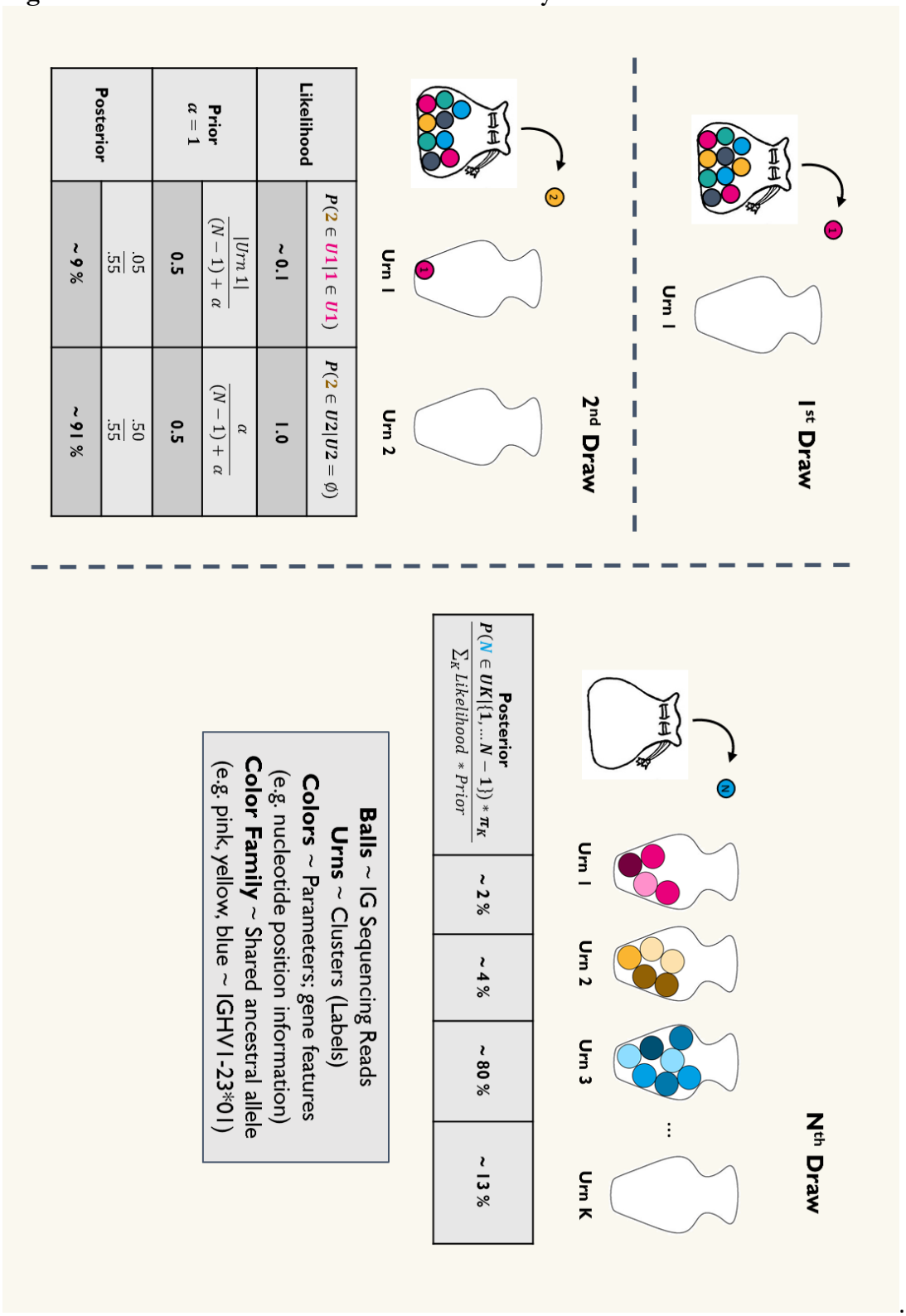
equivalent to the Blackwell-McQueen urn scheme, which describes a generative model for the Dirichlet Process, instead of a classification model.)

Imagine we have a bag filled with a large number of differently colored balls, and we wish to sort these balls into a series of urns such that all of the balls in a given urn are shades of the same color family. Given the continuous nature of the color spectrum, there are a potentially infinite number of 'base colors' that could be used to label each of our urns. However, given that there are only a finite number of balls in our collection, any arrangement we select will necessarily take on the form of a discrete distribution. Ideally, we would like to cluster the balls into different urns, such that the final partition minimizes the heterogeneity of colors within an urn, while maximizing the differences in features across urns.

We begin the clustering procedure by randomly drawing a sample ball from our starting collection and placing it into an empty urn. For the next randomly drawn ball, we then must evaluate the posterior probability of sorting the new ball into the same urn as the initial ball, and compare it with the posterior probability of sorting it into a new empty urn. In predictive Bayesian inference, evaluation of posterior probabilities is dependent upon two component distributions: the likelihood function and the prior. In this scenario, our likelihood function determines how likely it is that both sampled balls belong to the same color family, given a set of parameters which define the specific color features of the other member of that urn, such as shade, saturation levels, and hue. (For the empty urn, there are no other members to infer information from regarding these color parameters, so the likelihood function automatically evaluates to 1.) In contrast, the

prior distribution takes into account how probable it is we were to have clustered the balls together independently of their shared color features. This distribution is essentially a function of our prior expectations on the total number of expected clusters we expect to find, predicated on the total size of our previously observed clusters, relative to the total number of our observations.

Figure 11: Dirichlet Process Illustration with Pólya Urns



	$P(2 \in U1 \mid 1 \in U1)$	$P(2 \in U2 \mid U2 = \emptyset)$
Likelihood	**~ 0.1**	**1.0**
Prior $\alpha = 1$	$\frac{\lvert Urn\ 1 \rvert}{(N-1)+\alpha}$	$\frac{\alpha}{(N-1)+\alpha}$
	0.5	**0.5**
Posterior	$\frac{.05}{.55}$	$\frac{.50}{.55}$
	~ 9 %	**~ 91 %**

Posterior $\frac{P(N \in UK \mid \{1,\ldots N-1\}) * \pi_K}{\sum_K Likelihood * Prior}$	**~ 2 %**	**~ 4 %**	**~ 80 %**	**~ 13 %**

Clustering Immunoglobulin Sequences with the Dirichlet Process

The model illustrated in Figure 11 demonstrates how the two components of Bayesian inference under the Dirichlet Process can be used to assign colored balls to different urns, dependent upon a set of commonly shared features. This model can also be applied to our project by using the same clustering procedure to identify putative alleles for the germline gene segment candidate libraries directly from immunoglobulin sequencing data.

Instead of sorting balls into urns based on shared color, we are sorting immunoglobulin sequences into separate clusters based on their shared sequence similarity. Just as we can average out variations in hue and saturation amongst individual balls within a given urn to estimate a set of common color features for that urn, we can also estimate the most likely unmutated common ancestor of the sequences belonging to a particular cluster by examining the nucleotide positional information of the individual sequences within that cluster.

Under this model, the inferred germline allele represents the ‘base distribution’ for a given cluster, and the individual immunoglobulin sequences within that cluster are the ‘random samples’ from that base distribution. As strings, each immunoglobulin sequence can itself be considered a type of multinomial distribution, where the nucleotides are a series of categorical random samples. Thus, we can think of our clusters of immunoglobulin sequences as ‘distributions of distributions’, just as we would in the case of the abstract infinite mixture model.

In the following sections, we derive the two individual components necessary for Bayesian statistical inference (the likelihood function, and the prior), as they are used in the context of inferring germline gene segments. We begin with the special case of clustering two immunoglobulin sequences, and then generalize to cases of $N > 2$. We then conclude with a discussion of how the Dirichlet Process clustering procedure is updated through a Gibbs sampling machine learning algorithm, which features simulated annealing as a metaheuristic for global optimization.

Bayesian Inference of Candidate Germline Alleles

Definitions of Sequence Transition Probabilities

Let s represent a nucleotide in sequence $\vec{\boldsymbol{s}}_{\boldsymbol{j}}$,, and let a represent the nucleotide from it was originally derived in ancestral sequence $\overrightarrow{\boldsymbol{a}_{\boldsymbol{i}}}$. We then define $P_k(s|a)$ as the probability that we would observe nucleotide s given a, at some fixed position k within the full sequences $\vec{\boldsymbol{s}}_{\boldsymbol{j}}$, and $\overrightarrow{\boldsymbol{a}_{\boldsymbol{i}}}$. From here, we make the limiting assumption that each individual nucleotide within a sequence will have evolved independently with respect to its neighbors. That is to say, $P_k(s|a)$ is independent of $P_{k+1}(s|a)$ and thus:

Eqn. 3:

$$P\left(\overrightarrow{\boldsymbol{s}_{\boldsymbol{j}}}\middle|\overrightarrow{\boldsymbol{a}_{\boldsymbol{i}}}\right) = \prod_{k=1}^{L} P_k(s|a) = P_1(s|a) * P_2(s|a) * \ldots * P_L(s|a)$$

For simplicity, further assume that sequences $\vec{\boldsymbol{s}}_{\boldsymbol{j}}$, and $\overrightarrow{\boldsymbol{a}_{\boldsymbol{i}}}$ are of identical length L, and no insertions or deletions have been introduced. Thus, a pairwise sequence alignment between $\vec{\boldsymbol{s}}_{\boldsymbol{j}}$, and $\overrightarrow{\boldsymbol{a}_{\boldsymbol{i}}}$ will contain no gaps. Appropriate treatment of insertions and deletions adds an extra layer of complexity, which is discussed in the following section.

Justification for Simplifying Assumptions

Strictly speaking, the positional independence assumption does not fully capture all of the known biological complexities of our system. For example, it is known that certain specific sequence motifs can generate 'hotspots' of somatic hypermutation, which originate during the cycles of affinity maturation. However, the limitations that this assumption places on our model are precluded by the necessary gains it provides towards the computational tractability of our algorithms by reducing the number of parameters required for modeling nucleotide substitution.

The most generalized nucleotide substitution model which assumes positional independence entails 16 unique parameters. However, without positional independence, our substitution models need to account for each sequence as its own functional unit, instead of breaking it into smaller pieces. This drastically expands the number of necessary transitional probabilities needing to be parameterized, specifically at a rate of 16^L, where L indicates the length of the potential sequence. Even relatively short sequences of length 10 would have over a million unique permutations, and over a trillion possible ancestor-descendent substitution probabilities. The resources required for modeling without this assumption on immunoglobulin variable region genes, which are roughly 300 nucleotides in length, would exceed the capabilities of even high-end supercomputing clusters.

Similarly, our algorithm at this time contains a restriction that all sequences assigned to a given cluster must be of equivalent length, which limits its capacity to model insertions and deletions. The work in our lab typically resolves the issues posed by

gaps in sequence alignments by implementing a mapping function which indexes the nucleotide positions of individual sequences against a global maximum length template. Figure 12 provides an example for how this mapping function would be defined for a two-sequence pairwise alignment, but can be further generalized to multiple sequence alignments.

Figure 12: Gap Example of Sequence Alignment Mapping Function

Figure 12: Pairwise sequence alignment shown containing one insertion and two deletions with standard indexing shown above (grey) and below (purple) individual sequences respectively; the global maximum length template index is the top sequence shown in (black), and the bottom two sequences show how mapping function changes the corresponding sequence indexes with respect to this template.

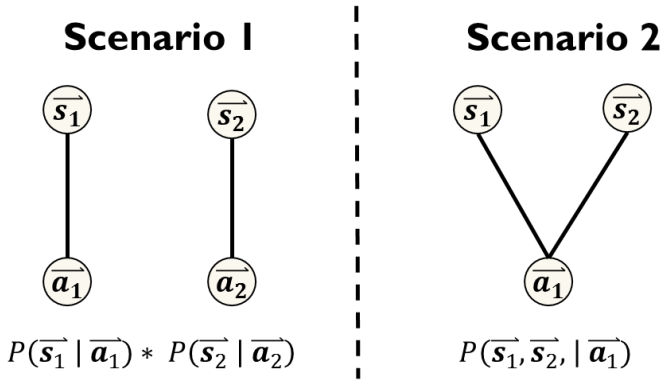
However, this framework has been difficult to properly implement for our Dirichlet process clustering paradigm due to questions it raises regarding the proper evaluation of the likelihood function. A gap represents a type of missing information about a given nucleotide position that is fundamentally different from the case when a nucleotide is known to be present, but its identity is uncertain. In the latter case, we can encode that uncertainty through a uniform probability mass function, but the former case is not so straightforward. This is because is it unclear how an absence of evidence (or non-observations) should be weighted relative to actual nucleotide observations. We

recognize our methods currently sidestep this underlying issue, and we look at this as an opportunity for future development. For now, the restriction is itself implemented by pre-separating sequences into subgroups of equivalent length, and then follow the Dirichlet process clustering procedure on each of the subgroups independently.

Definitions of Probabilities for Cluster Membership in 2-Sequence Case

Our goal is to create clusters whose member sequences are all ultimately descended from the same unobserved common ancestor sequence, i.e. a candidate germline gene segment allele. For example, if two sequences $\overrightarrow{\boldsymbol{s_1}}, \overrightarrow{\boldsymbol{s_2}}$ have both been assigned to a given cluster c_1, we are making a claim that both sequences are derived from the same ancestor sequence $\overrightarrow{\boldsymbol{a_1}}$. Alternatively, if these two sequences are assigned to separate clusters, we are making a claim that these sequences are derived from two separate ancestors $\overrightarrow{\boldsymbol{a_1}}$ and $\overrightarrow{\boldsymbol{a_2}}$ respectively.

Figure 13: Two ancestor vs. one ancestor model[32,33]



We can make quantitative inferences about the quality of our clustering assignments, by measuring the overall sequence similarity between cluster members, and

evaluating the probability of each given clustering arrangement in the context of our nucleotide substitution models.

For example, if $P(\overrightarrow{s_1}, \overrightarrow{s_2}, \mid \overrightarrow{a_1}) > P(\overrightarrow{s_1} \mid \overrightarrow{a_1}) * P(\overrightarrow{s_2} \mid \overrightarrow{a_2})$, then it is more probable that sequences $\overrightarrow{s_1}, \overrightarrow{s_2}$ are derived from a shared common ancestral gene segment than they are from separate gene segments. We can use this concept to build an iterative clustering procedure which evaluates the likelihood that a new 'probe' sequence belongs to an existing cluster based on shared nucleotide content, over a novel empty cluster.

Full Derivation of Likelihood Component: generalizing our model for iterative clustering

Suppose we have some cluster c_i which already contains a collection of sequences $\overrightarrow{s_1}, \overrightarrow{s_2}, \ldots, \overrightarrow{s_{|c_i|}}$, and that each of these sequences are derived from some ancestral sequence $\overrightarrow{a_i}$.

We are interested in solving for $P\left(\overrightarrow{s_{new}} \mid \overrightarrow{s_1}, \overrightarrow{s_2}, \ldots, \overrightarrow{s_{|c_i|}}, \overrightarrow{a_i}\right)$. In other words, we wish to quantify how probable it is that a newly observed 'probe' sequence $\overrightarrow{s_{new}}$ belongs to cluster c_i, given all of the current assigned members of cluster c_i, and the common ancestral sequence $\overrightarrow{a_i}$.

From the axioms of probability, we can deduce:

Eqns. 4.1-4.3:

$$P\left(\overrightarrow{s_{new}} \middle| \overrightarrow{s_1}, \overrightarrow{s_2}, \ldots, \overrightarrow{s_{|c_i|}}, \overrightarrow{a_i}\right) = \frac{P\left(\overrightarrow{s_{new}}, \overrightarrow{s_1}, \ldots, \overrightarrow{s_{|c_i|}}, \overrightarrow{a_i}\right)}{P\left(\overrightarrow{s_1}, \ldots, \overrightarrow{s_{|c_i|}}, \overrightarrow{a_i}\right)} \quad \text{Defn. of Marginal Probabilities}$$

$$= \frac{P\left(\overrightarrow{s_{new}}, \overrightarrow{s_1}, \ldots, \overrightarrow{s_{|c_i|}} \mid \overrightarrow{a_i}\right) * P(\overrightarrow{a_i})}{P\left(\overrightarrow{s_1}, \ldots, \overrightarrow{s_{|c_i|}}, \mid \overrightarrow{a_i}\right) * P(\overrightarrow{a_i})} \quad \text{Separation of Joint Probability into Marginals \& Priors}$$

$$= \frac{P\left(\overrightarrow{\boldsymbol{s_{new}}}, \overrightarrow{\boldsymbol{s_1}}, \ldots, \overrightarrow{\boldsymbol{s_{|c_i|}}} \mid \overrightarrow{\boldsymbol{a_i}}\right)}{P\left(\overrightarrow{\boldsymbol{s_1}}, \ldots, \overrightarrow{\boldsymbol{s_{|c_i|}}}, \mid \overrightarrow{\boldsymbol{a_i}}\right)}$$ Cancel Like Terms

We define $P(\overrightarrow{\boldsymbol{a_i}}) = (¼)^L$ as our uninformative prior for the starting content of our ancestor sequences.

We also define:

Eqn. 5: $$\left(\overrightarrow{\boldsymbol{s_1}}, \ldots, \overrightarrow{\boldsymbol{s_{|c_i|}}}, \mid \overrightarrow{\boldsymbol{a_i}}\right) = \prod_{k=1}^{L} \left(\max_{s \in \{A,T,G,C\}} \left(\prod_{j=1}^{|c_i|} \mathcal{L}_{ijk}(s|a) \right) \right).$$

$\mathcal{L}_{ijk}(s|a)$ is the likelihood function of observing a nucleotide s in the $k$kth position of sequence j for cluster c_i, given some unknown nucleotide a in the same position of the ancestral template sequence of cluster c_i. This function is evaluated according to Kimura's 1980 nucleotide substitution evolutionary model.

$P\left(\overrightarrow{\boldsymbol{s_1}}, \ldots, \overrightarrow{\boldsymbol{s_{|c_i|}}}, \mid \overrightarrow{\boldsymbol{a_i}}\right)$ is thus the maximum likelihood function for observing sequences $\{\overrightarrow{\boldsymbol{s_1}}, \ldots, \overrightarrow{\boldsymbol{s_{|c_i|}}}\}$, given that they all are descendants of specified common ancestor sequence $\overrightarrow{\boldsymbol{a_i}}$.

