Statistical modeling for understanding affinity maturation of antibodies

by

Kristian Davidsen

Submitted to the Department of Bioinformatics
in partial fulfillment of the requirements for the degree of

Master of Science in Engineering

at the

Technical University of Denmark

July 2017

Author .....
Department of Bioinformatics
July 1, 2017

Certified by .....
Frederick A Matsen IV, Ph.D.
Associate Member, FHCRC Computational Biology Program
Thesis Supervisor

Certified by .....
Anders Gorm Pedersen, Ph.D.
Professor, DTU Bioinformatics
Thesis Supervisor

Statistical modeling for understanding affinity maturation of antibodies

by

Kristian Davidsen

Submitted to the Department of Bioinformatics
on July 1, 2017, in partial fulfillment of the
requirements for the degree of
Master of Science in Engineering

Abstract

Adaptive immunity is a highly active branch of biology dealing with issues that are relevant for understanding and treating a wide range of human diseases as well as basic understanding of animal physiology. B cell and their B cell receptors (BCRs) take a central role in the adaptive immune system e.g. in vaccine immunity caused by antibodies, the secreted form of BCRs. The adaptivity of BCRs stem from the darwinian evolution which they undergo to specifically neutralize foreign antigens, but only recently high throughput sequencing (HTS) has enabled to study this evolutionary process. An important objective of these HTS studies is to reconstruct the phylogeny of the evolutionary process, such that it is possible to understand the developmental path of immunity. Tools and theory are already well tested in the field of phylogeny, however most have been developed to study population genetics or evolution of organisms over millions of years, while BCR evolution occurs under very different conditions. We simulate BCR evolution under both a neutral model and a model we derive to represent realistic BCR sequence evolution, in which the fitness function is coupled to antigen affinity. We simulate data to recapitulate summary statistics of real BCR data, use a number of different phylogenetic methods to infer the simulated phylogenies and finally perform a validation using topological similarity and a novel metric we define to capture the correctness of an ancestral sequence reconstruction. Our results show that phylogenetic inference is robust to both simple and advanced simulations, and when sampling sequences at a single time point, sequence reconstruction is largely insensitive to the different methods tested. This indicates that inferring BCR phylogeny might not be as hard, regardless of its unconventional nature.

Thesis Supervisor: Frederick A Matsen IV, Ph.D.
Title: Associate Member, FHCRC Computational Biology Program

Thesis Supervisor: Anders Gorm Pedersen, Ph.D.
Title: Professor, DTU Bioinformatics

Acknowledgments

I would like to thank the whole Matsen group for hosting me at the Fred Hutch and showing me Seattle. Especially in the context of my work, thanks to the people I have been actively working with during the stay (in no particular order): **William DeWitt** for working on GCtree code, validation of ancestral state reconstruction and plotting. **Duncan Ralph** for help running, as well as adding new functionalities to partis. **Amrit Dhar** for working on amino acid substitution profiles B cell repertoire data. **Arman Bilge** for working on Bayesian phylogenetic pairing of heavy/light chain sequences using BEAST. **Andy Magee** for working on deeptime phylogenetics of IGHV genes. **Kenneth Hoehn** for help running IgPhyML and providing unpublished scripts for ancestral state reconstruction. **Chris Small and Laura Noges** for working on running clonal family phylogenetic analysis. **David Shaw and Jean Feng** for providing figures describing unpublished motif models of somatic hypermutation. **Vladimir Minin** for guidance and supervision in several projects. **Nikolaj Dietrich** for collaboration and enabling data sharing with Symphogen A/S. **Anders Gorm Pedersen and Ulrik Nicolai de Lichtenberg** for academic supervision and helping me winning the the Novo scholarship.

Lastly I am sincerely thankful to my supervisor at the Fred Hutch, **Frederick Matsen**. Erick has been very open to all my crazy ideas and included me in so much exciting work that I would not have been without.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Since the first vaccine research, by Edward Jenner on smallpox, it has been clear that the human immune system is adaptable and able to protect its host from serious infections. During the 20th century an impressive amount of basic research effort resulted in the deep understanding we have about immunology today. Many areas of immunology later turned out to be directly applicable in medicine as vaccines, anti-inflammatory agents, cancer treatment etc. Especially the area of adaptive immunity has been extensively studied due to its fascinating abilities to protect humans from diseases.

B and T cells and their receptors (BCRs and TCRs respectively) play a central role in the adaptive immune system which have classically been studied by low throughput techniques like microscopy, ELISA and PCR. However the most important part of these cells are their receptors and the binding abilities of these, which are purely sequence dependent. After this was recognized the first sequencing studies were slowly undertaken but with no where near enough sequences to give detailed insights about BCR and TCR function. It was first recently, around 2008, that high-throughput pyrosequencing (Roche 454) was used to sequence thousands of BCRs and TCRs [9], [7]. Since then high throughput sequencing is becoming a more standard lab technique and sequencing BCR/TCR repertoires (Rep-Seq) is now done by many academic groups and commercially by several companies.

Despite the initial enthusiasm around Rep-Seq, the dust has settled, revealing a slightly disappointing reality. Slightly disappointing because Rep-Seq is, for various reasons, not answering as many scientific questions as it was hoped. There has been a number of studies showing just how vast a diverse the immune receptor repertoire is [115], [21], and how there is a repertoire bias across different individuals [16], [100], but because BCR sequences themselves does not provide information about their function, interpretation of Rep-Seq results is difficult and too often ends up being a

hunt for significant case vs. control changes that has little meaningful interpretation. For example there might be a significant change in the use of certain germline genes, under circumstances of an auto-immune disease, but is this caused by the self reactive auto-antibodies or is it just a side effect of the disease's influence on the repertoire? Questions like this are difficult to answer without extra information about the function of the antibody present in the sample. However Rep-Seq is still in its infancy and indeed is starting to find a niche of applications e.g. lineage tracing for vaccine response [17], [108] and in antibody discovery applications [75].

Missing functional information for Rep-Seq data is what makes it difficult to interpret because there is no known truth to use as a reference. E.g. it cannot be known exactly which B cells are sharing the same common ancestor cell (clonal relationship), it can only be inferred from sequence similarity and epitope binding specificity. Most often epitope binding specificity it not available for each cell so clonal relationship is inferred solely based on sequence similarity with no test to validate this assumptions. In such cases when sequence information is rich, but functional and relational information is sparse, or non-existing, the importance of simulation studies cannot be understated. Simulating the missing information is often the only way to validate Rep-Seq analysis tools and therefore there is a real need for setting up simulation protocols mimicking real Rep-Seq data.

In this work I will present tools and methods for simulating the phylogeny of a B cell germinal center (GC) reaction, and then apply the simulated sequences to validate phylogenetic methods. Lastly I will present a method for integrating massive amounts of Rep-Seq data to make better amino acid substitutions in antibody engineering.

Chapter 2

Practical and theoretical considerations

Integrating high throughput sequencing data into immunology research is a challenge that requires insight in many aspects ranging from basic cellular pathways of the immune system to statistical modeling and advanced data analysis. Even when the objective is to build a statistical model there are many benefits to draw from a solid knowledge about the pathways governing the immune system, and likewise it will avoid many pitfalls to have practical experience with sequencing data, read processing, quality control etc.

In this chapter I will describe the theoretical fundamentals of the chapters to follow, mixed with some of the practical considerations that are important but somewhat implicit knowledge in the field.

2.1 Adaptive immunity

The function of the immune system is to protect its host against invading pathogens. To carry out this role the effector cells of the immune system have been equipped with a license to kill, and not only to kill pathogen, but also to kill its own host cells. With this potential harmful weapon the immune system needs to be tightly regulated to balance between fighting off pathogens while doing least possible harm on the host.

At the lowest level of classification the immune system is split up into innate and adaptive immunity. As the name suggested the innate immunity is hard coded in the genome and cannot change during the course of an infection. The opposite is true for the adaptive immunity, this is not hard coded in the genome and is undergoing development throughout the course of an infection. Innate immunity is the oldest of

the defense mechanisms, adaptive immunity came along later with a huge advantage to the host, but always on the terms of the innate immune system. The innate immune system is broadly defined as both the physical barriers, like skin, mucous, stomach acid etc. but also covers cells like macrophages, dendritic cells and the proteins of the complement system. The adaptive immune system includes less elements and is defined as the cells of the B and T cell lineages. There is a tight interaction between the two systems and events in the innate system can cause activation of elements in the adaptive, and vice versa, see figure 2-1.

To get a more complete picture of the processes involved lets walk through an example of a simple bacterial infection cause by a skin tear. First bacteria enters the blood stream where they proliferate, but some are also getting engulfed by macrophages. A macrophage will process the engulfed bacteria in small intracellular membrane vesicles (called lysosomes), first by killing and thereafter by chopping it up the content. The vesicles of chopped up bacteria will contain short peptides that was previously part of full-sized bacterial proteins, and these peptides are loaded into special surface receptors called the major histocompatibility complex class II (MHCII) and presented on the surface of macrophages. A large number of T cell with different TCRs are monitoring the blood stream and some will carry a TCR that binds specifically to an MHCII presenting a bacterial peptide on the surface of a macrophage. Upon binding the macrophage activates the T cell to undergo proliferation and a burst of T cells with identical TCRs will emerge (clonal burst). At the same time a large number of B cells with different BCRs are monitoring the blood stream binding a variety of different proteins. Whatever these B cells bind to their BCR is getting transported into the cell, chopped up and presented in MHCII is a similar fashion as the macrophages (this is a highly idealized description, for a detailed mechanistic review of antigen presentation to B cells see [5]). Some of these B cell will bind proteins from the immunizing pathogen and present peptides from these protein in MHCII on its cell surface. Now some of these MHCII:peptide complexes are identical to the ones presented on the macrophages and the clonally expanded T cells bind them. When a TCR binds an MHCII:peptide complex presented by a B cell this activates the B cell to undergo a number of steps, which will be discussed in details later in this section. The result is secretion of large quantities of antibodies (a secreted form of a BCR) that binds to the surface proteins of the pathogen and thereby signalling the complement system and other immune cells to clear the infection. The initial unspecific engulfing of extracellular fluids by macrophages will not be efficient enough to clear an infection, but once flagged with antibodies, clearance can be mediated by the complement system and occur rapidly.

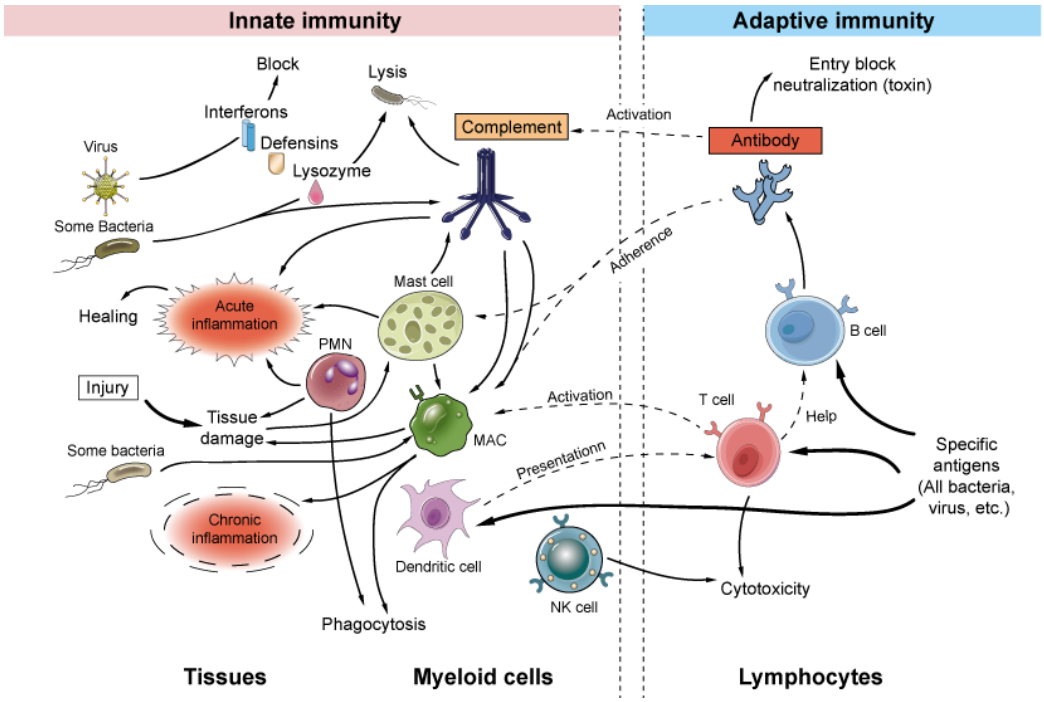


Figure 2-1: Connection between the innate and adaptive immune system. Macrophage (MAC), Polymorphonuclear leucocyte (PMN), Natural killer (NK). From http://www.creative-diagnostics.com/innate-and-adaptive-immunity.htm.