$P(\overrightarrow{\boldsymbol{s_1}} \mid \overrightarrow{\boldsymbol{a_i}})$ is defined with Eqn. 3 above for the first element of a cluster c_i. Since we are interested in evaluation the likelihood of a sequence $\overrightarrow{\boldsymbol{s_{new}}}$ joining cluster c_i, relative to the pre-existing sequences already present in cluster c_i, we take a likelihood ratio test of the two components.

By combining Eqns. 4.3 and Eqn. 5, we arrive at the following for our final likelihood formula:

Eqn. 6:

$$\left(\overrightarrow{\boldsymbol{s_{new}}}|\overrightarrow{\boldsymbol{s_1}}, \overrightarrow{\boldsymbol{s_2}}, \ldots, \overrightarrow{\boldsymbol{s_{|c_i|}}}, \overrightarrow{\boldsymbol{a_i}}\right) = \left(\frac{\prod_{k=1}^{L}\left(\max_{s\in\{A,T,G,C\}}\left(\prod_{j=1}^{|c_i|+1}\mathcal{L}_{ijk}(s|a)\right)\right)}{\prod_{k=1}^{L}\left(\max_{s\in\{A,T,G,C\}}\left(\prod_{j=1}^{|c_i|}\mathcal{L}_{ijk}(s|a)\right)\right)}\right)$$

Derivation of Prior Component

In the previous section, we derived the principle formulas used for the evaluating the component for the likelihood function of our in Bayesian inference clustering procedure. In this section, we will discuss the determination of the formulas required for the prior probability components, as derived from a Dirichlet Process.

π_i designates the prior probability of a sequence being assigned to nonempty cluster c_i, while π_0 designates the prior probability of a sequence being assigned to a new (currently empty) cluster.

Eqns 7.1-7.2:

$$\pi_i = \frac{|c_i|}{N + \alpha}$$
$$\pi_0 = \frac{\alpha}{N + \alpha}$$

$\boldsymbol{|c_i|}$ is the number of sequences already present in cluster c_i (excluding the most recent 'probe' sequence) and N is the number of sequences previously assigned to all clusters c_i, $\forall\, i$. Note that $N = \sum_i |c_i|$.

α is a scaling parameter, determined prior to beginning the clustering procedure by the user. For this project, the selection of an appropriate α was determined as the result of empirical testing via synthetic data, and is discussed at length in Chapter 4.

Gibbs Machine Learning for Cluster Reassignment & Library Inference

In the previous section, we discussed the role of the two major components, the likelihood and the prior, of the Dirichlet process clustering procedure in our Bayesian statistical model. Here we review how each component comes together as part of an iterative Gibbs machine learning engine for generating a proposal of clustering assignments. At the conclusion of the clustering procedure, the initial candidate set of internal gene segment libraries will be derived from the clustering arrangement by inferring the most likely candidate for the ancestral template sequences of each cluster.

The DP engine is initialized by randomly selecting a single sequence from within the input sample from high-throughput immunoglobulin-sequencing, and assigning it to an empty cluster. This will serve as a seed sequence from which all other cluster assignments are based. For the second randomly selected sequence (and every sequence in the input dataset thereafter), a posterior probability of cluster assignment is calculated using the designated formulas of the likelihood and prior components, for each potential cluster assignment. This ‘probe’ sequence is then randomly assigned either to an existing cluster, or to a new empty cluster, with its choice of placement weighted according to this marginalized posterior. This assignment continues until every input sequence has been assigned a cluster label, thus completing the first round of clustering.

For each subsequent round of clustering, each sequence is given an opportunity to be reassigned to alternative clusters. This is because for the majority of sequences, the available information regarding the current clustering state will be different from the conditions during initial assignment, as newer sequences will have since then been

observed and placed accordingly. Any reassignment of sequences will also result in a change in the overall cluster state, which in turn alters the marginalized posterior probabilities for further sequence assignments, and represents an update in the understanding of a cluster's inferred ancestral sequence. During each round of reassignment, each sequence is given the opportunity to be reassigned once, in randomized order. This clustering procedure is currently set to terminate after 50 rounds, an empirically derived upper limit on the time it takes for the system to consistently reach a steady state. Our preliminary trials indicated that 50 rounds was more than sufficient to achieve steady state, under a wide range of the parameters α (defined in earlier section on priors) and β (to be defined in the following section). However, given the stochastic nature of the Dirichlet process, we recognize that the system is not guaranteed to reach steady state within 50 rounds, and so we plan to provide a feature for adjusting this limit and recording the clustering rearrangement movements as part of the advanced settings in our user interface. Figure 14 provides an example of one of these preliminary trials.

Figure 14: Sequence Migration Plot

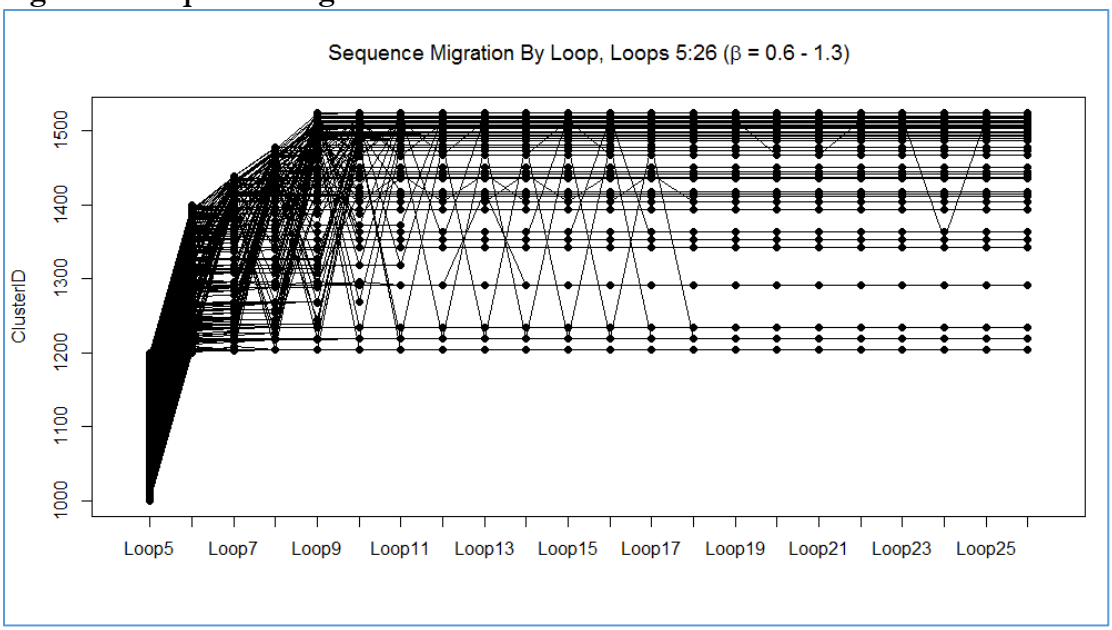


Figure 14: The x-axis refers to the round number, while the y-axis refers to a unique cluster index. The lines indicate the path of sequence migrations between clusters over time. There is substantial sequence reassignment and migration during the early rounds, but later rounds have reduced movement and a 'crystallization' of cluster assignments.

Simulated Annealing[34,35]

Many of the machine learning algorithms developed to tackle optimization problems in mathematics and computer science have a significant drawback in that they tend to get stuck at local optima. This is typically caused by the 'greedy' nature of such algorithms, which require that the only accepted steps are ones which improve the overall scoring function. Finding algorithms which can guarantee globally optimal solutions remains a formidable open question in computer science. One popular alternative has been to instead approximate the global optimum by applying a metaheuristic called 'simulated annealing', which allows for the temporary exploration of 'worse' regions in overall solution space.

The concept of simulated annealing is analogous to a common problem in materials science involving the heating and cooling of metals. At high temperatures, the overall free energy of the system is greater, and so molecules have the freedom to explore less optimal configurations. The rate at which the metal is cooled will affect the overall size of the final crystals, as well as the defects within the crystals. By controlling the rate of cooling, larger crystals can be achieved than would otherwise be possible.

We can implement simulated annealing into our clustering procedure by making one small adjustment to our likelihood components in Eqn. 6, adding an additional parameter β.

Eqn. 7:

$$P\left(\overrightarrow{\boldsymbol{s_{new}}}|\overrightarrow{\boldsymbol{s_1}},\overrightarrow{\boldsymbol{s_2}},\dots,\overrightarrow{\boldsymbol{s_{|c_i|}}},\overrightarrow{\boldsymbol{a_i}}\right)=\left(\frac{4^{-L}*\prod_{k=1}^{L}\left(\max_{s\in\{A,T,G,C\}}\left(\prod_{j=1}^{|c_i|+1}\mathcal{L}_{ijk}(s)\right)\right)}{4^{-L}*\prod_{k=1}^{L}\left(\max_{s\in\{A,T,G,C\}}\left(\prod_{j=1}^{|c_i|}\mathcal{L}_{ijk}(s)\right)\right)}\right)^{\beta}$$

Here, β represents an inverse temperature parameter. For smaller β, the overall temperature (and 'free energy') of the system is increased, while larger β indicates cooler temperatures. We can simulate a controlled cooling by beginning with low values of β in the early rounds of the Gibbs machine learning process, and gradually increasing it as the number of rounds increases. A **β** set equal to one across all rounds of clustering would indicate a 'non-annealing' schedule. Generally speaking, at lower β (higher temperature), more weight is given to the prior component of the Dirichlet Process, and sequences are thus more likely to migrate to alternative clusters than would normally be indicated solely by their shared nucleotide content. Conversely, at higher β (lower temperatures), more

weight is given to the likelihood component, and sequences eventually come to settle or ‘crystalize’ into the clusters from which they originate in a given round.

The selection of an appropriate annealing scheme for β was also determined empirically through preliminary trials with synthetic data. We include a review of these trials in an appendix. The scheme we selected for our algorithm evaluations discussed in Chapters 4 & 5 set $\beta = 0.5$ as an initial value, and increased it by 0.1 after every 3 rounds of reassignment for a total of 50 rounds.

.

CHAPTER FOUR: EVALUATION OF DEVELOPED METHODS WITH SYNTHETIC DATASETS

Introduction

In the previous two chapters, we discussed the details of the statistical methods and algorithms used to arrive at a set of candidate model germline gene libraries, and a variable region analysis of the immunoglobulin sequencing data given those inferred libraries. In the next two chapters, we will discuss the means by which these statistical methods and algorithms are evaluated for both accuracy and robustness across a wide range of parameter settings.

The evaluation tests described in this chapter will concentrate on the use of synthetically generated data, whereas the tests described in the following chapter will use data derived from actual biological sources. We begin with a discussion of the advantages and disadvantages associated with using simulated data to evaluate our methods. We follow with an explanation of the types of outcomes we are interested in measuring in our trials with synthetic data, as well as our definitions of what differentiates a successful run versus an unsuccessful run. We continue with an overview of the conditions used to generate a synthetic dataset, and conclude with a detailed breakdown of the primary results of the various trials run on those datasets.

Advantages and Disadvantages of Synthetic Data Trials

The primary advantage of utilizing synthetic data is the certainty it provides to the experimenter regarding the ground truth of the different sources of variation within the generated datasets. For example, having knowledge about both germline level variation

(e.g. true number and sequence identity of the alleles used during recombination), and clonal level variation (e.g. statistics regarding number of clones, their founder sequences, and any mutations acquired during a simulated process for affinity maturation) allows the experimenter to isolate the two primary sources of dataset variability and assess their effects on any given clustering trial independently.

Moreover, the level of certainty that one can have using synthetic data in the interpretation of final results has tremendous power in identifying opportunities for further algorithm improvements. By identifying discrepancies between predicted results and actual outcomes for synthetic data, the experimenter can probe the limits of the developed methods under a wide variety of initial conditions and model assumptions with a high degree of precision. However, due to the hierarchical complexity of our machine learning system, it is imperative that the experimenter remains vigilant against 'parameter-hacking', or an overly fine-tuning of selected parameters in order to optimize for a candidate set of ideal results, that do not translate well over to biological sources. In addition to preventing over-tuning, trials using biological data sources are necessary because they contain an inherent variability that may not be completely captured by our existing models. The reasoning behind this is analogous to research studies which utilize both in vitro and in vivo experiments to probe their system of choice.

Key Measurable Outcomes for Synthetic Data Trials

There are three categories of results that we are concerned with evaluating. First, we are concerned with evaluating our predictions of the germline gene segment libraries used in our VDJ recombination models. Evaluation of this category will be concerned

with measuring both the quantity and quality of our predicted germline gene segment alleles. Specifically, we are testing to see whether the machine learning algorithm is able to accurately infer both the total number of alleles in our starting germline gene library, and the actual nucleotide content of those inferred alleles. A successful trial would be produce a candidate set of libraries whose sequences were identical to those in the starting library, whereas an unsuccessful trial would fail to meet at least one of those strict requirements.

The next two categories of results have to do with the algorithm's capacity to perform typical immunoglobulin repertoire analysis methods (given the inferred candidate libraries), specifically in regards to variable region gene (VRG) segment assignment and clonal partitioning. For VRG assignments, we are interested in comparing how individual reads are annotated using the candidate alleles from our inferred libraries, and whether their assigned recombinations match those from their original source data. For assessing clonal partitioning, we are interested in whether the algorithm is able to accurately recapture the total number of clones, the accuracy of each clone's inferred founder specifically in regards to their rearrangement parameters, the assignment of individual sequences to clones, and the determination of acquired mutations within the clonal lineage.

Taken in conjunction, each of these categories of results will be used to assess the algorithm's capacity to differentiate between germline gene variation and acquired somatic hypermutation variation in a synthetic context.

Generation of Datasets for Synthetic Trials

Synthetic Allele Libraries

The synthetic library we selected is a curated subset of the functional human immunoglobulin heavy chain V segments available from IMGT, and contains pairs from a total of ten unique gene segments derived from both human IGHV3 and IGHV4 families of genes. Alleles were selected in order to provide identical SNP-distances between allele pairs, in order to control for germline variation while testing the influence of parameter choice on clustering capabilities. A SNP-distance refers to the metric which measures the total number of single nucleotide polymorphisms (SNPs) in the optimal alignment for a given sequence pair e.g. if alignment AB has a SNP-distance of 3, then alleles A and B contain a total of 3 string-level mismatches. Figure 15 shows a maximum likelihood tree of the 10 unique alleles used in the starting V gene segment library used for synthetic VDJ recombination. Note that each allele pair has equivalent distance of 3 SNPs, but there is an increased distance between gene segments, and an even greater distance between the V3 and V4 gene families.

Figure 15: Allelic Variation of Synthetic V Gene Segment Library

Figures 16a-b represent the distribution of SNP-distances from all possible pairwise alignments between any two alleles found in the IMGT's human V gene segment database. In Figure 15a, the distances are colored according to whether the alleles in a given pair belonged to the same gene family (blue) or not (red). (E.g. IGHV1-23*01 & IGHV1-46*02 (blue) vs. IGHV1-23*01 & IGHV3-33*03 (red)) In Figure 15b, the distances are colored according to whether the alleles in a given pair belonged to the same gene segment (blue) or not (red). (E.g. IGHV1-23*01 & IGHV1-32*05 (blue) vs. IGHV1-23*01 & IGHV1-46*02 (red))

Figure 16a: Histogram of Interfamily Allelic Variation

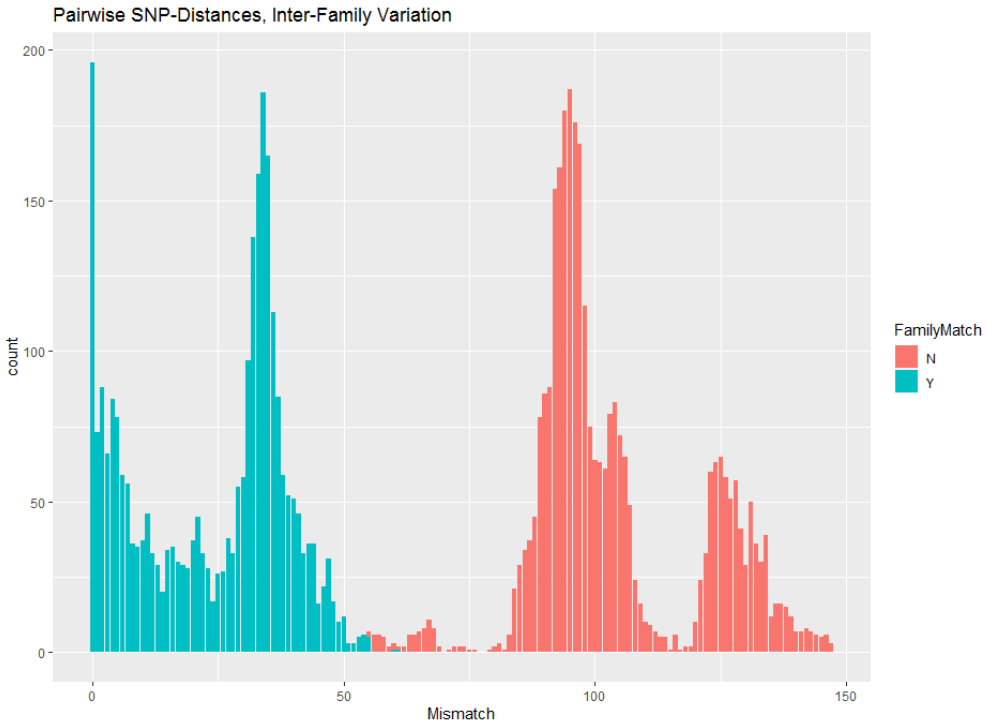
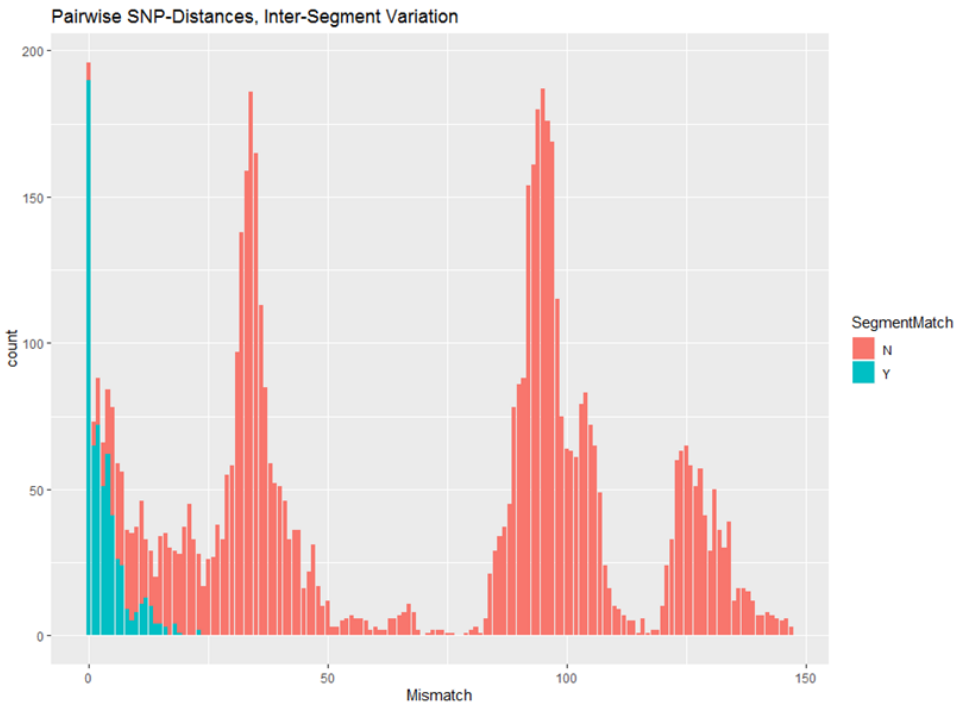


Figure 16b: Histogram of Intersegment Allelic Variation



As a general rule of thumb, intersegment variation will most often have fewer than 10 SNPs between alleles, while interfamily variation will have on the order of dozens of SNPs.

As we were primarily interested in evaluating the developed clustering methods on inferring V gene libraries alone, we used the standard D and J gene segment libraries available from IMGT for these simulated recombinations.

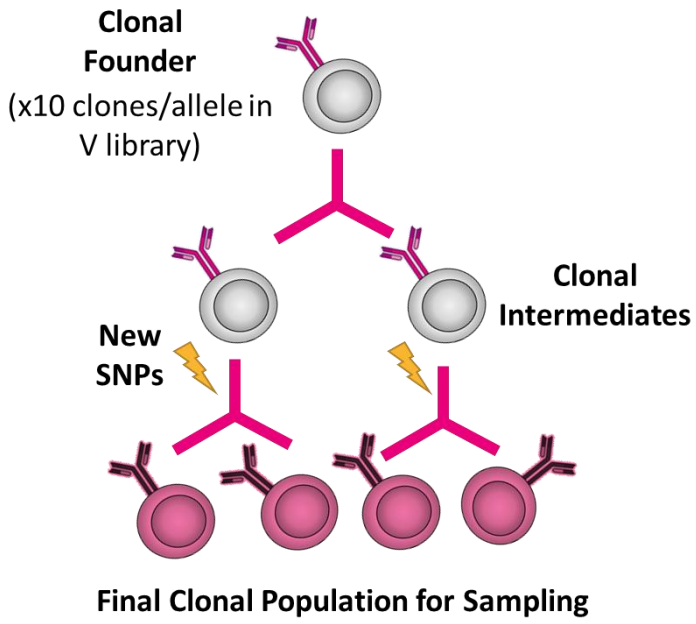
Generating Synthetic Clones[32,33]

Every allele in the V starting library was recombined with D and J gene segments stochastically selected from their respective libraries to form a founder sequence for a particular clone. Each V allele was used in the founder sequence of at least 10 unique clones. These clones were directed to propagate for 4 generations at a pre-specified mutation rate. For each generation, 2 'child' sequences were duplicated from every 'parent' sequence in the previous generation, with mutations stochastically applied to the child sequences according to the pre-specified mutation rate. Since immunoglobulin sequences are roughly 300 bp in length, a mutation rate of 0.001 would translate to approximately 1 new mutation per generated sequence. The statistical model for applying the mutations uses a weighted uniform distribution, with weights drawn from estimates of pentameric nucleotide motifs empirically-derived from non-productive rearrangements (T. Kepler, personal communication, August 22, 2019).

Eqn. 8:

$$Population = alleles * founders * 2^{generations} = 10 * 10 * 2^4 = 1600\ sequences$$

Figure 17: Generation of Synthetic Clones for Alleles in Starting V Gene Library



We generated three synthetic clonal populations; their mutation rate parameters were selected such that the average mutation frequency amongst final generation of sequences compared to their founders would be 0% (control/low), 2% (medium), and 5% (high) respectively.

Filtering Synthetic Clonal Populations; Artificial Selection of Clones

During the course of accumulating mutations, our synthetic clones would often mutate away the conserved cysteine critical to immunoglobulin folding. This was especially prominent in our high mutation frequency population, which had a higher probability of introducing the mutation in an earlier generation, and would then propagate the mutation to subsequent generations. This phenomenon marks a departure of our model for synthetic clone generation from actual biological systems, which would eliminate the mutation through natural selection, as any immunoglobulin receptors which

contained this mutation would by definition be non-productive. Since the presence of this conserved cysteine is a critical component to both actual productive immunoglobulins and our algorithm's capacity to distinguish the V portion of our immunoglobulin sequences, we generated a surplus of clones for each V allele, and then filtered such that every clone would preserve the conserved cysteine's codon for all of its descendent sequences.

Summary of Synthetic Datasets

In addition to overall mutation frequency, we were interested in testing the influence of both the total size of the dataset (N), and the choice of the hyperparameter $log\ (\alpha)$ on algorithm performance. This is because under a standard Dirichlet process (i.e. when priors alone determine cluster membership, in the absence of a likelihood function), the number of expected clusters on N sequences is expected to converge at a rate proportional to $\alpha\ log\ N$. (Our interest in $log\ (\alpha)$ rather than α stems from by applying log transformations to the likelihood and prior components prior to evaluating the corresponding posterior probabilities in order to prevent memory overflow errors during computation.)

To generate populations of varying size (N), we further filtered the original three synthetic populations (control, medium, high mutation frequency) to create four subpopulations for each group (12 datasets in total). The four subpopulations within a group form a successive chain of subsampling, so that every sequence sampled in the smaller populations are present in the next largest sample of its type. For example, all of the sequences contained in dataset 4 are also contained within dataset 3, which in turn are

all contained within dataset 1, etc. Table 2 below lists these twelve datasets, along with their individual parameters for clone size, counts, and total allele coverage.

Table 2: Summary of Synthetic Datasets

SAMPLE ID	MUTATION FREQUENCY	# CLONES	# SEQS / CLONE	TOTAL SEQS / ALLELE	TOTAL SEQS (N)
1	0%	10	10	100	1000
2	0%	8	8	64	640
3	0%	5	5	25	250
4	0%	3	3	9	90
5	2%	10	10	100	1000
6	2%	8	8	64	640
7	2%	5	5	25	250
8	2%	3	3	9	90
9	5%	10	10	100	1000
10	5%	8	8	64	640
11	5%	5	5	25	250
12	5%	3	3	9	90

Each of the twelve generated synthetic datasets was run through a total of eight trials, where $log\ (\alpha)$ was given a value from the set $\{500,300,250,200,175,150,75,40\}$ for a total of 96 synthetic trials. We selected these values for $log\ (\alpha)$ based on predictions made by calculations for an expected 'switchpoint' or transition between preferential assignments of newly observed sequences to existing clusters vs. generating novel clusters. This switchpoint was partly dependent on the length of the sequences being clustered, which for V gene sequences is ~300bp.

The simulated annealing cooling scheme was kept constant across all synthetic trials. In this cooling scheme, the initial starting value for beta was 0.5, and then increased by 0.1 after every third iteration of Gibbs sampler, up to a final value of 2.1. These parameters had also been determined empirically during the early stages of Gibbs

sampler implementation, and were optimized to achieve a steady state of clustering assignments within 50 rounds of attempted cluster rearrangement.

Results

Effects of Mutation Frequency, N and $log\ (\alpha)$ on Final Library Size

Figures 18a - 18d show bar charts which display the final size of the predicted germline V gene libraries for each of the 96 synthetic data trial runs. The y-axis refers to the number of predicted alleles, while the x-axis for each chart refers to the mutation frequency of the population that the datasets were derived from. Each different colored bar represents a different choice for the parameter $log\ (\alpha)$. The black line indicates the true number of alleles for that dataset, which is 10.

Figure 18e reformats the data present in Figures 18a-d. Whereas Figures 18a-d emphasize the contrasting effects of differing $log\ (\alpha)$ settings within a given dataset size, Figure 18e emphasizes the contrast between dataset sizes for a given $log\ (\alpha)$ setting and mutation frequency. As before, the y-axis indicates the size of the final predicted library, the black line indicates the true number of alleles (10) and charts have been grouped by overall dataset size. However, in Figure 18e, the x-axis now represents the selection of the $log\ (\alpha)$ parameter, and the color of the bars indicate the population's mutation frequency.

Figures 18a-18d: Predicted Alleles Chart, (90, 250, 640, 1000 Sequences)

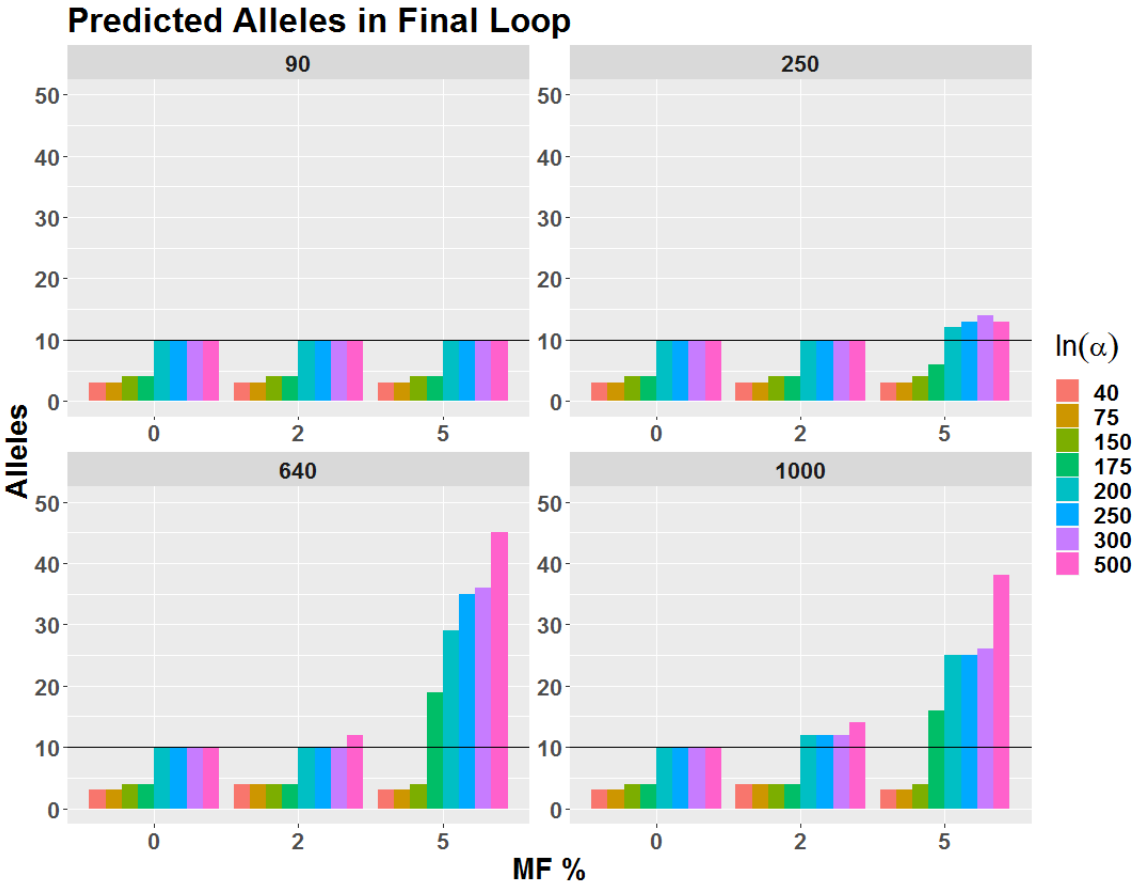


Figure 18e: Predicted Alleles Chart, Alternate Emphasis

We find that the algorithm consistently underestimates the true number of germline alleles when the parameter $log\ (\alpha) \leq 150$, regardless of dataset size or population mutation frequency. For values of $log\ (\alpha) \geq 175$, the size of final predicted libraries varies with both dataset size and population mutation frequency.

Figure 18a demonstrates that the datasets which contained the fewest sequences (4, 8, 12) were able to correctly guess the true number of alleles, regardless of population mutation frequency when $log\ (\alpha) \geq 200$.

Figures 18b-d demonstrate that populations with 0% mutation frequency and $log\ (\alpha)$ settings greater than 150 are able to correctly predict the total number of alleles, regardless of library size. We are also able to correctly predict the true number of alleles in the 2% mutation frequency populations, as long as the dataset is small enough.

However, the algorithm overestimates the true number of germline alleles for trials with larger datasets and higher mutation frequencies. For all but the smallest datasets, a population of 5% mutation frequency results in an overestimation of the true number of alleles.

Effects of Mutation Frequency, N, $log\ (\alpha)$ on Quality of Allele Prediction

The quality of our predicted allele libraries was measured through exhaustive pairwise comparisons of predicted allele libraries with external human IGHV libraries. This analysis demonstrated that the synthetic trials which had managed to correctly predict the true number of alleles (10) were also able to accurately infer the sequence content of those alleles. This perfect string-wise matching of predicted alleles with external alleles held constant across all trials which had correctly guessed 10 alleles, regardless of mutation frequency, $log\ (\alpha)$ or N.

For every synthetic trial which had predicted a final library larger than the true number of alleles (>10 sequences), there was always a 10-sequence subset of alleles which had string-wise perfect matches to the original library. The extraneous candidate alleles in the overestimated libraries generally had poorer clonal support than the correctly predicted subset, as shown in Figure 19 below.

Figure 19: Box Plot of Average Clonal Support in Overestimating Synthetic Trials

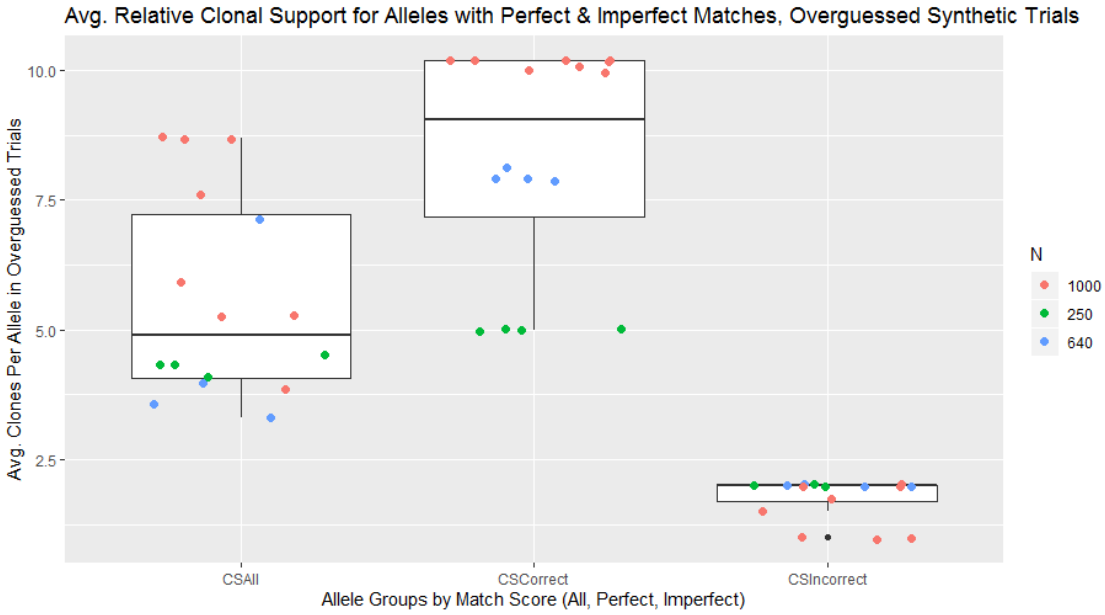
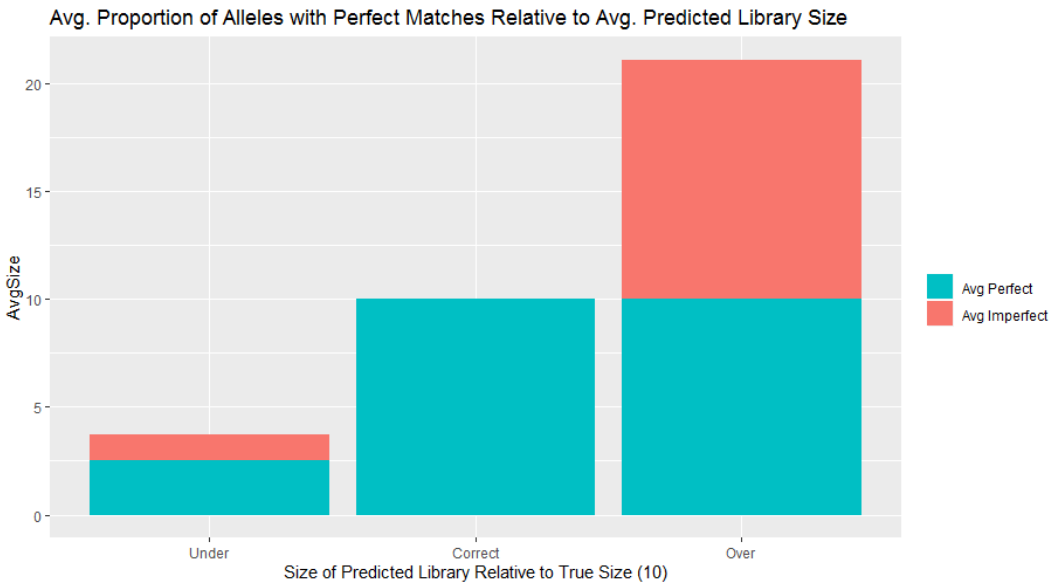


Figure 20: Proportion of Alleles which were perfect vs. imperfect matches relative to predicted library size, averaged over all synthetic trials



Trials which predicted smaller allele libraries than the original (<10 sequences) had variance in their overall quality prediction based on population mutation frequency.