Summarized, a typical path to adaptive immunity goes through the following steps:

- An antigen presenting cell (macrophage or dendritic cell) presents a foreign peptide in its MHCII.
- A T cell clone with TCRs binding the MHCII:peptide complex gets clonally expanded.
- A B cell clone binds a foreign protein, engulfs it and presents its peptides in MHCII.
- The clonally expanded T cells bind MHCII:peptide complexes complementary to its TCR, inducing B cell proliferation and the production of antibodies.

Each item on the list has many details omitted for the sake of simplicity, but one important thing missing is the concept of memory. The above list ends with clearance of the infection through secretion of antibodies binding the pathogen, but the process is slow and energy consuming, so keeping memory of the response is an mechanism for rapid response to repeated infections. Memory is build by immortalizing T and B cells

involved in the adaptive immunity reaction of the first encounter with a pathogen, then whenever the same pathogen is encountered again, memory cells will quickly start clearing of the invasion.

2.1.1 B cell receptors

The adaptivity of adaptive immunity lies in the diversity of BCRs and TCRs. Like so many receptors involved in the immune system both BCRs and TCRs are built up by protein domains from the immunoglobulin superfamily. But unlike other receptors in the immune system they are not encoded by a single gene but by several genes stochastically recombined from multiple loci. Mechanisms of generation of BCRs and TCRs are so overlapping that it will be sufficient to only describe the mechanism of generating BCRs.

Antibodies are the secreted form of a BCRs, but the structure is the same, except for the missing membrane anchor. A BCR is a tetrameric protein of two identical heavy chains and two identical light chains, so called because one is longer than the other, see a) in figure 2-2. The heavy/light chain pairs are linked together covalently by disulfide bonds, and so are the two heavy chains from each pair, making a BCR a very stable structure. Both heavy and light chain starts with a variable region, defining their antigen binding capacity, and ends with a non-variable constant region (CH1-3/CL). The number of different variable regions found in a sample of naive (non antigen experienced) human B cells reveal that there are orders of magnitude too many different BCRs for them to be encoded as single genes on the genome. Instead the diversity is introduced by recombining several pieces of DNA from different loci into a full-sized variable region, see b) in figure 2-2. These are called the germline genes and for heavy chain sequences there are three groups, the variable (V), the diversifying (D) and the joining (J). These V, D and J germline gene groups have multiple variants existing in sequential order on the chromosome, and during a stochastic process called VDJ recombination, one of the genes from each germline gene group is physically spliced together into a full-size variable region VDJ gene. Coupled to a constant region gene further downstream this is being expressed as a single chain in the BCR protein.

While VDJ recombination is not a completely random process [55], it is the basis of all the variability in the variable region. But the combinatorics of all the germline genes is only modest, $\approx 39 \times 27 \times 6 = 6318$ different VDJ combinations. The rest of the observed diversity comes from another stochastic process involving trimming and inserting new nucleotides at the DNA ends that are joined together, see figure 2-3. The extra nucleotides added in the junction between V-D and D-J are called N/P

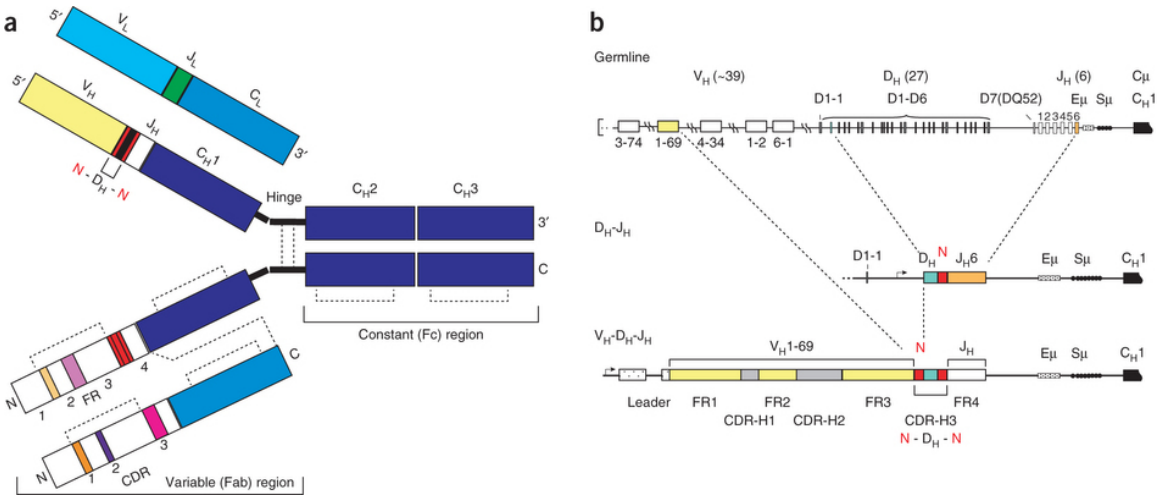


Figure 2-2: Antibody structure and germline organization on the genome. From [29].

nucleotides and the length of these insertions can vary from zero to tens of nucleotides. Although the length of the junctions are not uniformly random, and there is a bias in the inserted nucleotides [63], [21], the process adds orders of magnitude more diversity to the combinatorics of gene recombination. Furthermore all the junctional diversity is contained in the complementarity defining region (CDR), as opposed to the framework region (FR), see figure 2-3. As the name suggests the CDR is the structural region that binds the antigen, and to enable binding of a wide range of structures this must contain a lot of diversity. Based on a combination of gene recombination and junction diversity, and weighted by their biases, Elhanati et al. estimated that the naive repertoire is $10^{16} - 10^{18}$ productive sequences [21]. The resulting sequence of VDJ genes and N/P bases are referred to as the naive sequence to reflect that this is the stage before antigen experience and germinal center maturation explained later. All this diversification occurs within all individuals in the reactions of the adaptive immune system, but with the subtle difference that VDJ genes can differ between individuals. The different VDJ genes are called alleles and have the potential of expanding the binding breadth of the population wide BCR repertoire even further.

While the above description covers the concepts of BCR diversification in general it was applied specifically to an example using a heavy chain variable sequence. The light chain variable sequence is also made by recombination, but light chains have no D germline gene and therefore performs a VJ recombination with substantially less junctional diversity. Light chain germline genes are present in two non-identical copies named kappa and lambda residing on different chromosomes. The light chain sequence will only be expressed from one of these, the choice of which is determined

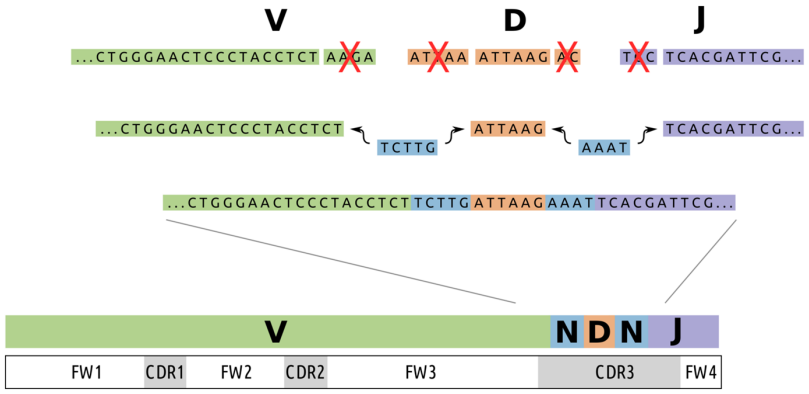


Figure 2-3: VDJ recombination and introduction of junctional diversity by exonuclease activity and N/P nucleotides added to the joining ends. From [72].

by a selection process in the bone marrow. Pairing of a heavy and light chain sequence yields an even larger possible naive repertoire, and with a production rate of at least 10^7 naive B cells per day [62], it is not surprising that the adaptive immune system can recognize nearly all foreign antigens. With such a broad range of antigen binding it also becomes possible to bind endogenous proteins. This is not desirable because it would lead to immune attach against host cells, also known as auto-immunity, therefore the adaptive immune system has devised as selection process in the bone marrow to destroy B cells with non-functional or self antigen binding BCRs. A similar process is happening for T cells in the thymus.

Germline nomenclature

The standard nomenclature for BCR genes used throughout this work is defined by the IMGT ontology summarized in figure 2-4 and described by Lefranc [53]. The first part of the name is always "IG" then followed by a locus identifier that can be either H (for heavy chain), K or L (for kappa or lambda light chain). Next letter represents the gene group which can be V, D or J for the variable region, and for the constant region the different isotypes e.g. A, E, G1, G2 etc.

For the variable region gene groups more layers of nomenclature is needed since there are many different V, D and J genes under each group. The first layer is the subgroup level which is defined as a cluster of genes sharing at least 75% nucleotide similarity. There are 7 of these subgroups and presumably they each derive from distinct gene duplication events, an interpretation supported by the distances separating subgroup genes on their inferred phylogeny, see figure 2-5. However the subgroup level is *not* defined phylogenetically, even though it might coincide with a phylogenetic interpretation.

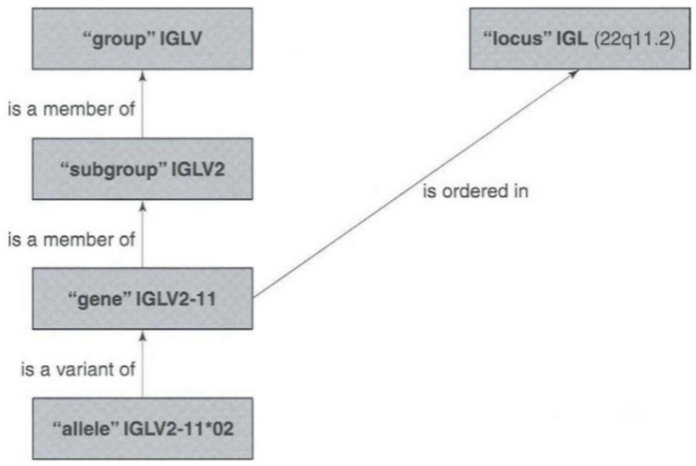


Figure 2-4: IMGT germline gene ontology. From [31].

Each subgroup has a number of genes that are defined by their chromosomal position (locus) and present in all individuals in a species. Each gene is encoded by a single sequence, the allele, that can vary between individuals. A gene can have many different observed alleles in a population, and while these allelic differences between individuals are usually just single nucleotide polymorphism, indels do occur causing some alleles of the same gene to have different length.

Antibody numbering

For both B and T cell receptors the huge diversity of the variable region is obvious on the sequence level, but much less so on the protein structural level. Actually even the variable region of a highly mutated BCR is strictly following the structural constraints of the beta-sheet sandwich structure of the immunoglobulin domain. Structural conservation enables consistent mapping from the amino acid sequence onto a protein structure. A mapping like this was first proposed as a numbering scheme where the immunoglobulin structural components are numbered and these numbers are mapped back to the linear amino acid sequence. First attempt on this was described by Kabat [66] and based on invariant residues and rule based matching (these rules are tabulated by Dr. Andrew Martins http://www.bioinf.org.uk/abs/). Unfortunately this scheme not good at handling insertion in the flexible CDR regions, e.g. CDR1 insertions and/or very long CDR3 [58]. Years later came another numbering scheme named after the first author of Clothia et al. [12]. The Clothia scheme was set out to correct the wrong mapping of CDR1 insertions and thereby achieve structurally consistent mapping, but later this has been updated to also accommodate structural consistent mapping of framework indels.