None of the underestimating trials with 0% mutation frequency populations contained any perfect matches to the original library. In contrast, the 2% and 5% mutation frequency trials in this group typically had all but one or two of their predicted alleles be perfectly string-wise matched to a query allele in the original library. These remaining one or two sequences from the underestimated libraries would have much higher mismatch scores for all of the alleles in original library.

These results are unsurprising given the nature of our clustering approach. Essentially, if the $log\ (\alpha)$ parameter is set too low, then the sequences will not separate into distinct clusters that reflect their true allelic origins, and instead gravitate towards one or two large 'miscellaneous grab bag' clusters. As these clusters grow larger, true allelic variation is averaged out and the inferred consensus sequence for that cluster becomes a kind of non-informative magnet for future assigned sequences. In contrast, if the $log\ (\alpha)$ is set too high, then sequences are more prone to separate and form smaller clusters. In many cases these smaller clusters only contain one or two sequences on their own, usually from the same clone. When this occurs, these singlets and doublet clusters do not contain adequate interclonal information to estimate the true allelic variation, causing the predicted libraries to be populated with extraneous alleles which contain poor clonal support.

Effects of Mutation Frequency, N, $log\ (\alpha)$ on VDJ assignments & Clonal Lineage Inference Final Clone Prediction

Figures 21a – 21d contain bar charts which show how the final number of predicted clones for a given synthetic trial compared with the true number of clones

(black horizontal lines) in its respective dataset. The specific layout of the axes and color legends are identical with the corresponding format in Figures 18a-18d, and are chosen to emphasize the compounding influence of N and mutation frequency on algorithm performance. Similarly to Figure 18e, Figure 21e reverses the orientation of the x-axis group labels and the color legend in order to emphasize the influence of varying the parameter $log\ (\alpha)$ on algorithm performance.

Figures 21a-21d: Predicted Clones Chart (90, 250, 640, 1000 Sequences)

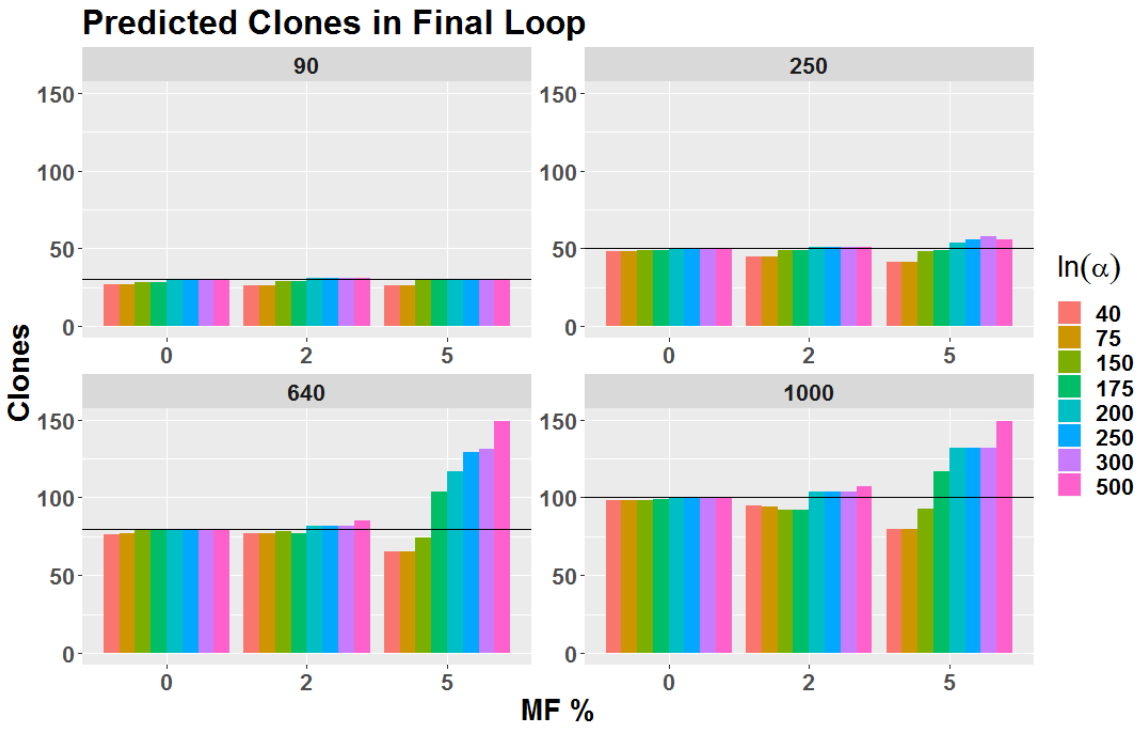
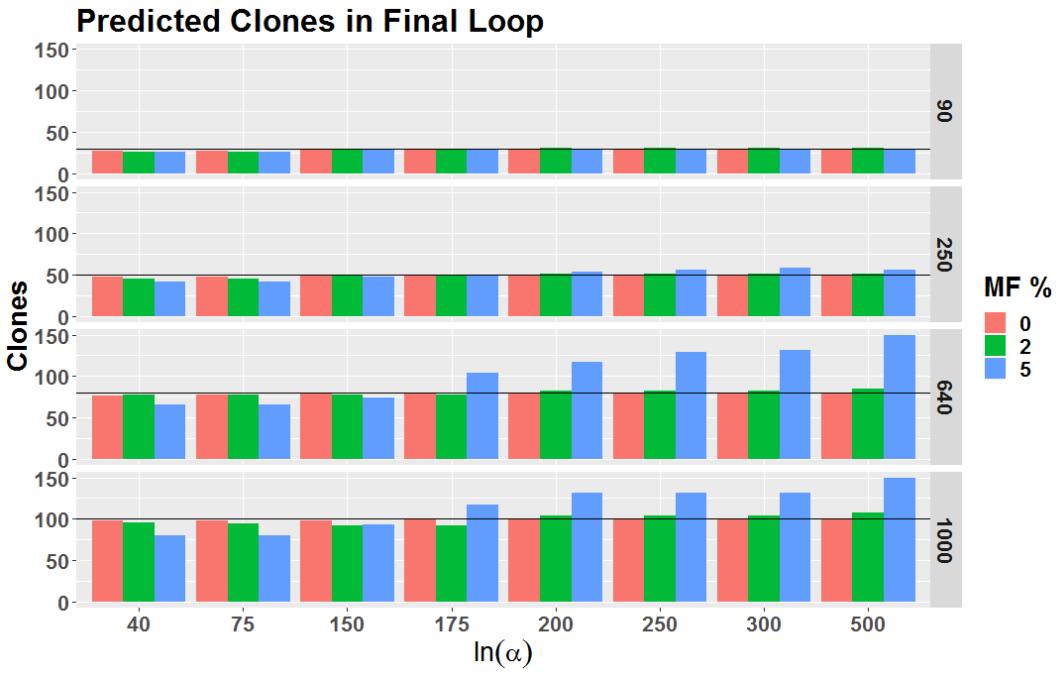


Figure 21e: Predicted Clones Chart, Alternate Emphasis



In general, we see concurrent results to that of predictions of allele quantity and quality. Smaller dataset sizes are closer in approximating the true number of clones than larger datasets, as are populations with smaller mutation frequencies. In general, choice of $log\ (\alpha)$ appears to be less significant in the determination of final clone counts, except in cases when both datasets are large and mutation frequency is high.

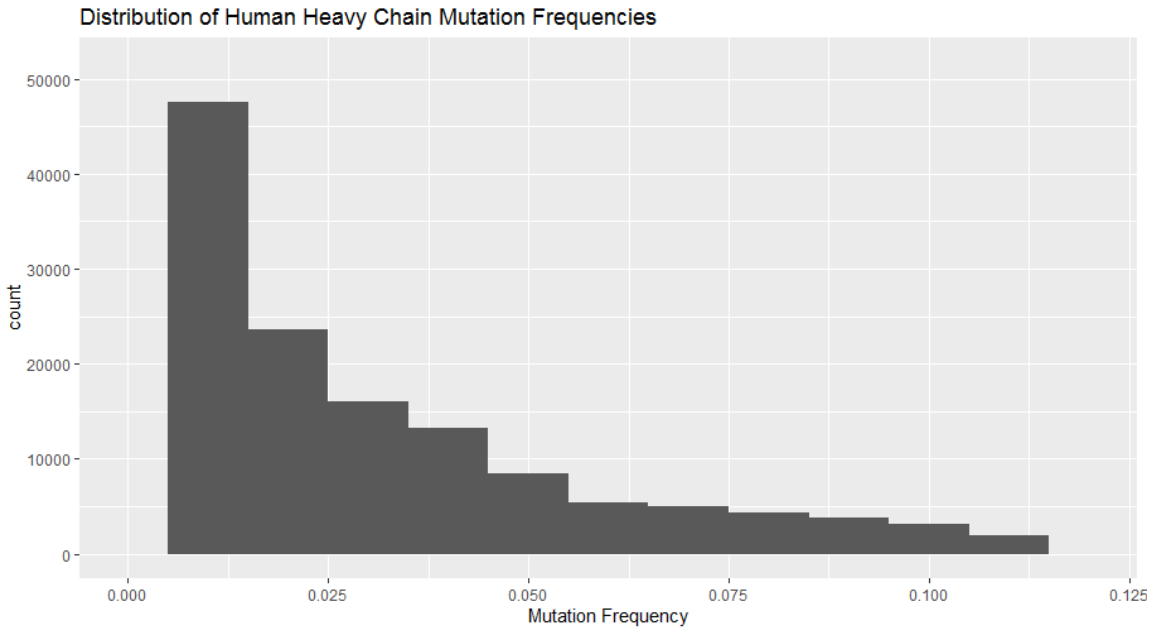
Synthetic Trial with Mixed Clonal Populations of Varying Mutation Frequencies

The relative decrease in algorithm performance at higher mutation frequency populations is not that surprising, given the decreased certainty in available information for allelic inference. However, in a realistic biological scenario, there will be a mixture of mutation frequencies amongst the different clonal subpopulations, with some clones being highly differentiated with greater numbers of mutations and other less-

differentiated clones with fewer overall mutations. We decided to test whether the presence of these less-differentiated clones would be enough to ‘rescue’ the inference of the overall population.

To do this, we used antibody data collected as part of an earlier study on acquired mutations of immunoglobulins in HIV-1 infected individuals. This data came from bulk-DNA sequencing from 75 human subjects. We used it to estimate the shape of the underlying distribution of mutation frequencies across clones. Figure 22 shows a histogram of this data, which approximates that of a power law distribution. Low mutation frequency clones make up the majority of clones, followed by a long tail of progressively higher mutation frequency clones.

Figure 22: Histogram of Human Heavy Chain Mutation Frequencies



To approximate this biological power law distribution using our previously simulated data, we created a new “mixed” dataset, which combined all of the sequences

from datasets 2, 6, and 12. In this mixed dataset, there were a total of 980 sequences with the largest subpopulation contained 640 sequences (8 clones/ allele, 8 sequences/clone) at 0% mutation frequency; the second subpopulation had 250 sequences (5 clones/allele, 5 sequences/clone) at 2% mutation frequency; the final subpopulation had 90 sequences (3 clones/allele, 3 sequences/clone) at 5% mutation frequency.

When setting the parameter $log\ (\alpha) = 300$, we find that the algorithm is able to accurately predict both the true number of alleles (10) in this mixed proportions dataset, and every allele in this library is a perfect match with an allele from the original starting library. As with other trials, it also came very close to the true number of clones (predicted 159 vs. an actual 160). The sizes of the predicted clones were remarkably consistent with the true structure of the dataset, with 88% of predicted clones being the correct size, and the overall proportion approximating the original 8:5:3 ratio reflective of the underlying subpopulations.

CHAPTER FIVE: APPLICATIONS TO NOVEL BIOLOGICAL DATA & DISCUSSION

The purpose of this chapter is to evaluate the robustness of our developed machine learning algorithms in the context of real-world biological data by applying them to human immune heavy chain repertoires. In contrast to the synthetic data analyses of the previous chapter, we do not know the ground truth of the allelic content of our immunoglobulin sequences, and thus our evaluations of the accuracy of our algorithms through biological data can only ever be approximations.

In this chapter, the focus of our evaluations will be on cross-comparing the results of our inferences with three competing algorithms which rely on a pre-existing reference database of alleles: IgBlast, IMGT's V-QUEST, as well as our own in-house software Cloanalyst. Each of these tools currently uses an identical reference database of alleles[5] (the one produced by IMGT) and are popular methods for inferring human immune repertoires.[9,36,37]

We conclude this chapter with a review of the major successes of the overall project in light of the original project aims. We also discuss some of the significant obstacles and limitations of the project in its current state, highlighting the areas available for future work and some remaining open questions.

Sources and Pre-Processing of Biological Data

All of the human heavy chain immunoglobulin sequences used in these evaluations had been previously obtained as part of earlier work in our laboratory. While

we include a brief overview of the methods here for the sake of clarity and introduction, readers interested in a more detailed methodology are referred to the original paper.[38,39]

Anthrax Vaccine Adsorbed (AVA) Trial

Six human subjects had been previously enrolled as part of a study regarding the response of the human immunoglobulin repertoire to the anthrax vaccine. During the trial, subjects were injected with a series of up to six vaccinations, and their blood was drawn at time of injection and one week post-vaccination, in order to capture the peak adaptive immune response. Plasmablasts were isolated from the blood via flow cytometry, and their immunoglobulin-specific mRNA was extracted and sequenced.

Commercial Computational Processing of Sequenced Immunoglobulin Reads

The sequenced reads collected from these plasmablasts underwent some additional computational processing steps in order to satisfy quality control and formatting requirements for our novel statistical methods. These processing steps included the joining of the matching paired-end reads to form contiguous sequences of roughly 300bp in length, the removal of adapter sequences from the ends of reads introduced during standard library preparation, and a filtering step to eliminate reads which were of poor quality. These computational pre-processing steps were performed as part of the external commercial sequencing service Atreca.

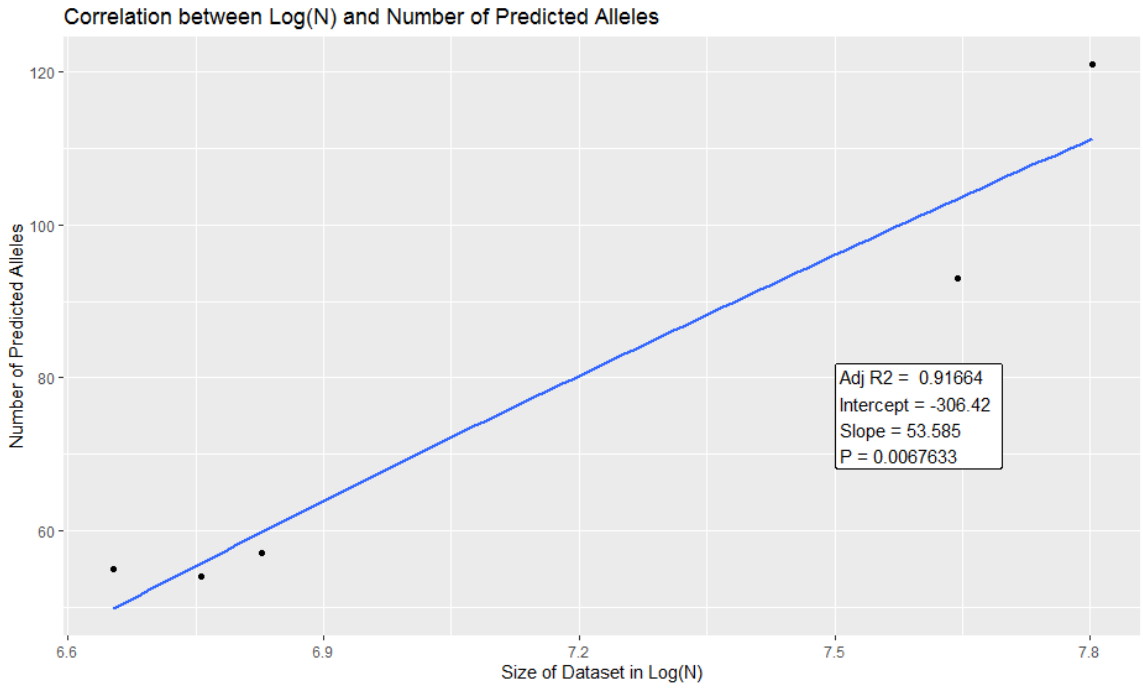
Subsampling of Biological Data Sources

To evaluate the algorithm in its present state, we required only the subset of our available data which encoded for immunoglobulin heavy chains. In the event our

algorithm is updated to process light chain rearrangements, then the cross-comparisons analysis detailed below can also be extended using information obtained from the same source.

The total number of immunoglobulin heavy chain sequences varied substantially between subjects, with the average read count of 1746 and a standard deviation of 1067. Table 3 below lists the total reads (N) available for each subject. From our synthetic trials, we had learned that N was positively correlated with the total number of predicted alleles in the final library of candidate germline gene segments, with larger datasets tending to include more spurious alleles. This trend was corroborated in our initial assessments of the biological data, as shown in Figure 23. Note that the x-axis is a log transformation of N.

Figure 23: Correlation Plot of Log(N) and # of Predicted Alleles



To control for this effect, we decided to randomly subsample our data into sets of 750 sequences. We took these 750-sequence random subsets in triplicate for each of our subjects, in the hopes of recapturing some of the original biological variability. In addition to the total size of the dataset for each subject, Table 3 also lists both the proportion of sequences which were covered in one replicate of size 750 sequences, and in the union of triplicate sets for each biological subject.

Table 3: Summary of Biological Triplicate Datasets

SUBJECT ID	TOTAL SEQS (N)	SAMPLE SIZE (K) (%)	UNION OF TRIPLICATES (%)
S1H	2090	750 (36%)	1504 (72%)
S2H	923	750 (81%)	919 (99%)
S3H	859	750 (87%)	858 (~100%)
S4H	775	750 (97%)	775 (100%)
S5H	3383	750 (19%)	1790 (53%)
S6H	2445	750 (31%)	1619 (66%)

Comparisons with Cloanalyst, IgBlast & IMGT V-QUEST

We compared our autonomous allelic inference approach with three different reference-database reliant methods: third party softwares IgBlast and High-VQUEST, and in-house software suite Cloanalyst. These platforms were selected for their relative popularity, ease of use, and consistency in their selection of a common allelic reference database, namely the one provided by IMGT. We ran each of the eighteen datasets listed in Table 4 under each platform using their respective default parameters. For our

Dirichlet process approach, we set our $log\ (\alpha) = 300$ and kept the same β parameter annealing schedule as in our synthetic trials.

Allele Quantity Comparisons

Table 4 below compares our inferred library size to that of the alternate platforms for each of the triplicate datasets. As discussed in a later section, a subset of reads resulted in a multiplicity of potential gene segment assignments, which complicates the counting of unique gene names. The datasets labeled "All" refers to the inclusion of these alternate gene names in the total unique gene name count. For High VQUEST, the "single match" column refers to the number of unique gene names when ambiguous gene names are excluded. For IgBlast, the 'Top Allele" column refers to the count when only the first of three top matching gene names are included in the total unique gene name count. In general, we find that our inferred libraries contain fewer alleles than any of the alternate platforms, coming closest in performance to the 'single match only' VQUEST libraries, and being dwarfed by both HighVQUEST (all) and IgBlast (all) at ratios of roughly 2:1 and 3:1 respectively.

Table 4: Summary of Predicted Allele Comparisons

Dirichlet	Dirichlet Inferred Alleles	Cloanalyst	High VQUEST (All)	High VQUEST (Single match)	IgBlast (All)	IgBlast (Top Allele)
S1H – A	54	72	87	55	151	64
S1H – B	51	72	89	57	141	64
S1H – C	49	65	86	50	144	58
S1H Avg.	**51.3**	**69.7**	**87.3**	**54.0**	**145.3**	**62.0**
S2H – A	49	76	101	58	143	67
S2H – B	49	76	99	54	140	66
S2H – C	51	76	100	56	146	67
S2H Avg.	**49.7**	**76.0**	**100.0**	**56.0**	**143.0**	**66.7**
S3H – A	54	84	100	55	161	68
S3H – B	51	86	100	56	159	69
S3H – C	52	86	100	56	161	69
S3H Avg.	**52.3**	**85.3**	**100.0**	**55.7**	**160.3**	**68.7**
S4H – A	54	75	101	59	156	68
S4H – B	55	75	103	60	157	69
S4H – C	56	75	100	59	155	67
S4H Avg.	**55.0**	**75.0**	**101.3**	**59.3**	**156.0**	**68.0**
S5H – A	52	84	115	62	161	74
S5H – B	49	78	108	58	154	70
S5H – C	55	77	97	51	151	69
S5H Avg.	**52.0**	**79.7**	**106.7**	**57.0**	**155.3**	**71.0**
S6H – A	55	88	119	66	160	82
S6H – B	58	90	114	66	164	82
S6H – C	52	94	122	64	168	84
S6H Avg.	**55.0**	**90.7**	**118.3**	**65.3**	**164.0**	**82.7**
Total Avg.	**52.6**	**79.4**	**102.3**	**57.9**	**154.0**	**69.8**

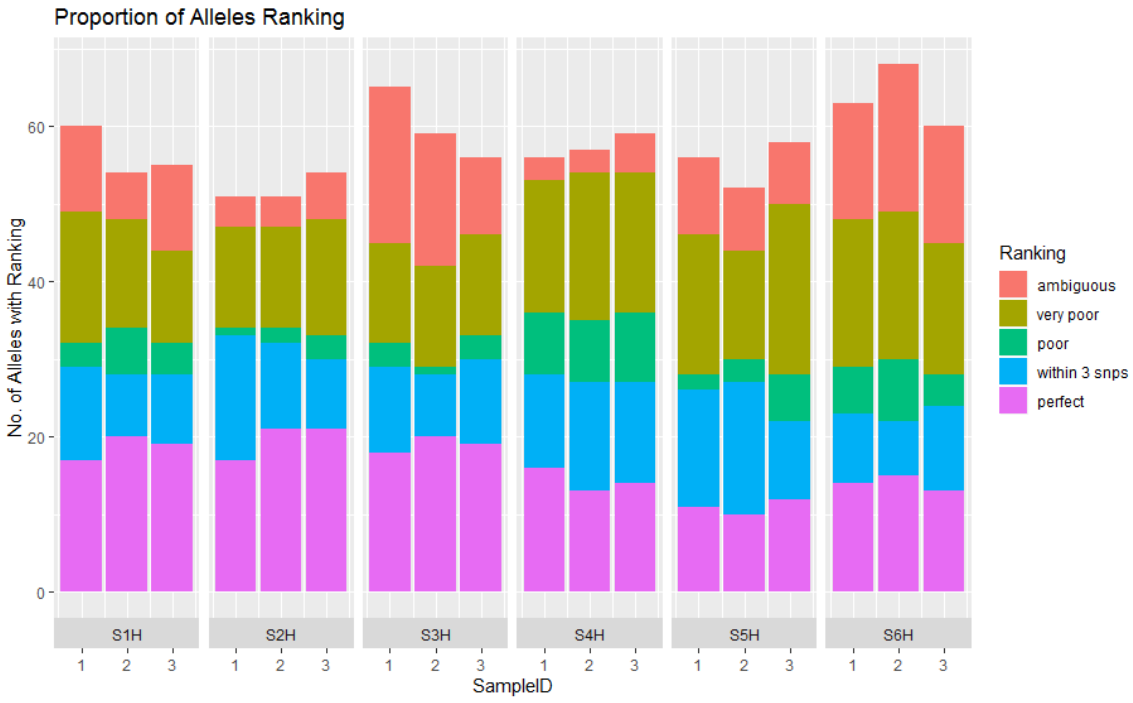
Gene Name Annotation of Inferred Allele Libraries for Cross-Platform Comparisons

Since our algorithm infers alleles autonomously from the input data, there is no pre-designated gene name annotation available. For the purposes of comparing gene segment assignments from our inferred allele libraries versus those assignments made using reference database-reliant approaches, we needed a means of labeling our inferred alleles in context with the other platforms. Therefore, we interpolated a gene name annotation of our inferred alleles by systematically cross-matching them with the IMGT reference allele library shared in common by Cloanalyst, IgBlast and VQUEST for each of our eighteen datasets. This consisted of an iterative many-to-many matching operation where pairwise sequence alignments between alleles from our inferred library and query alleles of the IMGT reference library. The IMGT gene name corresponding to the query allele which conferred the fewest mismatches to a given candidate (and whose pairwise alignment also contained zero inner gaps), was the one selected for annotation. We emphasize that this annotation interpolation is not included during the allelic inference phase unique to our platform, but is only included to provide external context for these third-party comparisons.

We then applied a crude ranking method for these interpolated gene name annotations based on the minimum mismatch value: perfect, close, poor, and very poor. Annotations were only given a ‘perfect’ ranking if the aligned sequence pair contained 0 SNPs. Matches which contained 1-3 SNPs were labeled as ‘close’. ‘Poor’ matches were those in the range of 4-10 SNPs, and ‘very poor’ matches with >10 SNPs. Alignments which contained gaps were excluded from our consideration for allele annotation, and

were thus generally left unranked. Figure 24 demonstrates the proportion of our annotations which fell into these five categories for each of our datasets.

Figure 24: Allele Ranking Proportions Across All Samples



Sometimes, there would be a tie for the annotated minimum mismatch gene name. Ties occurred for each of the four annotation rankings including ‘perfect’ matches. Ties in the case of ‘perfect’ matches indicate duplicate alleles found within the IMGT reference database; i.e. identical sequences under alternative gene names. In this case, ties were resolved by selecting the one of these gene names, and filtering out the duplicate alternatives.

For annotations with a non-perfect ranking that also resulted in a tie for best match (roughly 8.7% of sequences for all datasets), gene names were secondarily ranked

to minimize inner gaps, and if a tie still remained, then the alphabetically first gene name in the tied set was selected for annotation and were given a ranking of ‘ambiguous’.

Complexity of Gene Annotation for Cross-Platform Comparisons

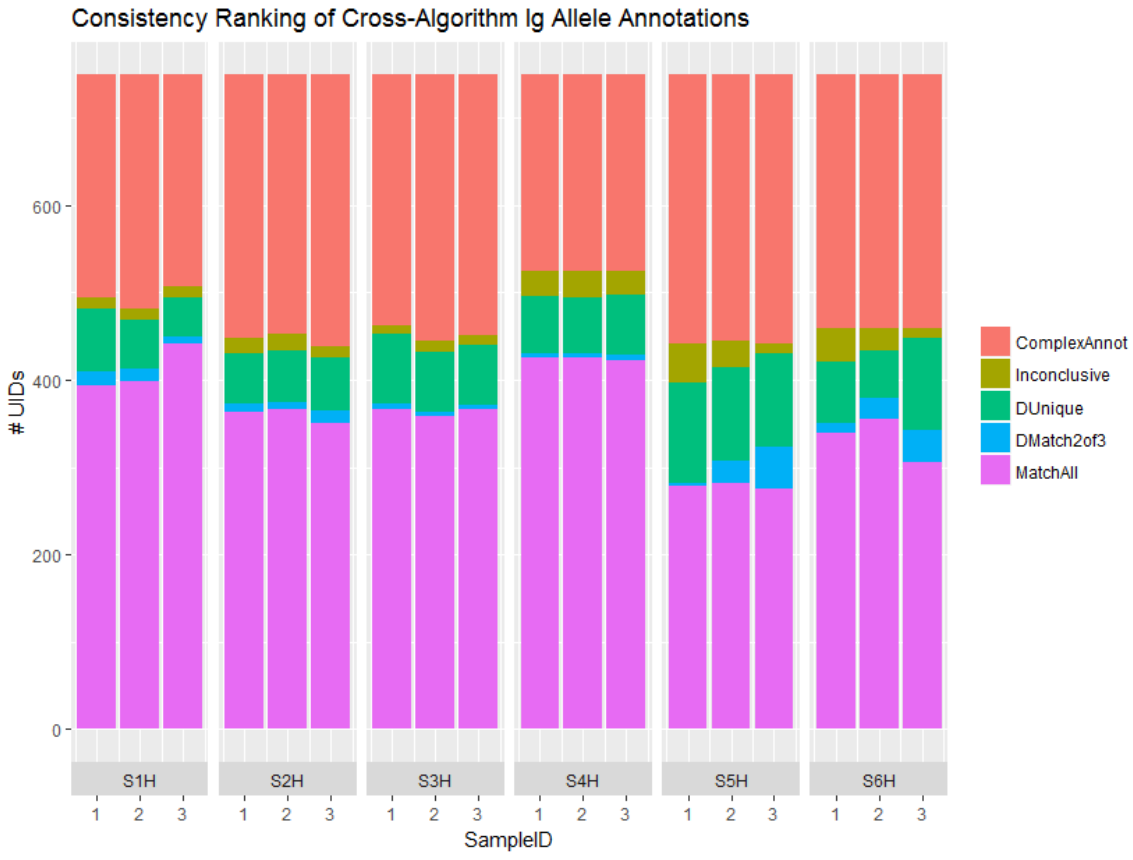
Ambiguous gene name assignments were also present in the results from both IgBlast and High VQUEST. By default, IgBlast reports the top three most significant gene segment candidates for each input immunoglobulin read. For High-VQUEST, only a minority of reads had ambiguity in regards to their gene assignment, but the size of the candidate gene set was highly variable. In one notable instance, VQUEST reported seven candidate gene segment annotations for the read being queried.

Given the inherent complexity of defining a consistency metric for overlapping gene name sets of variable size across software platforms, we opted for a more simplified approach. We limited our platform comparisons search space to only consider reads which had an unambiguous annotation for each of the software platforms. In particular, this excluded reads which resulted in non-perfect ties from our interpolated annotations or non-single annotations through HighVQuest. We also considered only one of the three IgBlast annotations for a given read. On average, these combined filters excluded approximately one third of reads in our datasets. Figure 25 below demonstrates the proportion of these excluded reads in orange. (‘ComplexAnnot’)

We then divided the remaining simply annotated reads into four populations of interest. The first group contained reads which were consistently annotated with the identical gene name across all four platforms (the ‘MatchAll’ group). The second group included reads where three out of four platforms agreed on a gene name, of which our

Dirichlet-clustering platform was included (the ‘DMatch2of3’ group). The third group of reads were those where agreement was shared amongst all three database-reliant platforms, but the interpolated annotation from the Dirichlet-clustering platform was inconsistent (the ‘DUnique’ group). The final group contained the small remaining fraction of reads where there was an inconsistent pattern of gene annotation across software platforms (the ‘Inconclusive’) group. Figure 25 below demonstrates the relative proportion of these groups across each of the eighteen datasets.

Figure 25: Consistency Ranking of Cross-Algorithm IG Allele Annotations

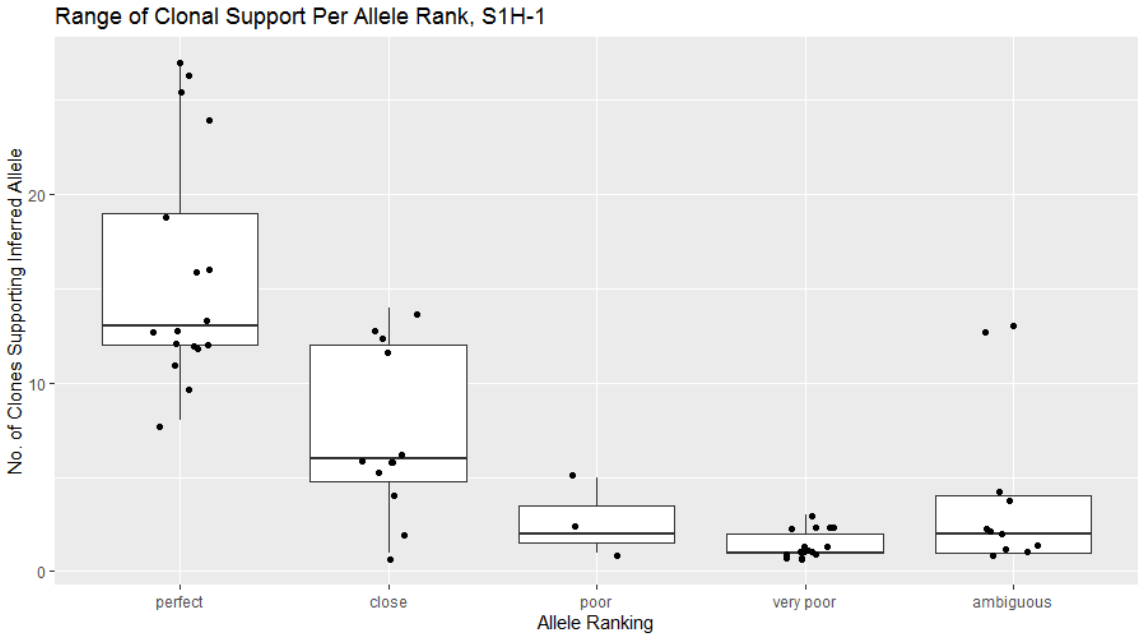


Investigation into Clonal Support of Low Ranking Alleles

Of the alleles in our inferred libraries, roughly half of them had a poor or worse ranking in terms of their best possible match with known IMGT reference alleles. The degree to which these alleles represent potential novel allele candidates vs. statistical clustering artifacts is unknown. However, we can measure our confidence in these inferred alleles by examining the level of clonal support which was used to infer them. Candidate alleles which exhibit a high degree of clonal support (i.e. alleles which are inferred from larger clusters of clones) are more likely to represent true allelic variation.

Figure 26 below contains a representative box plot of the data from one of our biological trials, S1H-1. The X-axis groups alleles according to the ranking of their best annotation match to IMGT reference alleles and the Y-axis shows the total number of clones that were used in the final inference of that allele. In general, we see a trend where the more poorly ranked alleles had less overall clonal support than their highly ranked counterparts. Box plots for the remaining 17 datasets included in the Appendix, however this pattern remains consistent across all eighteen datasets, with some variance of the ‘perfect’ and ‘close’ groups relative to each other.

Figure 26: Range of Clonal Support by Allele Ranking, S1H-1



Potential Novel Allele Detection

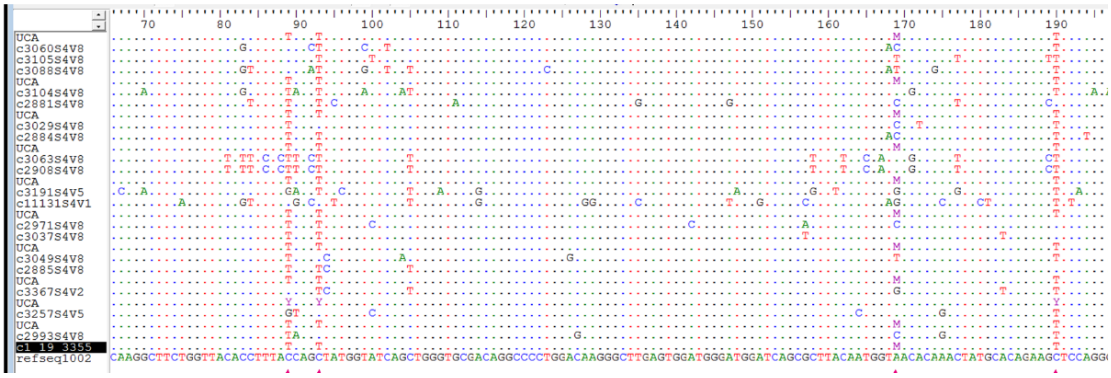
In addition to our previous analyses, we were also interested in whether any of the alleles in our predicted libraries could constitute the detection of a potential novel allele, previously uncharacterized in the IMGT V gene databases. While this particular line of investigation is still in its early stages, and would require further follow-up, we have identified 4 potential novel allele candidates from Subject 4 which have met all of the following criteria:

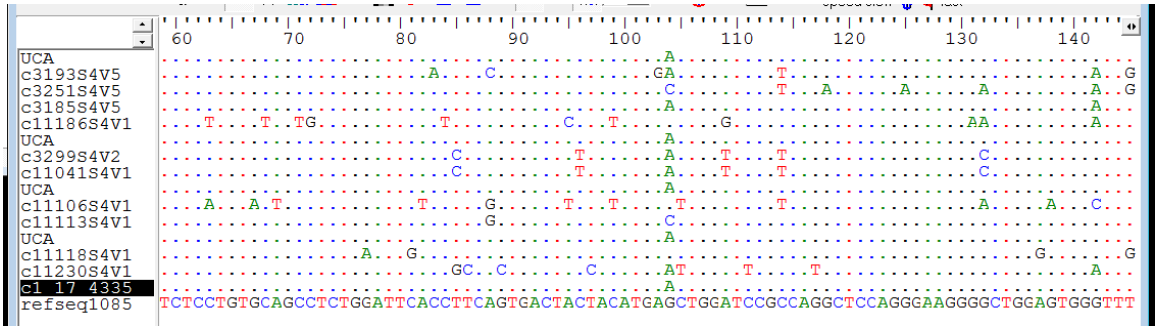
- The candidate allele contained at least one mismatch in the pairwise alignment comparison of their most similar IMGT reference allele.
- The candidate allele must have ≥ 3 clones with different V(D)J recombinations supporting the inference of the allele.