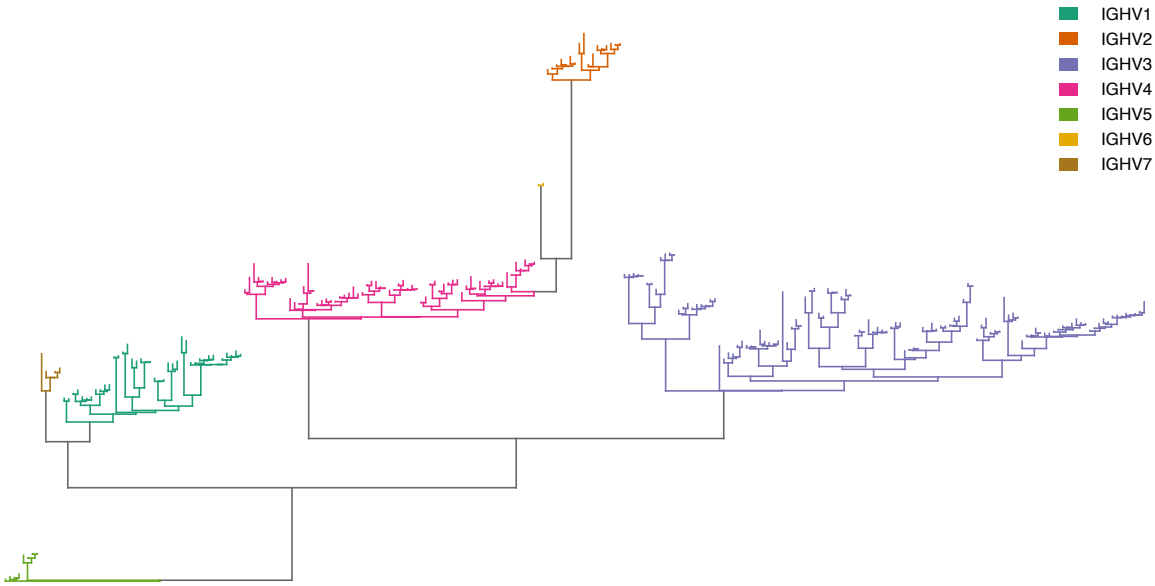


Figure 2-5: Tree of germline V genes. All IGHV alleles were downloaded from IMGT, pseudo genes were removed and a multiple sequence alignment generated with BAli-Phy [93] and MAFFT [45]. BAli-Phy was also used in tree inference, given a fixed topology within each IGHV subgroup determined by RAxML [91]. Figure credit: Andy Magee.

In 2003 the IMGT numbering was proposed to unify numbering of BCRs and TCRs [54]. Similar to the IMGT scheme another scheme, called AHo, was made unifying numbering of BCRs and TCRs [41]. Like Clothia, AHo numbering is structurally consistent, but unlike any other scheme it has a fixed standard length of 149 positions with enough space for even very long CDR3 sequences. Only in rare cases of germline indels, or an extremely long junctions, the scheme has to use insertion codes, which are otherwise standard use in all other numbering schemes. This fixed length sequence numbering make analysis much more convenient and results easy to interpret e.g. all BCR sequences can be encoded by a vector of length 149 and a given position in this vector always correspond to the same position on the protein structure regardless of the sequence. Therefore whenever BCR numbering is used throughout with work AHo numbering will be used if nothing else stated. The software ANARCI is used to annotate BCR sequences with AHo numbers [18], (`http://opig.stats.ox.ac.uk/webapps/sabdab-sabpred/ANARCI.php`).

With a numbered BCR sequence also comes the rather arbitrary decision of how to define the framework and complementary defining regions (CDRs/FRs). Clearly FRs are suppose to be more conserved and less structurally flexible while CDRs are suppose to be flexible and highly variable, but putting a boundary between them is a context dependent problem, that depends on the problem at hand. In this

work FRs are considered to be the most structurally conserved beta-sheets in the immunoglobulin domain, and CDRs are considered to be the structurally flexible loops connecting these beta-sheets. This definition is recapitulated in the CDR centric numbers tabulated by Andreas Plückthun (https://www.bioc.uzh.ch/plueckthun/antibody/Numbering/). The FR1-4 ranges are: `1-27, 41-57, 78-108, 138-149` and the CDR1-3 ranges are: `28-40, 58-77, 109-137`.

2.1.2 Biology of the germinal center reaction

It was discovered by Eisen and Siskind already in 1964 [20] that antibody affinity was increasing during the course of an immune reaction. The phenomenon is known as affinity maturation and by employing single cell techniques for staining, sorting and sequencing it has now been revealed that maturation is driven by a Darwinian selection process spatially confined into small nodules called germinal centers (GCs) residing in lymph nodes. Affinity maturation typically starts in the GCs 6 days after immunization and ends 4 weeks later when the GCs are dissolved and GC B cells die from apoptosis [103]. Before dissolving, a GC will export plasma cells, that secretes large quantities of high affinity antibodies, and memory B cells that will work as a permanent memory to be quickly re-initiated upon repeated antigen exposure.

Taking a step back, the formation of a GC starts with an immune reaction against an antigen. Two things are required in the initial phase, a) T cells with a TCR specific for an MHCII-peptide, with the peptide from the antigen and b) B cells with a BCR that can bind the whole antigen, engulf it and present its peptides in MHCII for the T cells to bind, see figure 2-6. When these two requirements are fulfilled GC foci will start to form in the lymph nodes. A GC is a micro-environment containing follicular dendritic cells (FDCs), T follicular helper (Tfh) cells and of course B cells. The FDCs are antigen storage cells, they engulf large amounts of whole immunogenic proteins and slowly presents them intactly on the cell surface where they can be extracted by B cells. The B cells undergo a mutation/selection process with two elements a) T cell help and selection and followed by b) mutation and proliferation. The two elements are spatially separated in the GC and defined as the light zone (LZ) and the dark zone (DZ), called so because the dark zone is more densely populated with cells and appears darker when viewed under the microscope. In the LZ B cells are attracted to the large surface of the FDC membrane through secretion of the chemokine CXCL13, some B cells will have BCR affinities strong enough that they are able to release antigen from the FDC membrane [94]. Bound and released antigen will be engulfed, processed and presented in MHCII molecules for the Tfh cells to bind. The more antigen a B cells is sequestering, the more peptide is presented and the more Tfh

binding will occur. If sufficient Tfh binding is achieved the B cell is signaled to migrate to the DZ by following another chemokine called CXCL12. In the DZ it proliferates while expressing high levels of the somatic hypermutation (SHM) inducing enzyme activation-induced cytidine deaminase (AID). With an approximate mutation rate of 10^{-3} per position per cell generation (10^6 higher than the normal rate) and a cell cycle time of only 6-12 hours, a large amount of variability is introduced in the DZ cells [103]. Eventually the progeny cells re-enter the LZ and the SHM induced variability will undergo selection, thereby completing the cycle. B Cells are constantly cycling between LZ and DZ by switching between high expression of chemokine receptor CXCR5 and low expression of CXCR4 to migrate towards chemokine CXCL13 in the LZ, and low expression of CXCR5 and high expression of CXCR4 to migrate towards CXCL12 in the DZ. This process is called cyclic re-entry and it continues until the GC dissolves. During selection there are a limited amount of Tfh cells, so this is where B cells have to compete to get the activation signal. The more antigen a B cell can sequester relative to the others, the more likely it is to get Tfh help, migrate to the DZ and go to proliferation, as oppose to undergoing apoptosis if Tfh help is insufficient. A fraction of the B cells in the LZ will get a special Tfh differentiation signal and these are exported outside of the GC with fate as either a plasma cell or a memory cell.

While many mechanistic details of the GC reaction is known much remains to be elucidated. The mechanisms of selection has been difficult to study but current evidence suggests that it is solely mediated through T cell interaction [103], [102]. Indeed using Tfh interaction as a model for understanding the observations, it is possible to explain all the modes of selection. Selection can be split into three main elements:

- Affinity
- Stability and expression
- Non-self binding

To improve antigen affinity is the most obvious point for selection to occur on. A gain in BCR affinity will enable a B cell to sequester more antigen from the FDCs and present more MHCII:peptide to the limited number of Tfh cells. Those B cells that have many MHCII:peptide complexes are much more likely to interact sufficiently to get the signal for proliferation, meaning that higher affinity is positively selected. An extension to this is in the case of mutations altering the stability and/or expressing of a BCR, resulting in less BCRs to be presented on the cell surface. Lower BCR

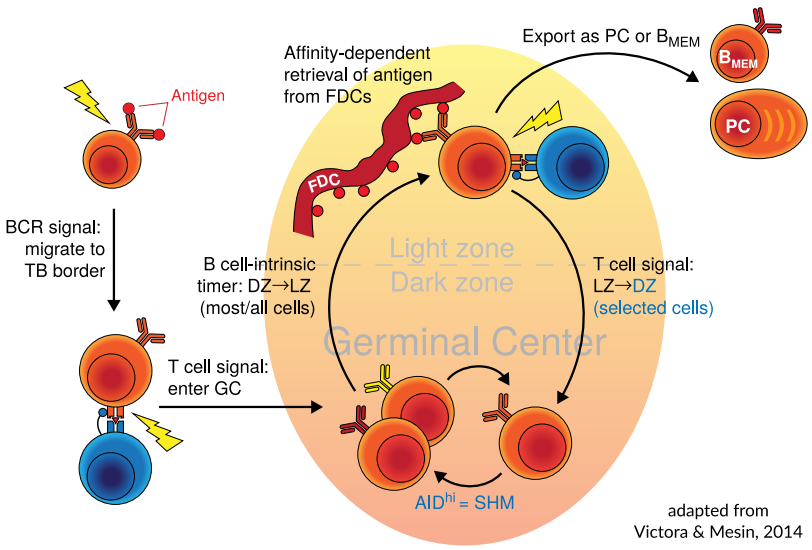


Figure 2-6: Dynamics of the GC reaction, adapted from [102]. Follicular dendritic (FDZ) cell in red, T follicular helper (Tfh) cells in blue and B cells in orange.

levels on the cell surface will cause less antigen sequestering from the FDCs resulting in less Tfh cell interaction and causing negative selection. Finally a more convoluted example of Tfh cell mediated selection is the negative selection of self binding BCRs. Self antigen binding B cells are simply clogging up MHCII molecules with self peptides leading to less T cell help, and again easily explained by a Tfh cell mediated selection process.

Despite the convincing mechanistic models it must be stressed that the GC reaction contains many, either mechanistically unknown, or purely stochastic elements. E.g. even though there is an observed relationship between clonal bursts and mutational gain in affinity, this does not appear to happen consistently i.e. large affinity gains are observed without clonal bursting [96], leading to the conclusion that either the reaction is highly stochastic, or a mechanism is unknown or unmonitored e.g. higher affinity but worse expression could make the net balance of selection turn negative, while it appears to be positive if only looking at affinity.

A notable case of a correction to the mechanistic understanding of the GC reaction happened recently. In 2012 a review by Victora, based on all previous evidence, suggested that GCs are initially colonized by little as 1-3 naive B cells [103]. However just 4 year later the same author concluded in Tas et al. [96] that this number was largely underestimated. Using an elegant set of experiments, they were able to visualize the colonization of GCs in multiple lymph nodes across different animals and concludes that mice GCs are consistently being colonized by 50-200 naive B cells. Before the Tas et al. paper in 2016 it had been a good approximation to assume that

GCs were being founded by just a single cell, and that this monoclonality would be upheld throughout the entire GC reaction. The GC identify was therefore a convenient definition of what a B cell clone is, but this view has to be redefined with the findings that GCs are starting out as highly polyclonal, and even ends up with a substantial fraction that keeps being polyclonal throughput the whole GC reaction. Complicating things further, Tas et al. also observed that B cells with the same naive sequences was found across multiple GCs. It would be fair not to distinguish between B cell clones with identical naive sequences, but matured in different GCs, as long as they mature against the same antigen, because then they undergo the same selection pressure. But two distinct clones would need to be defined if two B cells with the same naive sequence were independently matured towards different antigens. However it is a) practically very difficult to distinguish clones that have identical naive sequence and b) very unlikely that the exactly same BCR nucleotide sequence will mature towards more than a single antigen. Therefore this impractical definition of a B cell clone is discouraged and instead, the simple definition from Ralph et al. [73], is used throughout this work. In this scheme if two different BCRs are derived from an identical naive DNA sequence, they are in the same clonal family regardless of the GC context they came from.

SHM is driven by the enzyme AID that works by deaminating cytosines during DNA transcription. DNA repair enzymes are then recruited to the deaminated cytosines where they with some probability will introduce point mutations both at the site of deamination and at neighboring sites. Increased mutation rate during SHM is therefore the combined effects of AID and a number of DNA repair enzymes. With a whopping 10^6 times increase in mutation rate during SHM (from 10^{-9} to 10^{-3} mutations per bp per cell generation) there is a need to contain SHM so it does not destroy the function of the cell itself. SHM achieves this by working preferentially close to the chromosomal location of the variable region genes [114], [57]. In addition SHM appears to be biased and preferentially introduce mutations in some contexts rather than others, incidentally this context is over represented in the variable chain genes. The contextual bias of AID was initially described as hot/cold spots defined by a few specific 3-mer motifs (WRC/GYW hotspots and SYG/GRS coldspots) observed to change mutation rate significantly by preferentially recruiting AID [68].