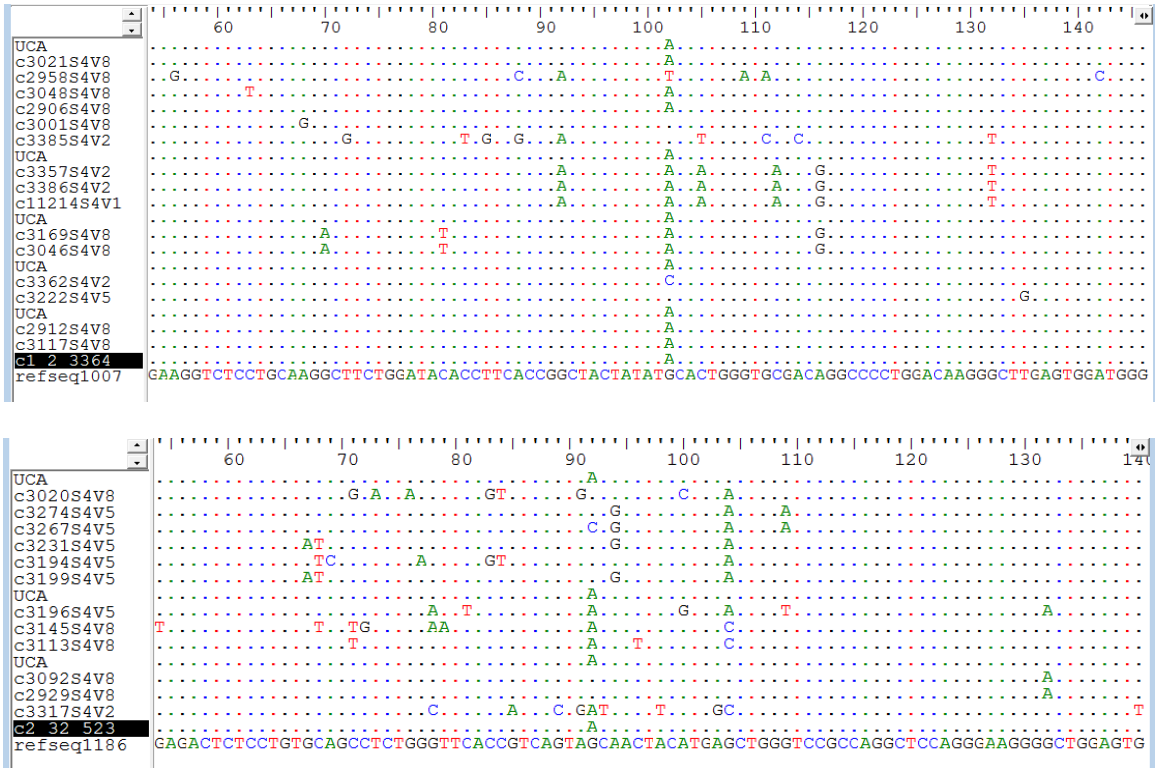
- These clones used for inferring the candidate allele must contain at least two sequences each. Singlet clones can add additional support to the inference of an allele, but they cannot stand on their own.

Figures 27a-d show alignments for each of these 4 novel allele candidates, their closest matching reference allele, and the non-singlet clones which used that allele candidate. In each diagram, the reference allele is used as a template; dots indicate positions where any other sequence matches the corresponding nucleotide in the reference template, while individual letters indicate where they differ from the template.

Figures 27a-d: Potential Novel Allele and Clones







For Figure 27a, there are four sites which differentiate our predicted allele from that of the reference template: positions 89 (C->T), 93 (C->T), 169(A->M), and 190(C->T). Of these changes, position 169 involves an ambiguous nucleotide encoding, so it can be discounted. Position 93 involves a silent mutation and located in a known mutation hotspot binding motif, and so likely does not represent true allelic variation. The other two sites represent candidates which should be explored further as they result in amino acid level changes; site 89 results in an amino acid change from leucine to isoleucine, while site 190 results in an amino acid change of a leucine to a phenylalanine. This latter site is particularly intriguing as it represents a significant change in the functional group of the amino acid, adding a ring structure to the end of the long hydrophobic chain.

For Figure 27b-d, there is only one site which differentiates the predicted allele from its reference template. In Figure 27b, the position at site 104 (G-> A) constitutes an amino acid switch from serine to asparagine; in Figure 27c the site 102 (G-> A) constitutes a switch from a methionine to an isoleucine, and in Figure 27d the position is at site 92 (G->A). Given the comparatively inconsistency of this SNP in the members of these three clones, this final sequence in Figure 27d a comparatively poor candidate for a novel allele.

There may be other candidates in one of the other five subjects available in this dataset. At time of writing, we could not pursue this avenue further due to project time constraints, but this remains an interesting opportunity for future research.

Summary & Conclusions

In this project, we have developed a robust set of statistical methods for performing autonomous immunoglobulin repertoire analysis. These methods operate underneath a cohesive paradigm of Bayesian statistical modelling via a clustering procedure derived from the Dirichlet Process. This paradigm allows for the inference of immunoglobulin germline gene libraries directly from high throughput repertoire sequencing data, independent of an external allelic database. These methods are further implemented in a series of machine learning algorithms which iteratively update the content of these libraries using information contained within the unique V(D)J gene segment assignments and clonal lineage derivations. We have demonstrated the capabilities of these methods on both synthetically generated data and actual biological human immunoglobulin repertoires.

Our extensive investigations with synthetic data have identified the limits of the clustering paradigm offered by the Dirichlet Process, in particular concentrating on the role of the prior parameter $log\ (\alpha)$ in the optimization of the final clustering state for our system. We find the impact of the 'rich get richer' property of Dirichlet Process clustering to have had particularly intriguing non-trivial effects on the dynamic behavior of our system, and warrants further research. Similarly, we believe that this method holds promise for the detection of additional novel allele candidates, pending further investigation.

APPENDIX I

Preliminary Empirical Trials for Annealing Schedule

This section will consist of a review of the early-stage experimental trials which were run to determine an appropriate simulated annealing schedule for the Gibbs sampler implementation of for immunoglobulin germline allele clustering.

The primary goal of these trials was to identify a range of acceptable values for the parameter β during the calculation of the log likelihood function discussed in Chapter 3. For these trials, we relied on an alternative set of synthetically generated data than the datasets discussed in Chapter 4, since at these trials were performed at an earlier stage in the algorithm's overall development.

Here, we review the characteristics of this alternative synthetic dataset, the parameters selected for the trials themselves, as well as the overall results.

Generation of 200 Sequence Synthetic Dataset

The V gene library used to generate this dataset was a subset of the human immunoglobulin V gene library; specifically 10 unique allele pairs were selected from the IGHV3 family of germline gene segments, for a total of 20 unique alleles in the starting V library. Each of these 20 V gene alleles was recombined with a unique DJ gene pair to create a founder sequence for a single clone (total of 20 clones). Each of these clones underwent a single round of division to produce two progeny sequences with SNP-distances of between 0-4 SNPs from the input founder sequence (total of 40 progeny sequences). By chance, 3 of the 20 clones applied 0 mutations to both of their progeny,

and so the child pair for these clones were identical (37 unique sequences out of total 40 progeny sequences).

All 40 progeny sequences were then manually given 5 identical replicates, for a total of 200 final sequences. These five intentional replicates were used as a positive control to test for technical errors associated with clustering. Identical sequences (i.e. perfect replicates) will ideally always cluster with themselves. If they do not, this would indicate that the clustering parameters are weighted too strongly towards generating new clusters. Since three of the twenty clones already had generated perfect replicates by chance, the most ideal clustering arrangement for this dataset of 200 sequences would be 34 'sets of 5' and 3 'sets of 10', with each set only containing only the replicates for a single gene segment.

$Log(\alpha)$ and β Parameter Preliminary Trials

Trials without Simulated Annealing (i.e. $\beta = 1$)

The first experimental trials were run while keeping β fixed at 1. This simplifies the process by removing the model component for simulated annealing, and allows $log\ (\alpha)$ to be tested in isolation. Table 5 shows the clustering results of a range of values for $log\ (\alpha)$ when β=1. For values of $log\ (\alpha) < 300$, all sequences in the dataset would be grouped together in a single cluster. For values of $log\ (\alpha) \geq 450$, every sequence was assigned into its own unique 'cluster'. For values of $300 \leq log\ (\alpha) < 450$, the clustering arrangement lay somewhere between these two extremes, with $log\ (\alpha) = 400$ coming the closest to the ideal clustering arrangement. In this range of values, we find that the clustering procedure is at best able to distinguish between different gene

segments of the IGHV3 family, but not between the members of a given allele pair. (e.g. IGHV3-21*01 & IGHV3-21*02 replicates would be erroneously clustered together, but IGHV2-23*01 and IGHV3-46*01 would not).

Table 5: Summary of Preliminary Trials, $\beta = 1$

$\log(\alpha)$	β	Cluster Sizes
0	1	1 cluster of size 200
150	1	1 cluster of size 200
200	1	1 cluster of size 200
250	1	1 cluster of size 200
300	1	5 clusters of size 20, 1 cluster of size 40, 1 cluster of size 60
350	1	3 clusters of size 10, 7 clusters of size 20, 1 cluster of size 30
400	1	20 clusters of size 10
450	1	200 clusters of size 1

None of the trials lacking simulated annealing experienced any sequence reassignment in successive rounds of the iterative Gibbs sampler; essentially the overall clustering state of the system was unchanged beyond the initial state, effectively 'crystalizing' from the first sequence assortment.

Trials with Simulated Annealing (i.e. $\beta \neq 1$)

Table 6 summarizes the β parameter trials which involved simulated annealing; in each trial, the parameter was given an initial starting value β_0 and updated in increasing increments after every three rounds of Gibbs sampler reassignment for a total of 50 rounds. Table 6 defines the starting β_0, the ending β_{50}, and the rule for increasing increments. Most trials use a simple additive or multiplicative for incrementing β. However, the last two trials also included a Fibonacci-series of increments, in an attempt to model 'rapid cooling behavior' in earlier rounds, and 'slower cooling behavior' in later

rounds. Table 6 also includes the corresponding values of $log\ (\alpha)$, and the clustering results themselves.

Table 6: Summary of Preliminary Trials, β ≠ 1

$\log(\alpha)$	β_0	β_{50}	Rule	Cluster Sizes
300	0.1	0.9	+0.1	1 cluster of size 4, 32 clusters of size 5, 1 cluster of size 6, 2 clusters of size 10
300	0.5	1.3	+0.1	34 clusters of size 5, 3 clusters of size 10
300	0.6	2.2	+0.2	34 clusters of size 5, 3 clusters of size 10
200	0.1	1.5	+0.1	32 clusters of size 5, 4 clusters of size 10
200	0.1	6.4	x2.0	1 cluster of size 1, 32 clusters of size 5, 1 cluster of size 9, 2 clusters of size 10
100	0.2	2.0	+0.2	32 clusters of size 5, 4 clusters of size 10
50	0.1	1.0	+0.1	22 clusters of size 5, 9 clusters of size 10
25	0.1	0.7	+0.1	1 cluster of size 29, 1 cluster of size 50, 1 cluster of size 121
25	0.025	0.875	+0.025	2 clusters of size 5, 19 clusters of size 10
15	0.025	0.425	+0.025	20 clusters of size 10
5	0.025	0.425	+0.025	1 cluster of size 30, 1 cluster of size 170
5	0.005	0.1	+0.005	4 clusters of size 10, 1 cluster of size 14, 3 clusters of size 20, 1 cluster of size 21, 1 cluster of size 25, 1 cluster of size 40
5	0.001	2.584	*	8 clusters of size 10, 3 clusters of size 20, 2 clusters of size 30
10	0.001	2.584	*	20 clusters of size 10

*Fibonacci-series of β's: 0.001, 0.002, 0.003, 0.005, 0.008, 0.013, …, 2.584

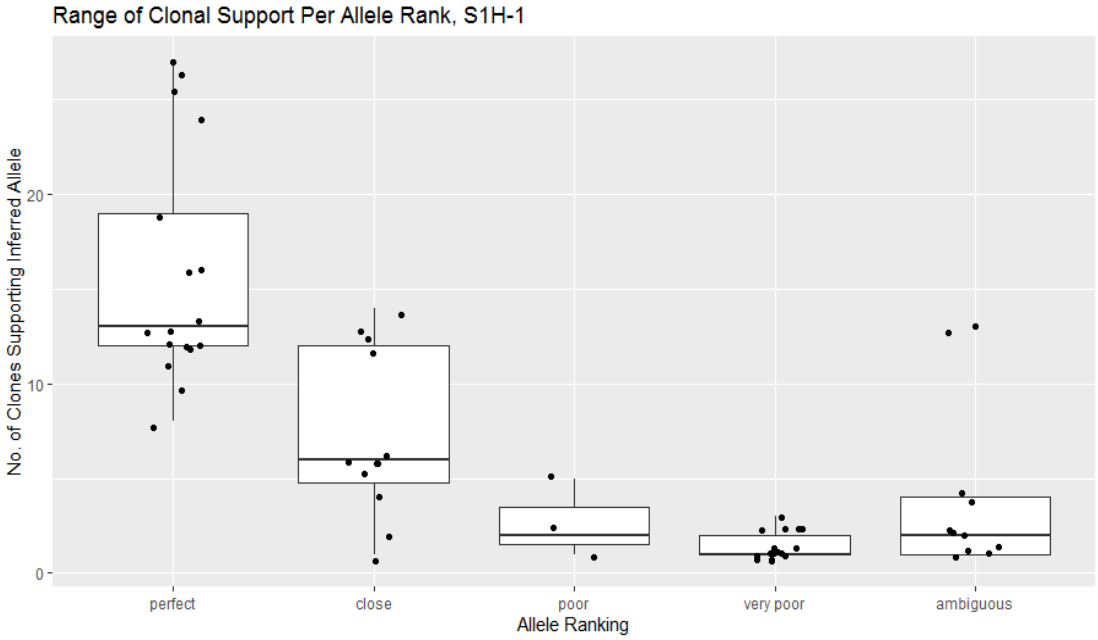
We were able to achieve our ideal clustering arrangement in the trials when $log\ (\alpha) = 300$, and β was incremented by 0.1 starting from $\beta_0 = 0.5$ up to $\beta_{50} = 0.9$. We were also able to achieve this arrangement with a higher increment of 0.2, but the starting value of $\beta_0 = 0.6$ to $\beta_{50} = 2.2$. This latter trial did not reach the ideal clustering

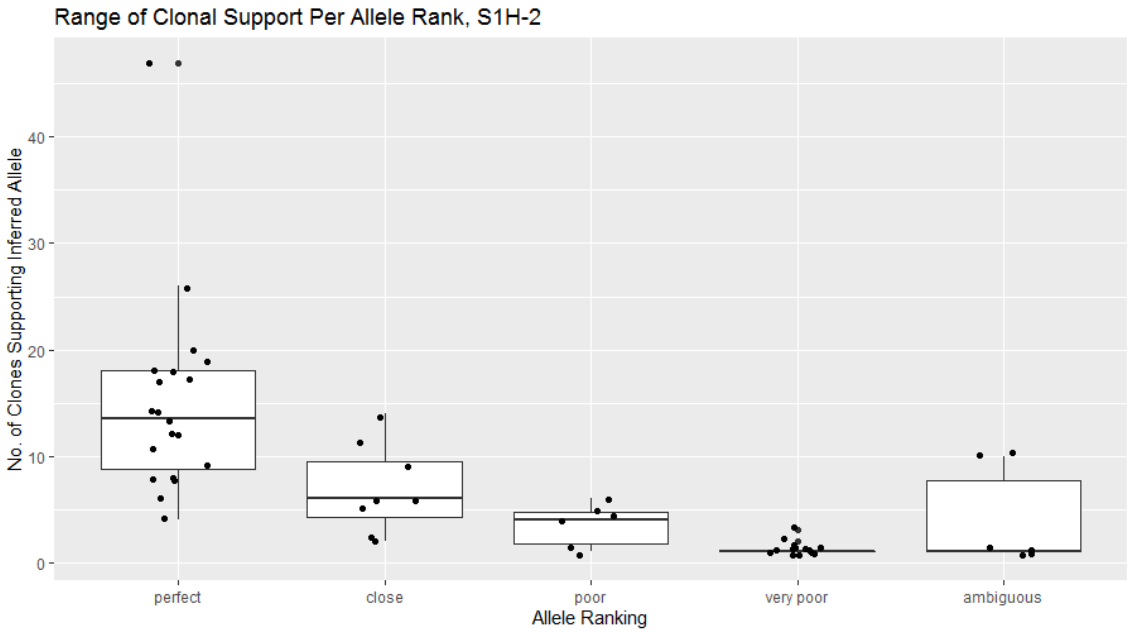
arrangement any sooner than the former trial, so the former was selected as the standard β parameter cooling scheme for future algorithm development and evaluation.