It can be difficult to tease apart the different biases introduced by both AID and the DNA repair enzymes, and though the majority of the observed SHM bias most likely comes from preferential AID activity [67], it is their combined effect that is observed as SHM during affinity maturation. Instead of a mechanistic model of SHM an empirical context model can be setup purely based on a large number of observed

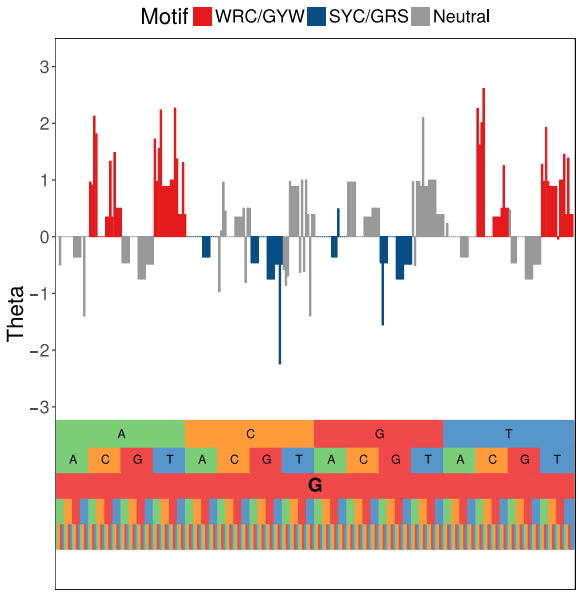


Figure 2-7: Plot of a 5-mer SHM motif model analogous to S5F [14]. Bottom row is the 5'-end of the motif progressing upwards to the 3'-end with G as the central base. Theta is the rate parameter of the survival model, used to fit the data, and is proportional to the logarithm of the mutation rate. Figure credit: David Shaw and Jean Feng.

mutations and their nucleotide context. Such a model has been described by Yaari et al. [110] and later in an improved version called S5F [14]. The S5F model is simply the complete set of 1024 5-mer DNA motifs and the relative probability of observing the middle base mutating. Given a motif and a mutation on its middle base, the S5F model also provides the probability of the nucleotide identity of the resulting mutation. A motif model is completely empirical and relies on the input data which is a large number of BCR sequences that have undergone SHM. A survival model [13] is a good way to describe the observed data, with the event rate Theta, being proportional to the logarithm of the mutation rate of the motif (unpublished data from David Shaw and Jean Feng), see figure 2-7. Fitting the model is non-trivial, because neighboring motifs share the same context and in the case of an observed event in one motif this changed the context, and thereby Theta, in the other motifs.

2.2 Monitoring adaptive immune responses

During the centuries of immunological research a large array of techniques have been directly invented or adapted from other fields of science to monitor the cells in an immune response. Starting with hybridoma techniques [51], antibodies could be pro-

duced and tested in a single cell setting after which sequencing would reveal the DNA encoding the antibody. A large number of other techniques was already used or followed e.g. ELISA, single cell staining and sorting, ELISPOT, fluorescence imaging techniques etc. Already in 1991, when Sanger sequencing was the most high throughput sequencing method, the diversity of the immunoglobulin genes and the matured BCR was being explored by sequencing single clones [111]. In more recent time sequencing by Illumina and Roche 454 have revolutionized the way we are monitoring adaptive immunity by enabling complete sequencing of millions of different BCR sequences.

2.2.1 Repertoire sequencing

In the following the focus will be on BCR sequencing but the same considerations apply to TCR sequencing. Several sequencing strategies have been devised to capture various aspects of the immunoglobulin genes, but most start at the same place; the BCR transcripts. The use of genomic DNA is also possible, but with only a single copy of a highly variable gene, this is prone to failure. Maybe more importantly, the transcript levels can also be useful information that genome level sequencing cannot provide and therefore most studies do bulk mRNA isolation from tissue or cells of interest, followed by reverse transcriptase PCR (RT-PCR) to create a pool of DNA with PCR primed ends, see figure 2-8.

An effective way to minimize errors introduced by the several steps of RT-PCR, PCR and sequencing is to attach a short unique molecular barcode (UMI, also known as unique identifier or UID) to the end of each read already at the RT-PCR step [98]. With enough random UMIs in the pool chances are extremely low that the same UMI will be incorporated into multiple BCR transcripts. Therefore each UMI will represent a single BCR transcript from the unamplified pool of mRNA, and reads with the same UMI should therefore be identical in the rest of the read. Conversely reads with different UMIs can have identical BCR sequences because a single cell can, and likely will, have multiple copies of the same BCR transcript. UMIs therefore also provide a way of transcript level quantification on top of its error correction.

Once the read library has been prepared sequencing is undertakes, usually either on Roche 454, or Illumina MiSeq. In the early days of Rep-Seq the advantage of Roche 454 was its long read length that could be adjusted to span the entire V gene, and additionally going all the way into J when a short CDR3 appeared. However with the development of the 2x300bp Illumina MiSeq paired-end reads, see figure 2-9, the 454 is out phased in favor of higher throughout and better read quality with much lower chance of indel calling.

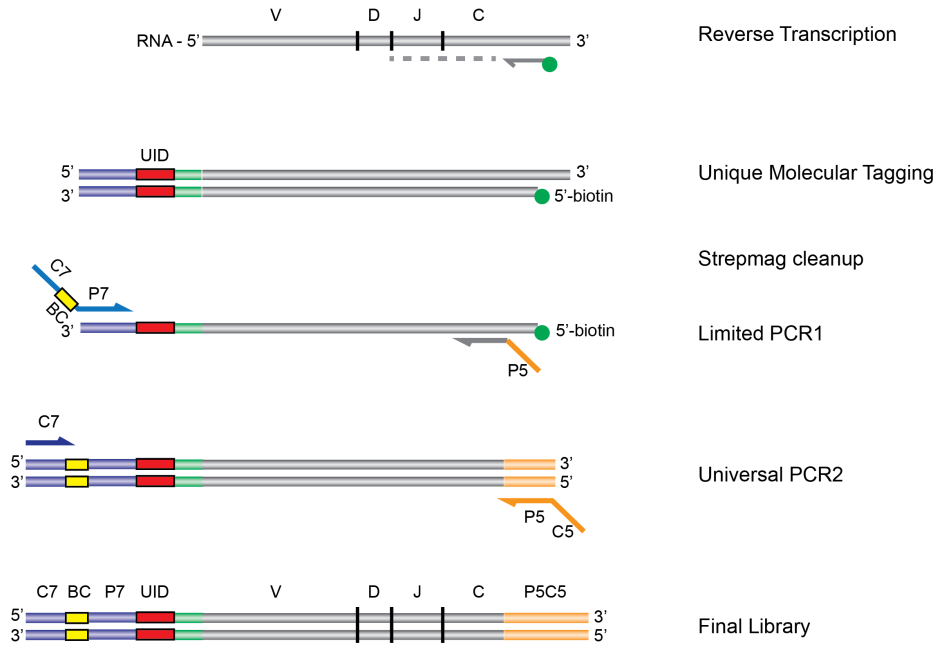


Figure 2-8: DNA prep method for BCR sequencing on Illumina developed and used by AbVitro (now Juno Therapeutics). In the first step an RT-PCR is run with template switching to introduce a UID. The fragments are then purified and the Illumina C7 clustering sequence, and barcode (BC), are attached to the 5'-end. Finally the C5 clustering sequence is attached and the library is ready for Illumina paired-end sequencing. A similar (but not identical) DNA prep method was used in Stern et al. [92] with figure 2-9 showing the resulting data format. Figure from Laustsen et al. [52].

After DNA reads have acquired from sequencing, error correction becomes a substantial issue. In the normal uses of high throughput sequencing, errors are just averaged out by the use of consensus sequences, but in the case of BCR sequencing the inherent variability of the BCR makes it difficult to tease apart what DNA variations that can be attributed to SHM and junctional diversity, and which can be attributed to PCR and sequencing errors. With the many different sequencing protocols there can also be large differences in the data processing afterwards, but pipelines exists that are fairly generic and will handle a wide range of data. A commonly used pipeline capable of processing a wide range of data is pRESTO (`presto.readthedocs.io`) [101]. Data processing with pRESTO is still sequencing protocol dependent but largely flexible and funnels down the data into the same end point regardless of protocol. In the case of paired-end Illumina reads with UMIs, similar to figure 2-9, the workflow can be summarized by the flowchart in figure 2-10. Input data is paired-end reads with Phred quality scores in fastq format. The data quality is first assessed by running FastQC [1], then bad reads are removed or ends are trimmed to sufficient minimum

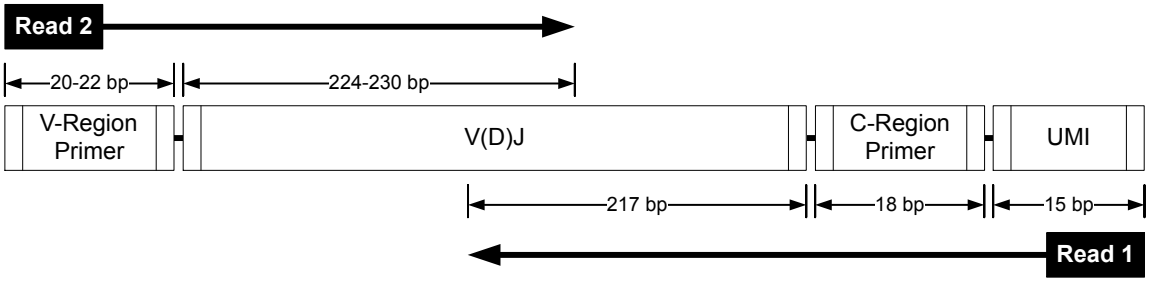


Figure 2-9: Example of a sequencing strategy for sequencing the full variable BCR chain and parts of the constant region. The strategy is designed for Illumina paired-end reads and seen used in Stern et al. [92]. Figure from pRESTO readthedocs [101].

quality. Many DNA prep protocols are using degenerate primers for PCR amplification by binding to the 5'-end of all possible V genes, see figure 2-9. This will possibly introduce what will look like a mutation, but in fact is a PCR artifact due to promiscuous primer binding. These should be removed by trimming off the primer binding region or "masking" it by reverting these mutations back to germline (red in figure 2-10). Then the paired-end sequences are merged by merging on the overlapping stretch of read 1 and 2 using simple alignment score maximization. Read pairs that failed to merge are discarded but the reads are still kept in separate files to build a consensus sequence over the sequences with the same UMI. Then anther instance of pairing is done on these consensus sequences to synchronize the files followed by merging reads into the same file (orange and green in figure 2-10). Next a series of filtering and deduplication steps are undertaken e.g. only allowing sequences with multiple reads, a maximum number of ambiguous nucleotides etc. to pass through. The final result should be a set of high confidence BCR sequences, and depending on the sequencing protocol, with or without information about the constant region and/or mRNA abundances.

All of the above described sequencing methods suffer under the substantial caveat that they are mixing the heavy and light chain sequences from multiple cells under the mRNA isolation step. This means that a given heavy chain sequence could have been paired with any of the light chain sequences observed for the same sample - and with high throughout sequencing this is millions. Pairing is not always necessary e.g. in some cases of diagnostics, but if the function of an antibody is to be tested the correctly paired heavy/light chain needs to be found. Some authors have done so using random pairing and selecting for binding via phage display [33], others have ranked heavy and light chains according to their frequencies from sequencing and matched the highest frequency clones [75], yet others have used phylogenetics to infer trees and then pair similar topology heavy/light chain trees, guided by a known pair

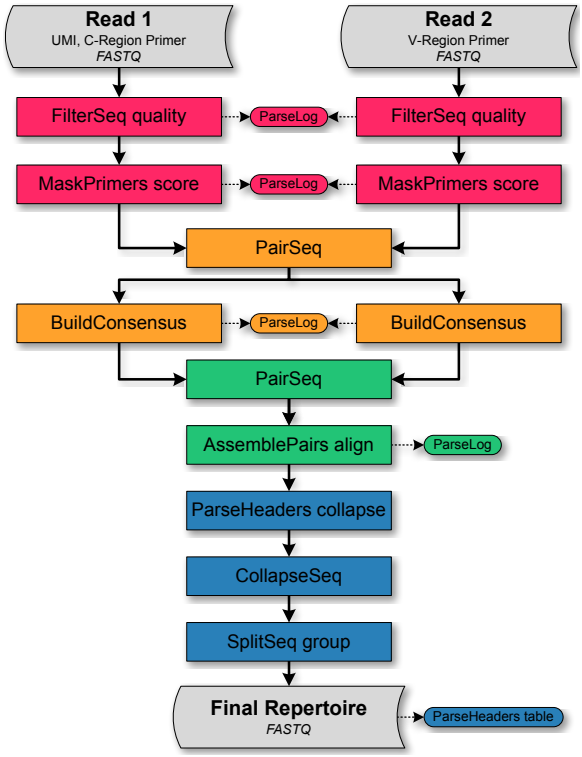


Figure 2-10: Flow of the raw sequencing data through the pRESTO pipeline. This strategy is designed for Illumina paired-end reads, e.g. 2-9, and seen used in Stern et al. [92]. From pRESTO readthedocs [101].

to link to trees [116], [50], [42]. The latest development is to use micro-droplets to capture single cells and then perform the DNA prepping steps inside these drops. Some authors are pairing heavy/light by physically linking them together via overlap PCR [60], while others are amplifying a UMI inside the drop and attaching it to both heavy and light chains [8]. Though the methods for paired heavy/light chain sequencing now exists they are not widely used because of the requirement of having a micro-droplet platform and expert knowledge which is not common stock for most labs. Still the vast majority of public BCR sequencing data is unpaired and it will likely remain so for at least some years to come.

2.2.2 Inferring B cell clonal families

Once high quality sequences have been obtained from a BCR repertoire sequencing they become the input to the following analysis steps, either as hypothesis testing or hypothesis creating. A first step in the analysis is simply to annotate sequences by their germline VDJ genes, which can be done simply by aligning each sequence from a database of germline genes to the BCR sequence at hand. The V, D and

J genes achieving the highest alignment score are the winners and inferred to be the true germline gene present on the ancestral naive sequence. This approach is used by many studies to VDJ annotate BCR sequences, usually either through the NCBI hosted IgBLAST [112] or IMGT's V-QUEST [56]. The advantage of using IgBLAST is its fast turnover and stand-alone software under a permissive license. Next, an extension to inferring the germline genes is to infer the full naive VDJ sequence of its unmutated ancestor. Inferring the full naive sequence is a bit more complicated because it requires inferring the original sequence flanking the D gene, where the junctional diversity from N/P nucleotides is complicating the problem. Given a BCR sequence and assuming some distribution of N/P nucleotide insertion, how many bases is trimmed of the V-D junction and how many are trimmed of the D-J junction? This question is non-trivial because of SHM e.g. if a single mismatch is preventing a V gene alignment match to extend further 4 bases downstream, see table 2.1, should that be regarded as a mutation due to SHM, or is it regarded as the last 3 random N/P nucleotides incidentally had the same identity (uniform random chances $0.25^3 = 0.0156$)? Clearly it looks like an extension to the alignment would be the optimal solution but what then if the last base was also a mismatch? Or that about the effect of the underlying distribution of N/P nucleotide insertions? In these complex cases intuition starts breaking down, instead there is need for a statistical model that can integrate over probabilities and give consistent estimates.

```
V gene   ...GTTGAGTGT
         ...|||||*|||
BCR seq  ...GTTGAATGT
```

Table 2.1: To extend, or not to extend, that is the question (for an HMM to answer).

The classical "go to" statistical model for biological sequence analysis is a hidden Markov model (HMM) [19]. Briefly an HMM is set of user defined states, e.g. the V, D or J genes, N/P insertions, etc., and the jump between states are then connected by probabilities. Each state has a list of emission probabilities that defines the probabilities of observing a given base. The hidden/unknown part is the identity of the states at a given position in a sequence e.g. when is the V gene stopping and the N/P insertion starting, like the example above. This is one of the question that is possible to answer using an HMM, specifically by calculating the so called Viterbi path, but what is more interesting is the inherent flexibility of an HMM to represent well known biological mechanisms in a statistical framework.

To address the problem of BCR sequence annotation Ralph et al. developed an HMM model of VDJ recombination and built it into a software called partis [72]. Us-

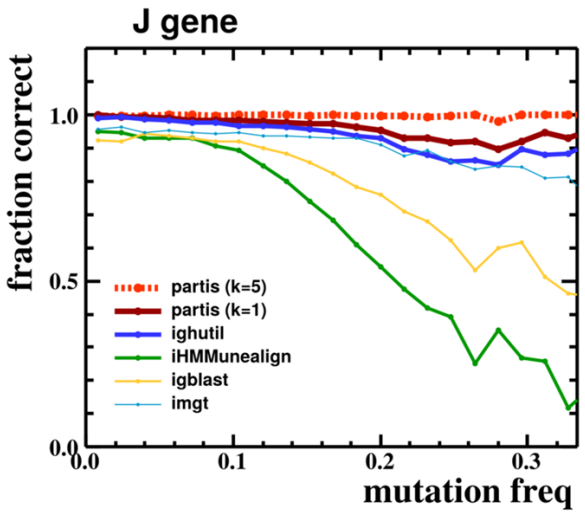


Figure 2-11: J gene assignment performance for different methods. With higher mutational burden most methods struggle to make the correct gene assignments, presumably because of the problem outlined in table 2.1. The HMM method in partis is robust to the mutation burden, and achieves even higher performance by integrate over multiple (k=5) sequences from the same clonal family. IHMMunealign and partis are the only HMM methods while the rest are alignment based. From [72].

ing extensive simulation studies, the authors observed clear shortcomings of the purely alignment based method like IgBLAST and IMGT V-QUEST. Especially, alignment based methods appeared to have a low fraction of correct gene calls for the short germline genes D and J carrying a few mutations, see figure 2-11. The length of the V gene makes it easy to correctly assign just by alignment, and therefore little performance difference was observed, however notice that this only applies for correctly calling the gene identity, regardless of allele identity. With the more subtle variations among the alleles, there probably is an even bigger performance gain here, by using HMMs like partis compared to pure alignment based methods.

Now returning to the alignment problem exemplified in table 2.1. The real shortcoming of using a pure alignment methods is that there is no robust way of deciding what is N/P nucleotides and what is inherited from germline genes, thereby making it very difficult to reconstruct the true naive sequence. A problem that is well handled by an HMM which will also, as a side effect of calculating the Viterbi path, return the maximum likelihood estimate of the naive sequence. The speculations were confirmed in the simulation study of Ralph et al. that showed substantial performance gains in naive sequence reconstruction using HMMs vs. alignment methods, see figure 2-12.

Next step is to partition the BCR sequences into clusters of sequence related by some definition of relatedness. These clusters are by some known as clones [36],

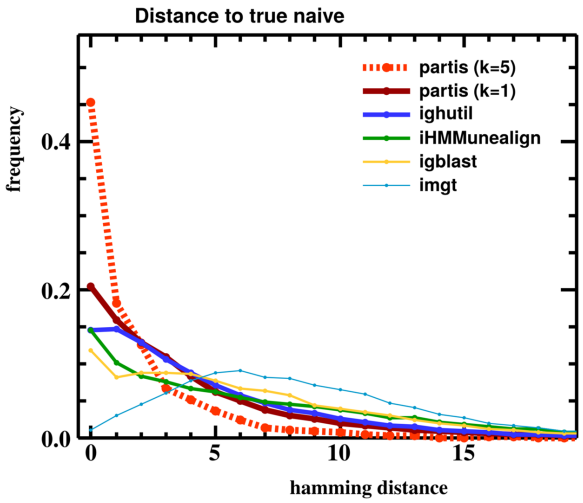


Figure 2-12: Distribution of hamming distances true vs. inferred for 30,000 simulations compared across different inference methods. There is a clear advantage of using HMM methods like partis, but the largest performance leap is to integrate over multiple (k=5) sequences from the same clonal family (explained in the clustering section). IHMMunealign and partis are the only HMM methods while the rest are alignment based. From [72].

which is assumed to mean that they actually came from the same GC reaction. However as previously discussed the view of GCs as monoclonal is outdated [96], so in this work the nomenclature from Ralph et al. [73] is used, with the assumption that there is sufficient BCR diversity to distinguish clonal families exclusively based on their shared naive sequence. Given this definition a logical first step would be to require that the assigned germline genes in a cluster to match, and indeed such simple clustering has been extensively used. However in acknowledgement of the uncertainty in assigning the correct D gene, only V and J genes are usually used, and regardless of their allelic variants. Furthermore the junction length or the CDR3 length has been used as second discriminator and the hamming distance between sequences is usually used as the last discriminator. Then the procedure for clustering is first to split BCR sequences into buckets with the same V and J gene and junction or CDR3 length, and then sequences in each bucket are clustered based on a distance measure like hamming distance [32], SHM weighted hamming distance [36] or amino acid based hamming distance confined to CDR3 [44]. This method, which will be referred to as "VJ junction agglomeration", has the inherent problem of putting 100% confidence on the V, J and junction annotations. However with just moderate SHM there are substantial uncertainties in the annotations and in such cases putting a hard boundary between sequences with different annotations causes problems. In an example, outlined in figure 2-13, a clonal family would wrongly be assigned into three

different clusters (boxes defined by the solid gene lines) if VJ junction agglomeration is used. Contrarily, if integrating over germline and junctional annotation uncertainties, clustering is not restricted by the point estimate of the gene or allele annotation, and at least has the possibility to merge all red dots into the same cluster.

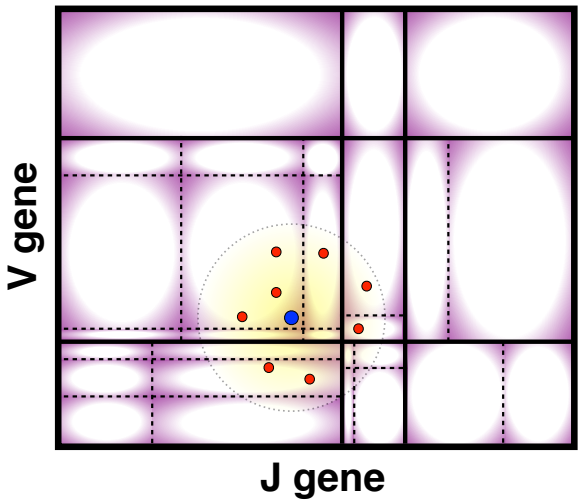


Figure 2-13: Two dimensional representation of the BCR sequence space that illustrates how V, J and junction point estimates leads to overestimation of the number of clusters. Vertical lines represent V genes and dashed lines represent alleles, same for J genes on the horizontal axis. Naive sequences with no N/P nucleotides are in the cross section between a V and J allele. Colored with purple gradient is the range of junctional diversity extending from the V/J gene combinations, less color means lower probability, all the way to white which is sequences not within the reach of any VDJ recombination. The dot in blue represents a naive sequence with its SHM "breadth" marked by a yellow circle. Red dots are observed BCR sequences from the clonal family defined by the naive sequence.

Now it should be clear that the root cause of the clustering problem is propagation of errors in point estimates, i.e. ignoring the substantial uncertainty in VDJ annotations of a BCR sequence that has undergone SHM. When a clustering criterion is based on the point estimate of both the V and J assignment, both the assignment uncertainties are affecting the clustering performance negatively. The junction length is also problematic to use because it is also just a point estimate, and often estimated from by a problematic alignment method as outlined in the example in table 2.1. Lastly, hamming distance is weighting a mismatch in the N/P base region equally likely as a mismatch in the middle of germline gene while there is much more certainty of a real SHM event in the later case. In the example in figure 2-13 VJ junction agglomeration is overestimating the number of clusters and indeed this intuition is also observed in simulation studies [73].