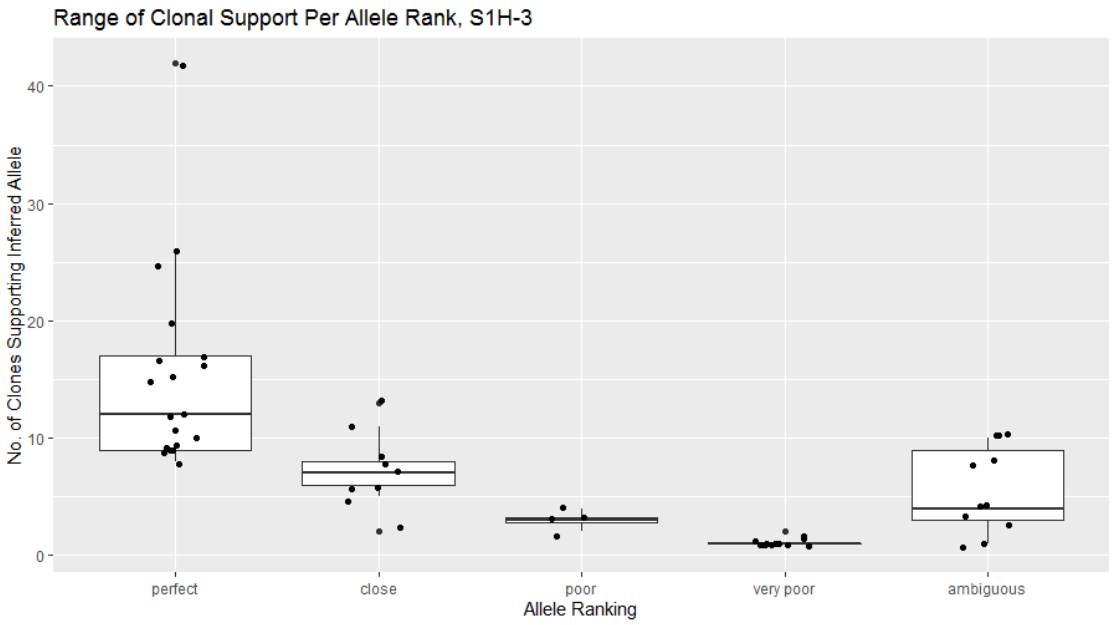
APPENDIX II

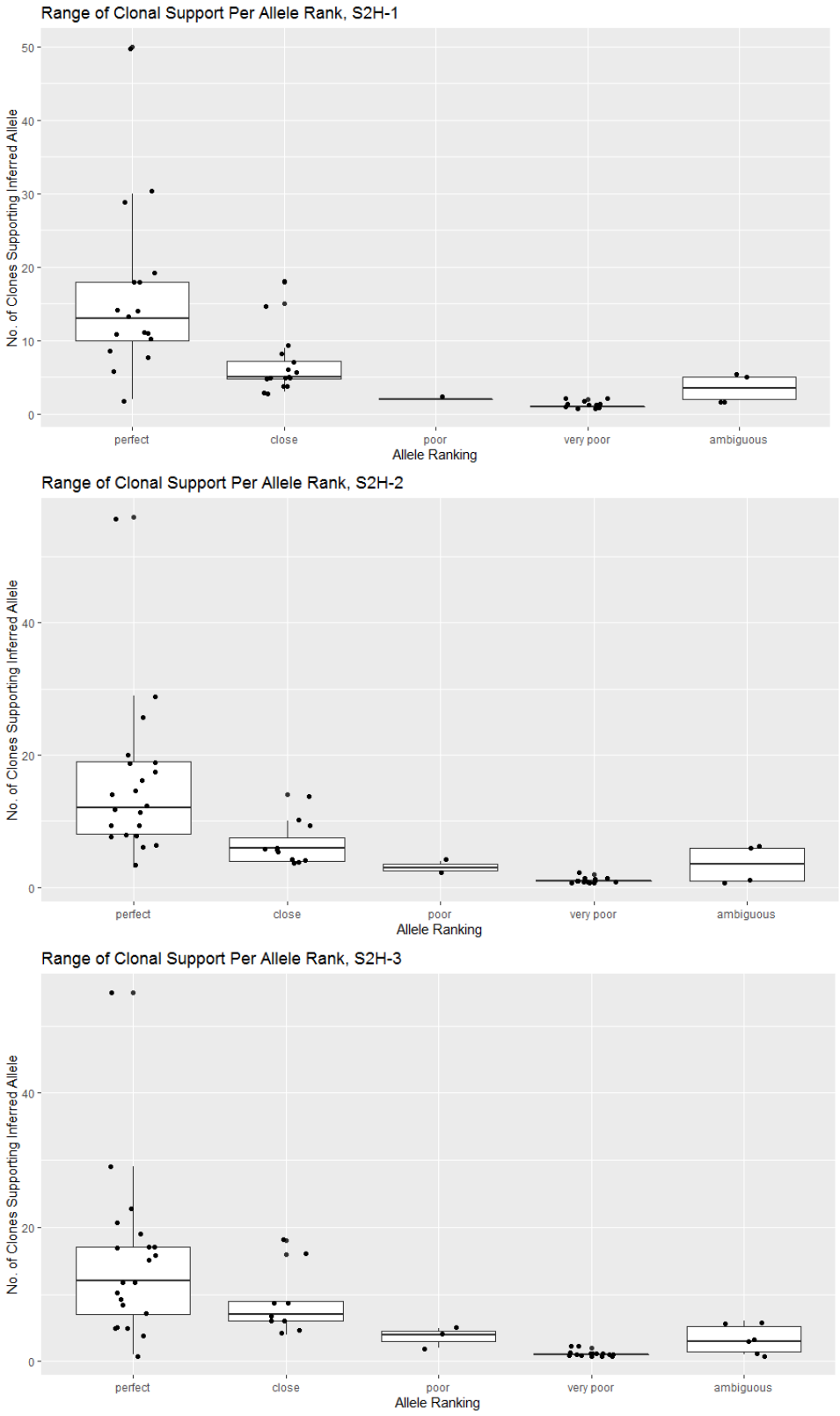
Additional Clonal Support Box Plots for S1H-S6H, All 3 Replicates

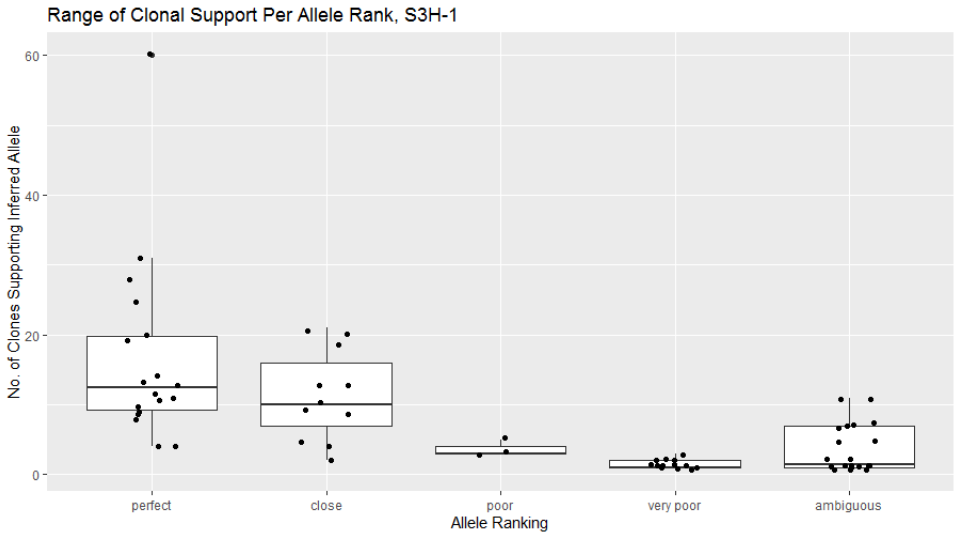
This section contains additional figures for the clonal support analyses completed in Chapter 5. Each box plot was obtained in an identical manner to Figure 26, but includes data from one of the corresponding eighteen biological triplicates.

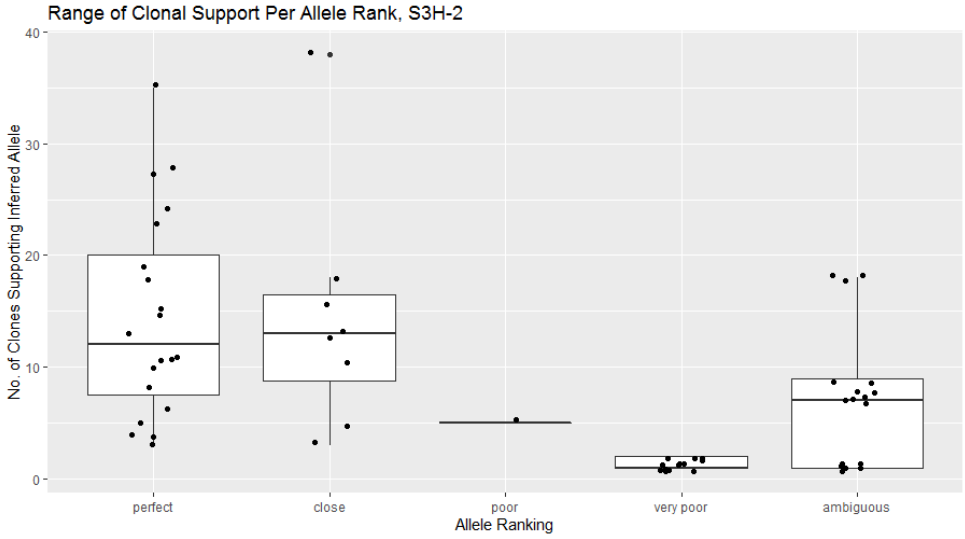


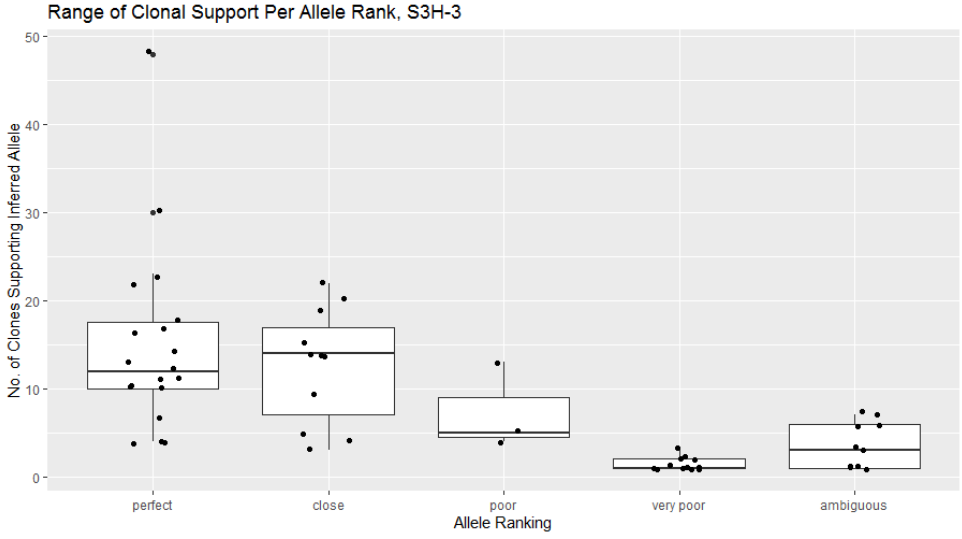

Range of Clonal Support Per Allele Rank, S1H-2
No. of Clones Supporting Inferred Allele
0
10
20
30
40
perfect
close
poor
very poor
ambiguous
Allele Ranking

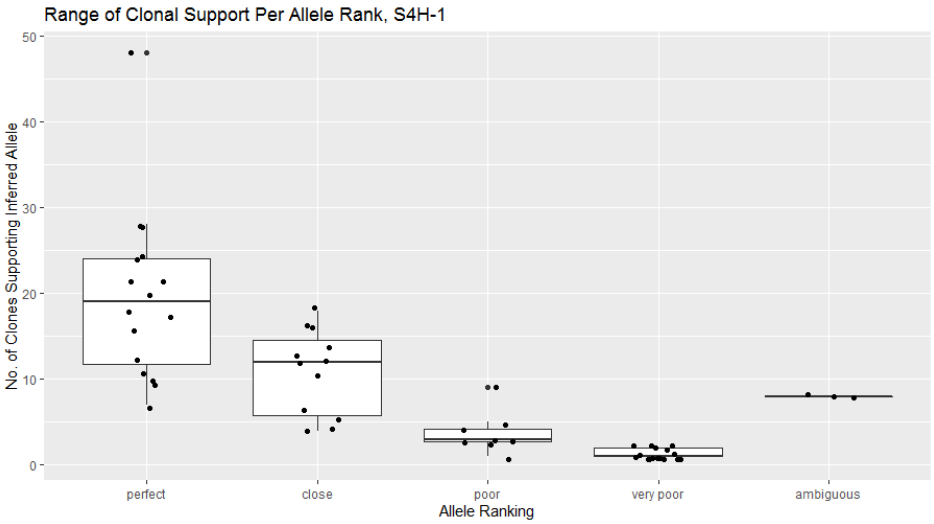

Range of Clonal Support Per Allele Rank, S1H-3
No. of Clones Supporting Inferred Allele
0
10
20
30
40
perfect
close
poor
very poor
ambiguous
Allele Ranking

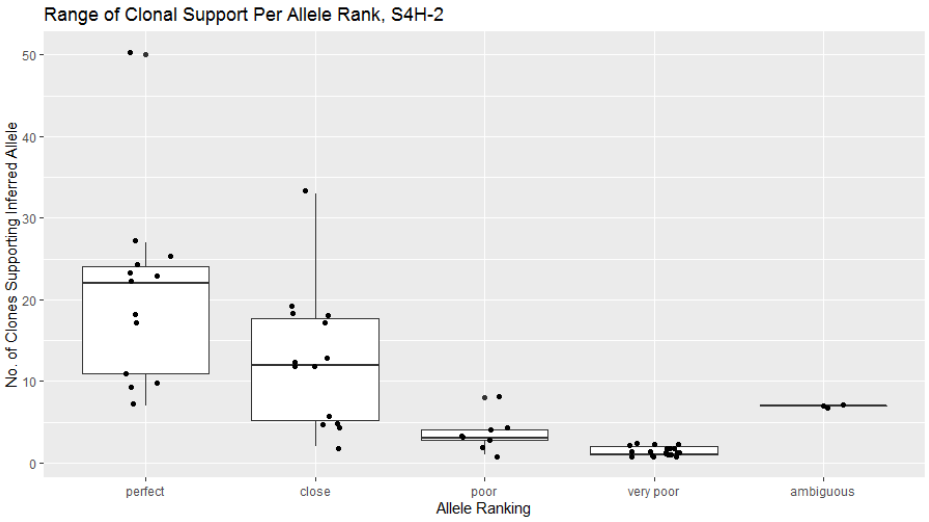

Range of Clonal Support Per Allele Rank, S2H-1
No. of Clones Supporting Inferred Allele
0
10
20
30
40
50
perfect
close
poor
very poor
ambiguous
Allele Ranking
Range of Clonal Support Per Allele Rank, S2H-2
No. of Clones Supporting Inferred Allele
0
20
40
perfect
close
poor
very poor
ambiguous
Allele Ranking
Range of Clonal Support Per Allele Rank, S2H-3
No. of Clones Supporting Inferred Allele
0
20
40
perfect
close
poor
very poor
ambiguous
Allele Ranking

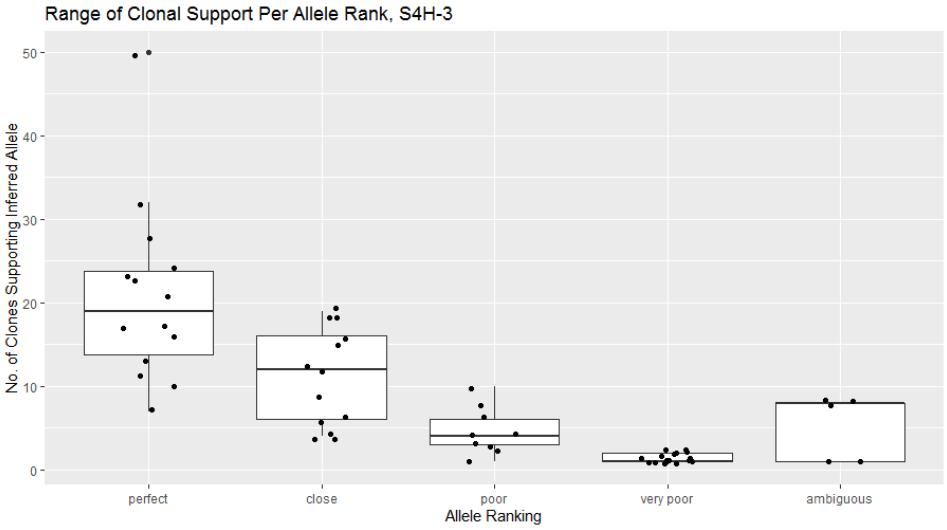

Range of Clonal Support Per Allele Rank, S3H-1
No. of Clones Supporting Inferred Allele
60
40
20
0
perfect
close
poor
very poor
ambiguous
Allele Ranking

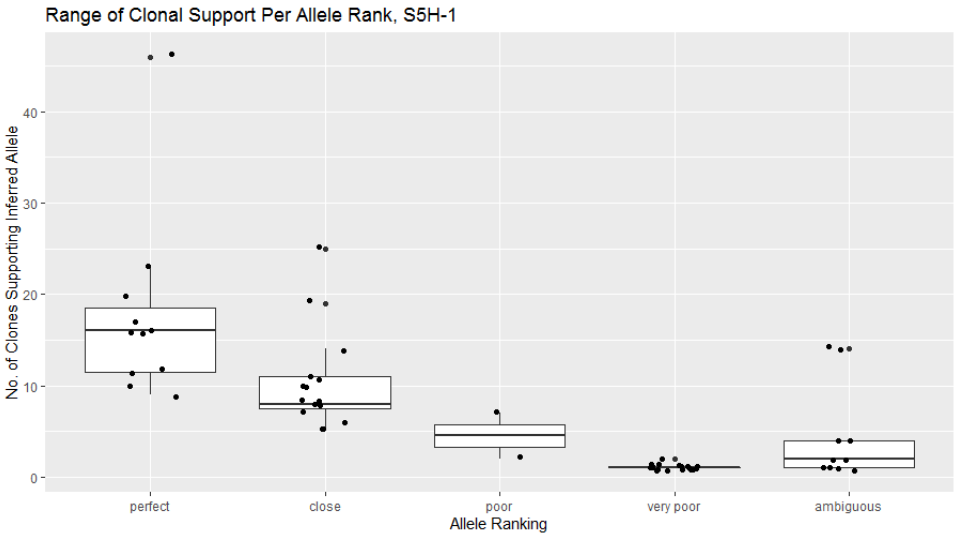

Range of Clonal Support Per Allele Rank, S3H-2
No. of Clones Supporting Inferred Allele
40
30
20
10
0
perfect
close
poor
very poor
ambiguous
Allele Ranking