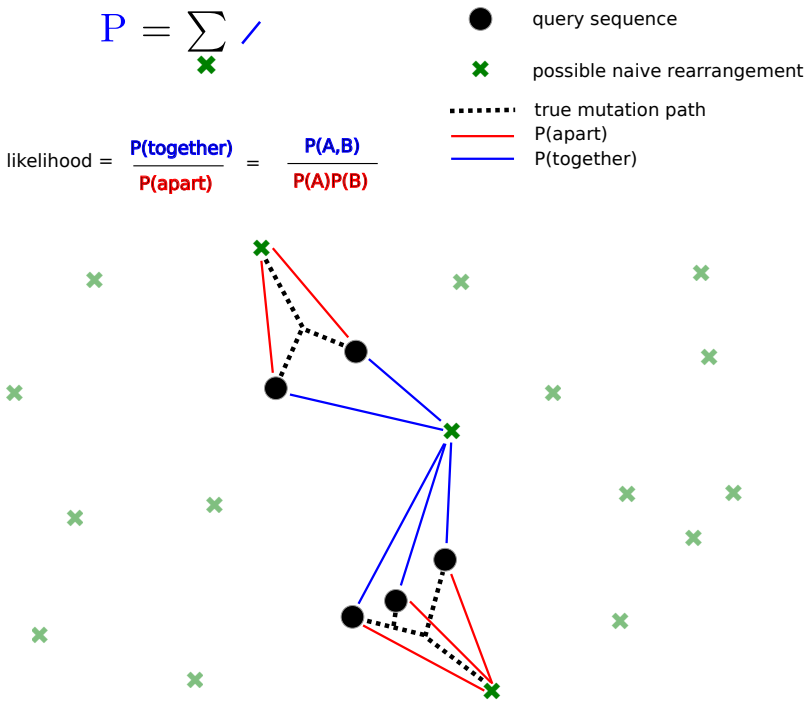


Figure 2-14: Likelihood ratio test to decide whether to merge a set of sequences into a cluster or not. Figure credit: Duncan Ralph.

A completely different approach is taken by Ralph et al. [73], extending on their HMM framework for germline gene annotation. They use the naive sequence as a centrality point for clustering, not in terms of hamming distance, but in the terms of likelihood. The HMM model is conveniently yielding a likelihood function defined over all possible naive sequences for each BCR sequence in the dataset. With this likelihood function it gets simple to do hypothesis testing based clustering e.g. to test whether a set of observed BCR sequences all belong to the same cluster through a common naive sequence ancestor, see figure 2-14.

Notice that the likelihood ratio test based clustering in partis is centered around the likelihood of the proposed naive sequence and the thus allow for merging of sequences with different VJ gene annotation point estimates. When clustering has run to an end the final annotation of each sequence will change to reflect a common VDJ rearrangement for all cluster members. The shared information among cluster members makes inference of the naive sequence much more probable because all sequences can be integrated into the same HMM, referred to as multi-HMM in Ralph et al. The strength of using multi-HMM to infer a naive sequence is obvious from a theoretical stand-point. Multiple sequences are regarded as multiple independent observations of the same mutation process and this will increase the probability of the germline annotation, but more so, it will increase the odds of observing the true

naive sequence in the N/P junction. This claim is supported by striking improvement in both annotation (figure 2-11) and naive sequence inference (figure 2-12) for just 5 clonal family sequences (k=5).

The advantages of HMM based BCR annotation and clustering should now be clear and therefore all germline annotation, naive sequence inference and clustering of BCR sequence throughout this work was done with the partis software (github.com/psathyrella/partis).

2.3 Phylogeny of a clonal family

The problem of determining phylogenies is the problem of reconstructing the unobserved evolutionary history, usually, using only a sample of states at a single time point. Since there are no observations of the events prior to the sampling time we cannot be certain about a phylogeny. The problem turns into an inference problem. For B cells this translates into finding the evolutionary history starting from a naive B cell, progressing through rounds of SHM in the GC reaction and finally getting sampled at some time t. The only information available is the BCR sequences at the sampling time.

To represent the process of cell division and apoptosis in the GC it is convenient to use a tree. The root of this tree is the naive B cell, a branching event is a cell division and a terminal leaf is a B cell that have died. The tree model of evolution is also a useful way of representing relationship among sequences e.g. some sequences which are all members of the same tree clade share a common ancestor and most likely are more similar than some other sequence not included in this clade. The phylogenetic tree should be viewed as a model framework used to describe the evolutionary process, the details that goes into the model framework depends on the problem at hand.

2.3.1 Parsimonious tree inference

Now given that the evolution is following a tree model, the model for inferring the topology must be specified. A classical method for inference is to maximize tree parsimony. Motivated by Occam's Razor the best tree is simply the tree that with the fewest changes explain the observed data. Practically this is done by calculating the tree parsimony score with Fitch's algorithm [26] and then finding the tree with the lowest score. The parsimony score of a tree (T) can be calculated as a sum of all the scores at each site (p_i): $S(T) = \sum_{i=1}^{m} F(p_i|T)$.

With a parsimony score function the possible tree space can be searched and the

maximum parsimony (MP) tree can be found. Tree space can be searched exhaustively only with few taxa but everything above 10 taxa has more than tens of millions of tree making it practically impossible to search exhaustively [24]. The branch and bound algorithm is a way of decreasing search space but still finding the best solution [38], however even with this hill climbing methods have to be used. In a hill climbing tree search various heuristic tree moves like, nearest neighbor interchange (NNI), subtree pruning and regrafting (SPR) and tree bisection and reconnection (TBR) are used to explore widely around the tree space.

It it clear that the MP method for tree inference is very simple and while this will work in some applications the criterion of maximum parsimony is not strictly how evolution works. In cases with rate heterogeneity due to a biased mutation processes or selection some sites might evolve much faster than others and this makes the parsimony assumption of equal weight on all substitutions break. Such complex processes are not accounted for in the MP method making it prone to errors, but even with the simplest sequence evolution process theoretical arguments by Felsenstein have shown that parsimony is inconsistent [23], and since adding parameters to model a complex process is not readily compatible with parsimony, most researchers have turned to model based tree inference.

2.3.2 Model based tree inference

As an alternative to maximizing the parsimony a likelihood function could be formulated to return the likelihood of the data given the tree. Then the problem turns into finding the most likely tree, referred to as the maximum likelihood (ML) tree. The likelihood function is conditioned on a tree with branch lengths representing distances in a continuous mutation process driven by a mutation rate.Using gradient decent tree parameters like branch lengths can be adjusted to their ML estimates. To accommodate more advanced things such as differences in substitution rates, e.g. higher rate of `T->G` than `T->A`, a rate matrix can be defined. If the data is at DNA level the rate matrix contains all possible substitutions between DNA characters. Examples such matrices: the F81 model [25] or the widely used general time reversible (GTR) model [97].

DNA substitution models are not aware of protein encoding and therefore does not explicitly discriminate between synonymous and non-synonymous mutations, something that is otherwise crucial for proteins under tight selection. Therefore it might be useful to translate a DNA sequence into its protein sequence and then use an amino acid substitution matrix like PAM [15], BLOSUM [39] or others. Information is lost in the translation process so if an amino acid model is to replace the DNA model,

information loss needs to be outweighed by the gain in performance. Alternatively a codon model is defined on DNA level, but with DNA level substitutions working on units defined by protein encoding codons. In this way a codon model will both account for a difference in the rate of synonymous and non-synonymous substitutions and the rate difference between amino acids. Two popular choices for a codon model are GY94 [34] and MG94 [64].

Bayesian phylogenetics

In the ML framework both tree, branch length and substitution parameters are all point estimates adjusted to maximize the likelihood function, but point estimates might not always be desirable. E.g. when information like time or rate is to be extracted from a tree then a single point estimate is not very useful because the variance of such estimate is unknown. Bayesian phylogeny addresses this challenge by integrating over the distribution of trees, branch lengths and substitution parameters by Monte Carlo sampling. When sampling has converged the resulting posterior distributions of all parameters can be used to express confidence in the maximum a posteriori (MAP) estimate e.g. by providing the high density interval (HDI) along with the MAP estimate. The real drawback of Bayesian phylogeny is that sampling can make runtime much slower than a similar ML problem. There can also be a problem of reaching convergence in cases where sampling is difficult.

2.3.3 Ancestral sequence reconstruction

As a side result of inferring a phylogeny it is also possible to reconstruct the sequence of those nodes that are internal and unobserved in a tree. This process is known as ancestral sequence reconstruction (ASR). For the MP algorithm these ancestral states are part of the tree inference and found via Fitch's algorithm, but for a model based method ancestral sequences must be estimated by extracting the maximum likelihood estimate of the sequence at each internal node. ASR on ML trees can either be done as a marginal or a joint reconstruction. In the marginal case each node is reconstructed independently from one another, found as the ML estimate without fixing the sequence of any of the other internal nodes. Notice that when one ancestral sequence is found this will change likelihood function for the tree and thereby affecting all the other internal node. This effect is ignored in marginal reconstruction but accounted for in joint reconstruction where the problem is to find the ML estimate of all the ancestral sequences jointly.

ASR is an important way of exploring the evolutionary trajectory and behaviour of

ancient proteins. It has been used to investigate the specificity of the important cancer drug type kinase inhibitor [107] and studying evolutionary trajectories of broadly neutralizing HIV antibodies [17]. Yet there has been little validation of ASR methods partly because it is difficult to test experimentally. An exception to this was enabled by a clever study design made by Randall et al. [74], using a controlled system of evolution on a single gene with sampling of ancestors, they could experimentally validate ASR inference methods on the using only the final time point. Results showed high accurary of 97.17-98.88% correct reconstruction and only small difference between chosen methods.

2.3.4 Genotype collapsed tree

While B cells in a GC reaction are having a high mutation rate there are still many B cell with the same genotype. Especially after a clonal expansion many B cells will have the same genotype, and furthermore ancestral B cells might have the same genotype as their descendents. BCR clonal family sequences are generally of low complexity and not very divergent, but because there are many B cells in a GC their phylogenetic trees are seemingly complex with hundreds to thousands of taxa. To reduce this complexity we use the concept of a genotype collapsed tree (GCtree). In a GCtree B cells with the same genotype are merged into a single node carrying their total abundance and a leaf descending from a B cell with the same genotype is merged upwards, see figure 2-15.

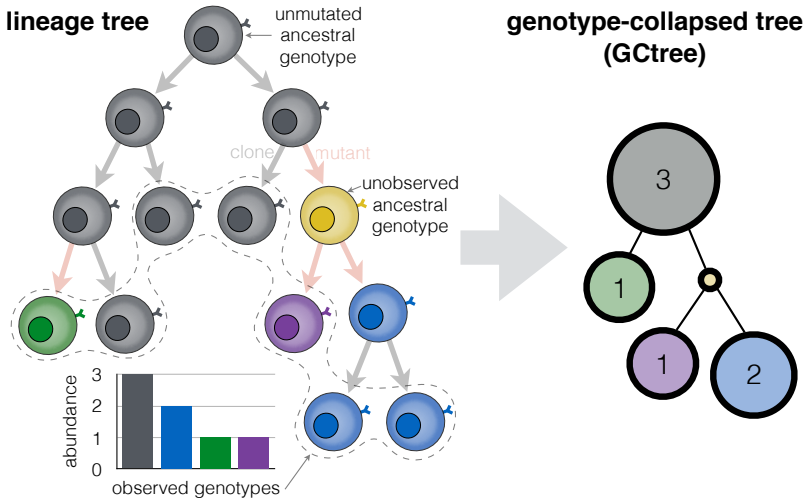


Figure 2-15: Genotype collapsed tree. Observed cells fenced by a dashed line are getting deduplicated and the total abundance is recorded. Next the tree is collapsed upwards, merging cells of the same genotype, ending up with the tree to the right. Figure credit: William S. DeWitt.

We find that the GCtree gives a good visual overview of the BCR phylogeny enabling interpretations such as recent clonal bursts, fitness advantages etc. In the rest of this work all trees will be shown as a GCtree using custom tree plotting from the ETE package [43]. Figure 2-16 is an example of how a standard phylogenetic tree is collapsed and represented in GCtree format. Embedded in the GCtree visualization is a number of traits: genotype abundance is annotated in the middle of each node and proportional to the node size, dashed branches represents synonymous mutations while fully drawn lines are non-synonymous, branch length is proportional to the hamming distance between nodes and branch thickness is proportional to the number of amino acid differences.

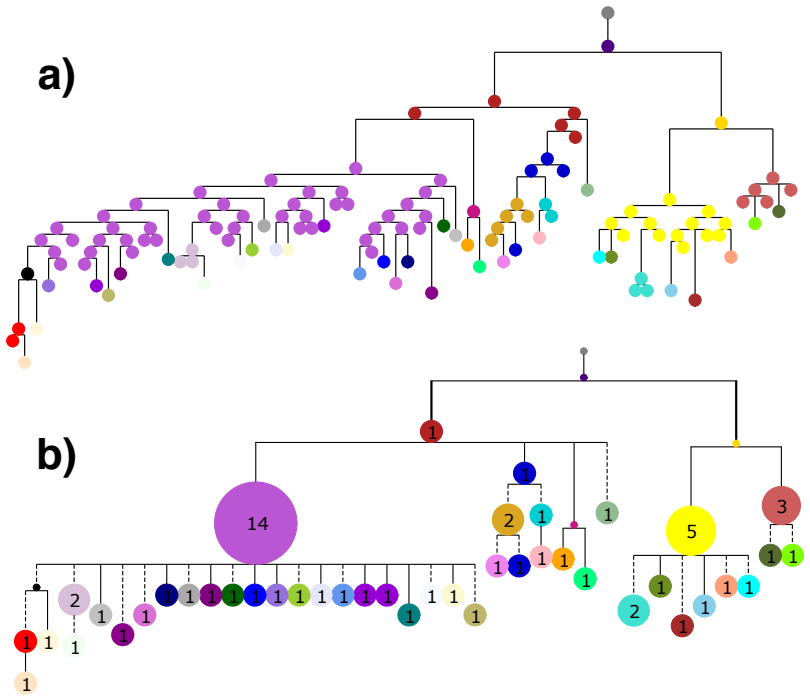


Figure 2-16: Genotype collapsing of a simulated tree with node coloring according to genotype. In a) a full lineage tree showing a simulation of several rounds of replication. Only the leafs are sampled at the end of the simulation and collapsed into the GCtree in b). The GCtree shows the leaf abundance of each node by way of node size and the integer in the middle of the node. In the GCtree synonymous mutations are indicated by dashed branch lines, branch length is reflecting the hamming distance between nodes at DNA level and solid branch line thickness reflects the number of non-synonymous mutations.

2.3.5 Clonal family tree

Most often the source of BCR sequences is throughput sequencing (HTS) data generated from bulk RNA, extracted from peripheral blood mononuclear cells (PBMCs). Clonal family relationships therefore needs to be inferred by means of clustering before sequences can be inputted to phylogenetic tools. An automated pipeline can be envisioned, with HTS data as input and inferred clonal family trees as output. One attempt at this task is the publicly available ImmuneDB [82]. Using relatively simple clustering and inference of naive sequences ImmuneDB still has room for improvements, but the concept remains extremely important as a way of aiding the interpretation of Rep-Seq data. On the conceptual side of the clonal family tree representation is it possible to map meta information about binding affinity, cross reactivity, self-binding etc. back to individual trees to get a detailed picture of the evolutionary process at a single cell level.

Chapter 3

Simulating sequences undergoing affinity maturation

3.1 Introduction

Antibodies have an important role in adaptive immunity by specifically binding to and neutralizing invading pathogens like bacteria and viruses. Developing specific antibodies involves reactions across the entire immune system, but it is B cells that mature and secrete antibodies, after undergoing several rounds of selections in the bone marrow and the germinal centers (GCs). B cells undergoing GC affinity maturation express antibodies as B cell receptors (BCRs), and these are subject to a strict selection scheme, where many different factors come into play e.g. antigen affinity, receptor expression and folding, codon choice affecting translation rates and affinity to self antigens. All these factors leads back to the same thing: the ability to bind antigen, and this is measured by nearby T cells, which will be affecting the mutual probabilities of proliferation or apoptosis [3], [103]. As an example in a GC there is a direct T cell mediated selection for higher BCR affinity by means of sequestering more antigen. Selection against binding of self antigens is mediated by the fact that binding self antigens will block binding of pathogen antigens, resulting in decreased T cell help. Therefore, in its most simple way, affinity maturation is driven by affinity towards the presented antigen, and all the effects also attributed to selection, like folding, stability and aggregation, is selected by means of their shared effect on antigen affinity.

Affinity maturation is a strong selective pressure imposed on the B cell evolution, and this poses a challenge to phylogenetic reconstruction. In order to validate the correctness of phylogenetic methods, simulation studies are among the most impor-

tant tool. Simulations are classically done by randomly permuting a sequence along a tree or sampling mutations from a distribution of substitutions, but a simulation can also be configured to have selection.

There already is a large body of literature suggesting different ways of modelling the GC reaction from mechanistic models [11], [78], to probabilistic frameworks [90], [76], systems of differential equations [86] and birth-death processes [2]. Shlomchik et al. [90] developed "Clone," a simple statistical framework inspired by the intuition that CDRs are more likely to accept mutations than FRs. In Clone mutations are introduced, by an intrinsic mutation rate (λ), into either CDR or FR and being either synonymous or non-synonymous with the two probabilities drawn from two Bernoulli distributions. This gives four categories associated with their own probability, and for each category a Poisson progeny distribution can be defined to simulate death and/or cell divisions. Later Kleinstein et al. [47] used Clone as a statistical framework to find the maximum likelihood estimates of its model parameters, given some real BCR sequence dataset. This resulted in consistent estimates of the rate of somatic hypermutation that have later been confirmed in multiple studies. A very similar model was suggested by Reshetova et al. [76], but unlike Clone, authors claimed that software can be accessed on request with a permissive open source license. An alternative model described by Shahaf et al. [86] is based on a system of differential equations and uses distance between bit strings to represent complementarity between BCR and antigen. Unfortunately, software implementing this model has not been released publicly. A very different approach is to model the GC reaction by a mechanistic model where each B cell is explicitly represented as a single entity, called an agent. Mechanistic models start with a set of first principal physical laws used to define the molecular behaviour of a system. The system is initiated with a 3 dimensional grid with a number founding cells, and then solved by discretizing into small time steps and solving the equations for diffusion, cell motility, receptor binding etc. Examples of mechanistic models are hyphasma, described in Robert et al. [78], and the model described in Wang et al. [104] used to explain generation of cross-reactive HIV antibodies. Unfortunately both of the above examples are not publicly available. Childs et al. [11] also used an agent based method to simulate the GC reaction and explaining the trade-offs between antigen specificity and breadth, and in addition this also appears to be one of the only GC simulation software to be published as open source somewhere in the public domain (`https://github.com/cobeylab/gcdynamics`).

Regardless of software availability none of the above mentioned simulation methods are suitable for validating BCR phylogenetics and testing the effect of antigen selection on tree inference. While the simplistic statistical framework of Clone is

appealing it does not model selection towards a known high affinity BCR, rather it models selection as a random effect of a non-synonymous mutation located in the CDR. None of the other above mentioned simulation methods are explicitly simulating the DNA sequences and therefore cannot be used in software to analyze DNA sequences without substantial adaptations. To get the best of the realism of a mechanistic model and the simplicity of a statistical model, we propose a simple model of affinity maturation that models sequence fitness solely as a function of antigen binding, modelled by a standard binding equilibrium. If the BCR affinity is high it means that the B cell will bind and endocytose a lot of antigen, thereby getting plenty of T cell help, decreasing the chances of undergoing apoptosis while increasing the chances of proliferation. Alternatively, if the BCR affinity is low no antigen will be endocytosed, and chances are high that the cell will undergo apoptosis. In this setup it is not the absolute affinity that matters, but rather the affinity relative to the affinities of all the competing cells in the whole GC, and the total amount of antigens cells compete for. By modularizing the simulation code we can first make a model for a neutral branching process, and then add an optional affinity selection step that works in conjunction with the neutral process used to introduce substitutions.

The purpose of the presented simulation method is to generate more realistic sequence simulations of clonal family evolution, to be used for assessing the performance of phylogenetic methods. The model is designed to capture only the most influential effects of affinity maturation, without the need of a detailed mechanistic model, but still sufficient to recapitulate the features of real GC sequence data.

3.2 Methods

3.2.1 Neutral branching process

A neutral branching process, independent from the later described affinity model, was setup as a reference point for simulation. It can be viewed as a model of cell divisions, where at each generation a cell can either die or produce a number of offspring, and each offspring has some probability of carrying a mutation. The neutrality assumption lies in the fact that the mutations introduced will have no fitness effect. The root sequence is given at initialization as a starting point to evolve through a stochastic branching process, controlled by an arbitrary discrete distribution, which in this case we take to be a Poisson distribution, due to its convenient mathematical properties. Thus, in the below sections the branching process is controlled by a Pois(λ) progeny distribution. Furthermore at each generation all progeny cells will undergo a muta-

tion process with the number of nucleotide mutations drawn from another Poisson distribution ($\text{Pois}(\lambda_{\text{mut}})$) and then introduced sequentially into the sequence given a substitution model. By using sequential introduction of mutations we allow the possibility of back mutations. A possible possible choice of substitution model is a uniform probability over all sites and substitutions, but because it is well known that BCR sequences do not mutate uniformly at random [113], we use an empirical approximation of the mutation context sensitivity, called S5F [14]. The S5F model describes mutability of the middle base of all 5-mer DNA motifs, and for each motif, the base preferences given a mutation i.e. if a random draw from the S5F site mutabilities of the sequence chose to mutate the 5-mer `AAAAA`, then the S5F model will also provides a list of probabilities for each middle base substitutions, either $P(\texttt{AAAAA} \rightarrow \texttt{AATAA})$, $P(\texttt{AAAAA} \rightarrow \texttt{AAGAA})$ or $P(\texttt{AAAAA} \rightarrow \texttt{AACAA})$, probabilities summing to 1. A 5-mer mutability cannot be used directly on sites at the start or end of a sequence because of missing context, therefore we fill in missing context with the unknown base, N, and average over all possible motifs fitting into this ambiguous context.

Termination of the neutral branching process is enforced in either of three ways: 1) by simulating under a subcritical process ($\lambda < 1$) [80] and following it until extinction, 2) by using a stopping time T, or 3) by stopping after a max population of N has been reached. Leaves are sampled from last time point, or in the case of 1) only terminated leaves. In addition we introduced a parameter for down-sampling the cell population to n cells. Down-sampling allows for emulating the incomplete sampling of a GC cell population, which is expected in HTS data due to fall out during PCR, sequencing, quality control steps and limited sampling e.g. in the case of data from mRNA extracted from PBMCs. The five parameters of the model is tabulated in table 3.1, but only one of the stopping criteria can be used in a run, effectively making it a four parameter model.

Parameter	Description
λ	$\text{Pois}(\lambda)$ progeny distribution
λ_{mut}	$\text{Pois}(\lambda_{\text{mut}})$ mutation distribution
T	Stopping time
N	Stopping number of sequences
n	Down-sampled number of sequences

Table 3.1: Parameters used in the neutral branching process simulation.

3.2.2 Simulations with affinity selection

To model the affinity maturation process with selection we will use the exact same framework as described for the neutral branching process, but now the Pois(λ) progeny distribution is no longer constant. We will consider the magnitude of λ as the fitness of a cell. In the neutral model λ is a fixed constant resulting in a completely flat fitness landscape, while in a system with selection the fitness landscape is more complex e.g. like a rugged surface. The following subsections will describe a model with the simple purpose of defining a function to calculate the fitness of any BCR sequence that could arise from the mutation process. The fitness is measured in terms of a single λ associated with each cell and defines a sequence specific progeny distribution. Thus, selection is condensed into a dynamic λ, and this is the only difference to the neutral branching process.

Model concept and biological assumptions

Now let us make some basic assumptions to keep later definitions simpler. First of all the system we intend to model is the affinity maturation that is happening in the GC reaction, as guided by the BCR's affinity towards a target antigen. A real GC reaction is seeded with 50-200 naive B cells and is therefore in its initial state highly polyclonal [96]. However, during the extend of a GC reaction the lineages descending from these seeds gradually die off [96] and the GC may even reach full monoclonality, in which a single lineage comes to dominate. Because we are interested in relatively large lineages that eventually lead to immune memory, we thus make the simplifying assumption that the simulated GC is monoclonal seeded by a single cell. Although polyclonality could be integrated into the simulation method, this would not change the simulation results significantly and therefore prefer to keep the description simple. In this model it is the BCR amino acid sequence that is under selection, thus we ignore the possible fitness effects of synonymous mutations. Although these might affect transcription and/or translation rates, we ignore these more minor effects to simplify the model.

The system is modelled as a GC with constant volume and constant total concentration of antigen which a number of B cells compete to bind, summarized here and detailed below (figure 3-1). Those B cells that have high affinity BCRs will bind more antigen and these B cells will be more likely to undergo proliferation, while the opposite is true for those BCRs with low affinity. When the binding equilibrium has been reached the progeny distribution for a B cell is evaluated given the amount of antigen bound. Affinity of a progeny cell is a function of the BCR sequence, as defined by the BCR fitness landscape, and after cell division the binding equilibrium

is updated according to the progeny cells and their affinities. Finally when all B cells have been evaluated a new round of selection starts.