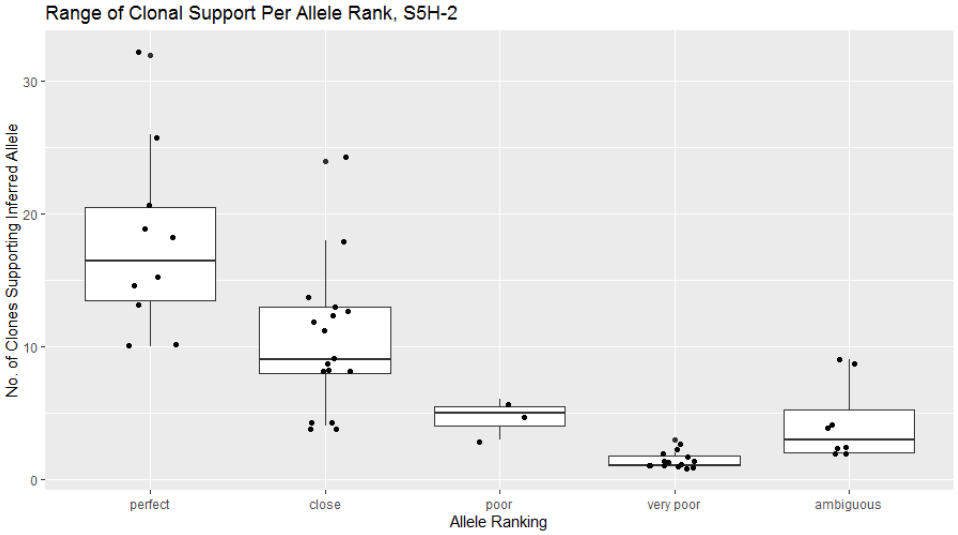

Range of Clonal Support Per Allele Rank, S3H-3
No. of Clones Supporting Inferred Allele
50
40
30
20
10
0
perfect
close
poor
very poor
ambiguous
Allele Ranking

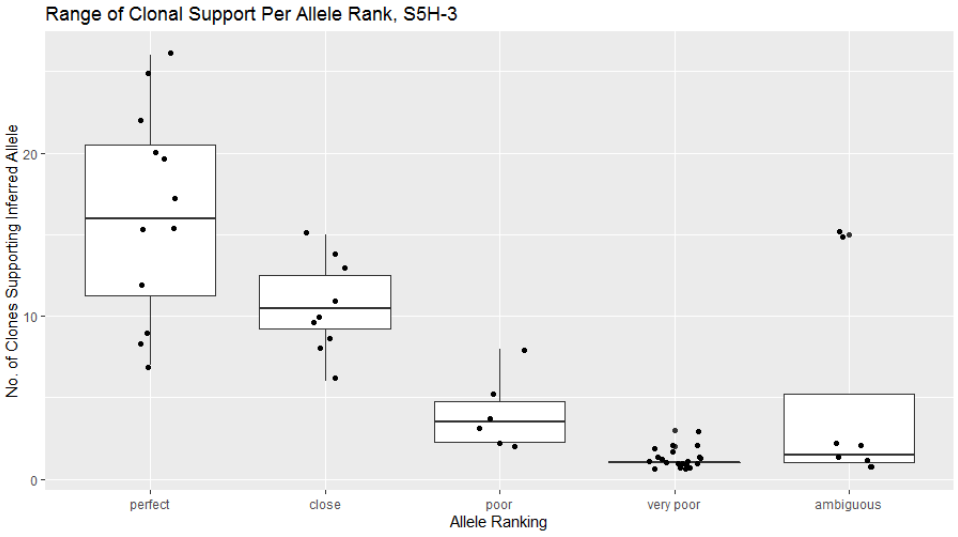

Range of Clonal Support Per Allele Rank, S4H-1
No. of Clones Supporting Inferred Allele
0
10
20
30
40
50
perfect
close
poor
very poor
ambiguous
Allele Ranking

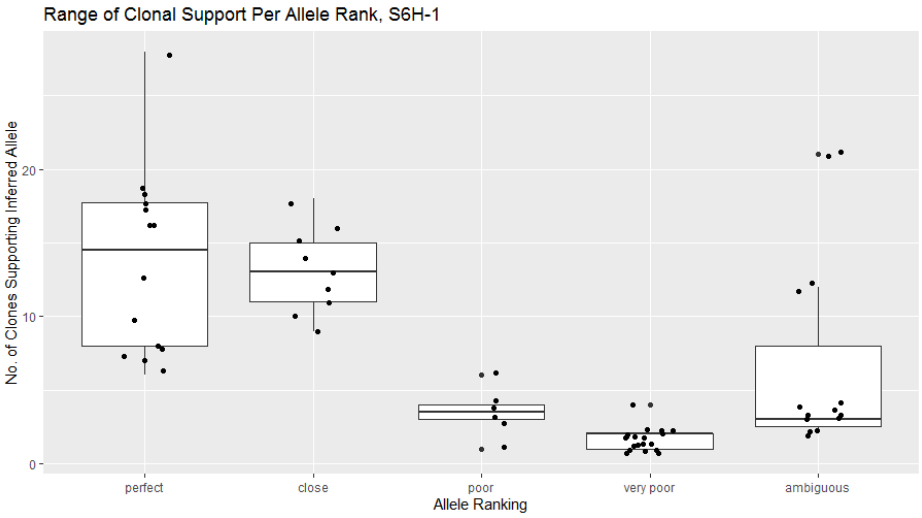

Range of Clonal Support Per Allele Rank, S4H-2
No. of Clones Supporting Inferred Allele
0
10
20
30
40
50
perfect
close
poor
very poor
ambiguous
Allele Ranking

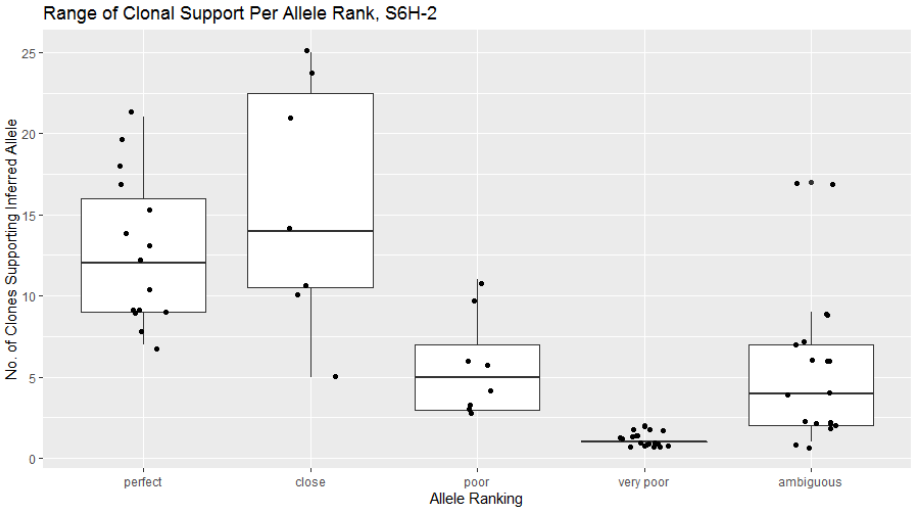

Range of Clonal Support Per Allele Rank, S4H-3
No. of Clones Supporting Inferred Allele
0
10
20
30
40
50
perfect
close
poor
very poor
ambiguous
Allele Ranking

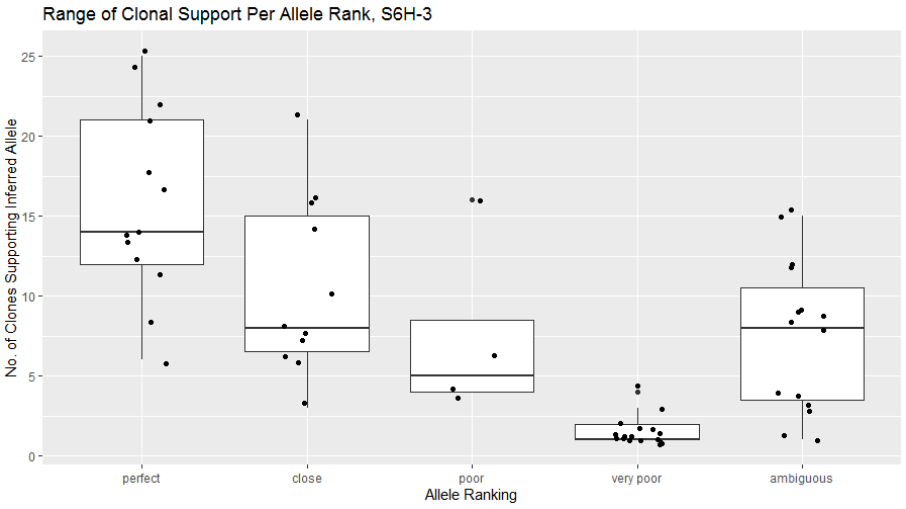

Range of Clonal Support Per Allele Rank, S5H-1
No. of Clones Supporting Inferred Allele
0
10
20
30
40
perfect
close
poor
very poor
ambiguous
Allele Ranking


Range of Clonal Support Per Allele Rank, S5H-2
No. of Clones Supporting Inferred Allele
0
10
20
30
perfect
close
poor
very poor
ambiguous
Allele Ranking


Range of Clonal Support Per Allele Rank, S5H-3
No. of Clones Supporting Inferred Allele
0
10
20
perfect
close
poor
very poor
ambiguous
Allele Ranking


Range of Clonal Support Per Allele Rank, S6H-1
No. of Clones Supporting Inferred Allele
0
10
20
perfect
close
poor
very poor
ambiguous
Allele Ranking


Range of Clonal Support Per Allele Rank, S6H-2
No. of Clones Supporting Inferred Allele
0
5
10
15
20
25
perfect
close
poor
very poor
ambiguous
Allele Ranking


Range of Clonal Support Per Allele Rank, S6H-3
No. of Clones Supporting Inferred Allele
0
5
10
15
20
25
perfect
close
poor
very poor
ambiguous
Allele Ranking

BIBLIOGRAPHY

1. Janeway, C. A., Travers, P., Walport, M. & Shlomchik, M. J. *Immunobiology*. (Garlic Science Publishing, 2005).

2. Benichou, J., Ben-Hamo, R., Louzoun, Y. & Efroni, S. Rep-Seq: Uncovering the immunological repertoire through next-generation sequencing. *Immunology* **135**, 183–191 (2012).

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

4. Sevy, A. M. & Meiler, J. Antibodies: Computer-Aided Prediction of Structure and Design of Function. *Microbiology Spectrum.* **2**, (2014).

5. Giudicelli, V. IMGT/LIGM-DB, the IMGT(R) comprehensive database of immunoglobulin and T cell receptor nucleotide sequences. *Nucleic Acids Research.* **34**, D781–D784 (2006).

6. Victora, G. D. & Nussenzweig, M. C. Germinal Centers. *Annual Review of Immunology.* **30**, 429–457 (2012).

7. Chaudhary, N. & Wesemann, D. R. Analyzing immunoglobulin repertoires. *Frontiers in Immunology* (2018). doi:10.3389/fimmu.2018.00462

8. Bonissone, S. R. & Pevzner, P. A. Immunoglobulin classification using the colored antibody graph. *Journal of Computional Biology.* **23**, 483–494 (2016).

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

10. Brochet, X., Lefranc, M. P. & Giudicelli, V. IMGT/V-QUEST: the highly customized and integrated system for IG and TR standardized V-J and V-D-J sequence analysis. *Nucleic Acids Research.* **36**, (2008).

11. Gadala-Maria, D., Yaari, G., Uduman, M. & Kleinstein, S. H. Automated analysis of high-throughput B-cell sequencing data reveals a high frequency of novel immunoglobulin V gene segment alleles. *Proceedings of the National Academy of Sciences of the United States of America.* **112**, 201417683 (2015).

12. Corcoran, M. M. *et al.* Production of individualized v gene databases reveals high levels of immunoglobulin genetic diversity. *Nature Communications.* **7**, (2016).

13. Ramesh, A. *et al.* Structure and diversity of the rhesus macaque immunoglobulin loci through multiple de novo genome assemblies. *Frontiers in Immunology.* **8**,

(2017).

14. Scheepers, C. *et al.* Ability To Develop Broadly Neutralizing HIV-1 Antibodies Is Not Restricted by the Germline Ig Gene Repertoire. *The Journal of Immunology.* **194**, 4371–4378 (2015).

15. Watson, C. T. *et al.* Complete haplotype sequence of the human immunoglobulin heavy-chain variable, diversity, and joining genes and characterization of allelic and copy-number variation. *The American Journal of Human Genetics.* **92**, 530–46 (2013).

16. Boyd, S. D. *et al.* Individual Variation in the Germline Ig Gene Repertoire Inferred from Variable Region Gene Rearrangements. *The Journal of Immunology.* **184**, 6986–6992 (2010).

17. Kidd, M. J. *et al.* The Inference of Phased Haplotypes for the Immunoglobulin H Chain V Region Gene Loci by Analysis of VDJ Gene Rearrangements. *The Journal of Immunology.* **188**, 1333–1340 (2012).

18. MacKay, D. J. C. Model Comparison and Occam's Raczor. in *Information Theory, Inference and Learning Algorithms* 343–353 (Cambridge University Press, 2003).

19. Kimura, M. A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution.* **16**, 111–120 (1980).

20. Jukes, T. A. & Cantor, C. R. Evolution of Protein Molecules. in *Mammalian Protein Metabolism* 21–132 (Academic Press, 1969). doi:https://doi.org/10.1016/B978-1-4832-3211-9.50009-7

21. Ferguson, T. S. A Bayesian Analysis of Some Nonparametric Problems. *Annals of Statistics.* **1**, 209–230 (1973).

22. Blackwell, D. & MacQueen, J. B. Ferguson Distributions Via Polya Urn Schemes. *Annals of Statistics.* **1**, 353–355 (1973).

23. Neal, R. M. Markov Chain Sampling Methods for Dirichlet Process Mixture Models. *Journal of Computational and Graphical Statistics.* **9**, 249–265 (2000).

24. Blei, D. M. & Jordan, M. I. Variational inference for Dirichlet process mixtures. *Bayesian Analysis.* **1**, 121–144 (2006).

25. Teh, Y. W. (Gatsby C. N. U. C. L. Dirichlet Processes: Tutorial and Practical Course (updated). 1–160 (2007). Available at: https://www.stats.ox.ac.uk/~teh/teaching/npbayes/mlss2007.pdf. (Accessed: 28th

February 2020)

26. Rasmussen, C. E. The Infinite Gaussian Mixture Model. in *Advances in Neural Information Processing Systems 12* (eds. Solla, S. A., Leen, T. K. & Muller, K.-R.) 554–560 (MIT Press, 1999).

27. Smolyakov, V. Bayesian Nonparametrics | Cube.js Blog. (2017). Available at: https://statsbot.co/blog/bayesian-nonparametrics/. (Accessed: 28th February 2020)

28. Rochford, A. Dirichlet process mixtures for density estimation — PyMC3 3.8 documentation. (2018). Available at: https://docs.pymc.io/notebooks/dp_mix.html. (Accessed: 28th February 2020)

29. Xing, E. P. 19: Bayesian Nonparametrics: Dirichlet Processes. *Lecture Notes* 1–8 (2014). Available at: moz-extension://b32a00c6-aef6-48a2-a57f-ecd3a1302327/enhanced-reader.html?openApp&pdf=https%3A%2F%2Fwww.cs.cmu.edu%2F~epxing%2FClass%2F10708-14%2Fscribe_notes%2Fscribe_note_lecture19.pdf. (Accessed: 28th February 2020)

30. Li, Y., Schofield, E. & Gönen, M. A tutorial on Dirichlet process mixture modeling. *Journal of Mathematical Psychology.* **91**, 128–144 (2019).

31. Gormley, M. The Dirichlet Process (DP) and DP Mixture Models. *Lecture Slides* 1–28 (2016). Available at: moz-extension://b32a00c6-aef6-48a2-a57f-ecd3a1302327/enhanced-reader.html?openApp&pdf=http%3A%2F%2Fwww.cs.cmu.edu%2F~epxing%2FClass%2F10708-16%2Fslide%2Flecture18-DP.pdf. (Accessed: 28th February 2020)

32. Volpe, J. M., Cowell, L. G. & Kepler, T. B. SoDA: Implementation of a 3D alignment algorithm for inference of antigen receptor recombinations. *Bioinformatics* **22**, 438–444 (2006).

33. Munshaw, S. & Kepler, T. B. SoDA2: A Hidden Markov Model approach for identification of immunoglobulin rearrangements. *Bioinformatics* **26**, 867–872 (2010).

34. MacKay, D. J. C. Simulated annealing. in *Information Theory, Inference and Learning Algorithms* 392 (Cambridge University Press, 2003).

35. Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. Combinatorial Minimization: Method of Simulated Annealing. in *Numerical Recipes: The Art of Scientific Computing* 326–334 (Cambridge University Press, 1987).

36. Giudicelli, V., Chaume, D. & Lefranc, M. P. IMGT/V-QUEST, an integrated software program for immunoglobulin and T cell receptor V-J and V-D-J rearrangement analysis. *Nucleic Acids Research.* **32**, (2004).

37. Greiff, V., Miho, E., Menzel, U. & Reddy, S. T. Bioinformatic and Statistical Analysis of Adaptive Immune Repertoires. *Trends in Immunology* **36**, (2015).

38. Sawatzki, K. Examination of the immunoglbulin repertoire before and after Anthrax Vaccine Adsorbed immunization. (Boston University School of Medicine, 2017).

39. Sawatzki, K. *et al.* Non-specific activation of autoreactive B cells after anthrax vaccination delays protection. *The Journal of Immunology.* **196**, (2016).

CURRICULUM VITAE