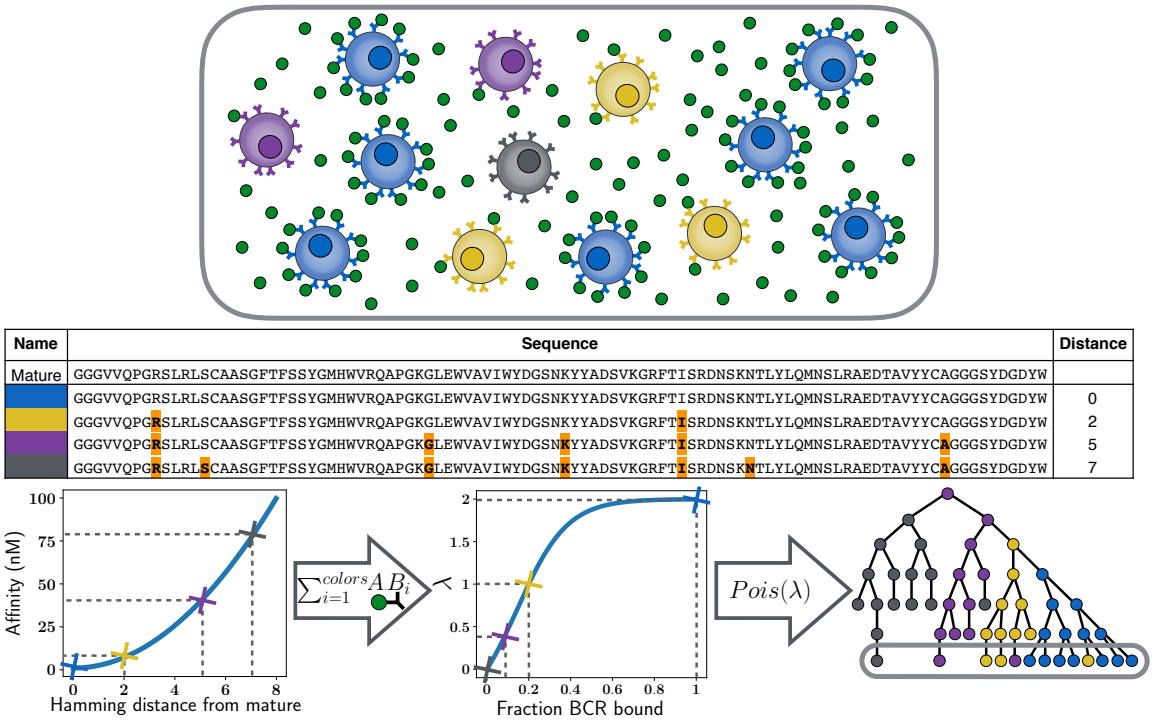


Figure 3-1: Simulation overview. The system is considered as a closed environment with free floating antigen and a number of B cells presenting BCRs on their surface, as illustrated in the top panel. Different colors correspond to different BCR sequences with different affinities. In the middle panel a sequence alignment shows how the different BCR sequences and their distance to the target mature BCR. Third panel shows first how distance from the mature BCR is used to find the affinity. Next affinity of individual BCRs relative to affinity of all BCRs is used to find the fraction of bound BCRs for a given B cell. The fraction bound BCR is then transformed to a λ used in the progeny distribution for the next generation. At the rightmost of panel three, a tree is showing the evolutionary path with an ellipse marking the B cells of the current generation also displayed in the upper part of the figure.

Kinetic model of BCRs binding antigen

Using the most simplistic view, B cells and antigens can be seen as molecules binding and unbinding at some rate intrinsic to the BCR sequences, and their dynamics can then be modelled as a continuously stirred tank reactor (CSTR) [106], widely used in chemical engineering models. The CSTR model makes the assumption that the antigen is spread evenly across the GC and that the binding between BCRs and antigen is at equilibrium at all time. Here we derive the steady state amount of

bound antigen given these assumptions.

First lets consider what should constitute the measure of fitness. Fitness is related to antigen bound and as an alternative to the total amount of bound antigen, we normalize this number between 0 and 1 by using the fraction of BCRs bound to antigen. Considering the BCRs as free molecules with a total concentration of $[B_{\text{total}}]$, the fraction of BCRs bound to antigen at equilibrium is:

$$B_{\text{bound}} = \frac{[AB]}{[B_{\text{total}}]}$$

This B cell specific fitness measure will later be integrated into the simulation of the GC reaction, but for doing this we need to setup the CSTR system and derive its solution.

We are modelling the binding equilibrium between free antigen ($[A]$), the free B cell receptor ($[B]$) and the two bound ($[AB]$):

$$[\text{A}] + [\text{B}] \underset{\text{k}_{\text{off}}}{\overset{\text{k}_{\text{on}}}{\rightleftharpoons}} [\text{AB}] \tag{3.1}$$

The on and off rate of binding is the expressed as constants k_{on} and k_{off}. Affinity is the ratio of substrate and reactants at equilibrium, which is the same as the fraction between on vs. off rate:

$$K_d \equiv \frac{k_{\text{off}}}{k_{\text{on}}} = \frac{[A][B]}{[AB]} \tag{3.2}$$

Isolating $[AB]$ and substituting in the affinity definition K_d from (3.2):

$$[AB] = [B]\frac{[A]}{K_d}$$

Substituting $[B]$ for its expression from mass conservation, $[B_{\text{total}}] = [B] + [AB]$:

$$[AB] = ([B_{\text{total}}] - [AB])\frac{[A]}{K_d}$$

Which rearranges to:

$$[AB] = \frac{[B_{\text{total}}]}{1 + \frac{K_d}{[A]}}$$

Extending to multiple BCR affinities:

$$\begin{aligned}
[\mathrm{A}] + [\mathrm{B}_1] &\underset{\mathrm{k}^1_{\mathrm{off}}}{\overset{\mathrm{k}^1_{\mathrm{on}}}{\rightleftharpoons}} [\mathrm{AB}_1] \\
[\mathrm{A}] + [\mathrm{B}_2] &\underset{\mathrm{k}^2_{\mathrm{off}}}{\overset{\mathrm{k}^2_{\mathrm{on}}}{\rightleftharpoons}} [\mathrm{AB}_2] \\
&\vdots \\
[\mathrm{A}] + [\mathrm{B}_n] &\underset{\mathrm{k}^n_{\mathrm{off}}}{\overset{\mathrm{k}^n_{\mathrm{on}}}{\rightleftharpoons}} [\mathrm{AB}_n]
\end{aligned}$$

Affinity is a constant for each BCR and since all B cells compete for the same antigen, each $[AB_i]$ is dependent only through the concentration of unbound antigen:

$$\begin{aligned}
[AB_1] &= \frac{[B^1_{\mathrm{total}}]}{1+\frac{K^1_d}{[A]}} \\
[AB_2] &= \frac{[B^2_{\mathrm{total}}]}{1+\frac{K^2_d}{[A]}} \\
&\vdots \\
[AB_n] &= \frac{[B^n_{\mathrm{total}}]}{1+\frac{K^n_d}{[A]}}
\end{aligned}$$

Now introducing mass conservation for A:

$$A_{\mathrm{total}} = [A] + \sum_{i=1}^{n} [AB_i] \equiv [A] + \sum_{i=1}^{n} \frac{[B^i_{\mathrm{total}}]}{1 + \frac{K^i_d}{[A]}} \tag{3.3}$$

By rearranging to a polynomial form the system can be solved to find B_{bound} for each B cell. A solution will be elaborated in the implementation section.

Transforming distance to target to affinity

Next we describe how to simulate the affinity (K_d) of the BCRs. Potentially the affinities can be generated in any imaginable way, by having a function transforming a BCR sequence (Bseq) into a number that represents affinity. Formally this would be a function: $\mathrm{Affinity}(\mathrm{Bseq}_i) = K^i_d$. In a realistic, yet still minimalistic, model we would imagine that the BCRs in a GC were evolving towards a specific target sequence denoted Tseq. A target is the sequence with the highest affinity, the state where fitness plateaus out. We will define a fitness landscape around this plateau using a distance function, $\mathrm{dist}(\cdot, \cdot)$, which we assume to be given as the Hamming distance between amino acid sequences.

Let us define the affinity of the naive input sequence as K^{naive}_d and correspondingly

affinity for the mature target sequence, K_d^{mature}. Now we can define an arbitrary function with reference points in K_d^{naive} and K_d^{mature}, that transforms a distance between Bseq and Tseq to an affinity:

$$\text{Affinity}(\text{Bseq}, \text{Tseq}, K_d^{\text{naive}}, K_d^{\text{mature}}) = K_d^i$$

There are two conditions we want to impose. If the BCR sequences is equal to the naive sequence (Nseq), then it takes the affinity of the naive, and if it is equal to the mature target sequence (Tseq), it takes the affinity of the mature:

$$\begin{aligned} \text{Affinity}(\text{Nseq}, \text{Tseq}, K_d^{\text{naive}}, K_d^{\text{mature}}) &= K_d^{\text{naive}} \\ \text{Affinity}(\text{Tseq}, \text{Tseq}, K_d^{\text{naive}}, K_d^{\text{mature}}) &= K_d^{\text{mature}} \end{aligned} \tag{3.4}$$

These conditions are satisfied by:

$$\text{Affinity}(\text{Bseq}, \text{Tseq}, K_d^{\text{naive}}, K_d^{\text{mature}}) = K_d^{\text{mature}} + \left(\frac{d}{t_{\text{naive}}}\right)^k (K_d^{\text{naive}} - K_d^{\text{mature}}) \tag{3.5}$$

Where $d = \text{dist}(\text{Bseq}, \text{Tseq})$ is the distance between BCR and target sequence and $t_{\text{naive}} = \text{dist}(\text{Nseq}, \text{Tseq})$ is the distance between naive and target sequence. The exponent, k, can be chosen to adjust the mapping between distance and affinity, with the restriction that $0 < k < \infty$ (figure 3-2).

It is easy to imagine that in a real affinity maturation process there will be many different BCR sequences that are practically equally fit. E.g. this will happen when multiple amino acids are equally fit on a given position. It will also happen if there are multiple distinct maturation paths that ends up with equally fit BCRs. The later effect can be introduced into the model be adding more target sequences and then determining the affinity based on the shortest distance to all target sequences:

$$d = \min_{\text{Tseq} \in \text{Targets}} \text{dist}(\text{Bseq}, \text{Tseq})$$

Transforming BCR occupancy to fitness

In the affinity model the fitness of a B cell is determined by the amount of antigen it has bound divided by the total number of receptors it has, we shall call this B_{bound}^i for the BCR B_i. The progeny distribution should adjust according to B_{bound}^i, so if B_{bound}^i is small the progeny distribution should favor terminating the B cell and opposite, if B_{bound}^i is large this should cause the progeny distribution to favor cell division. The Poisson distribution will reflect this behaviour by setting λ_i small when B_{bound}^i is

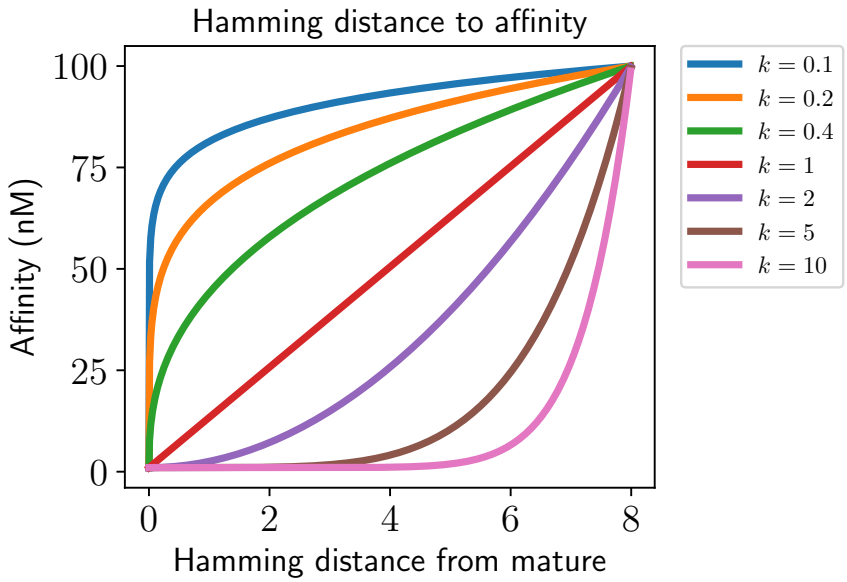


Figure 3-2: Varying the exponent k in (3.5) to achieve different mappings between distance and affinity. Naive and mature affinity is held constant, $K_d^{\text{naive}} = 100nM$ and $K_d^{\text{mature}} = 1nM$.

small and λ_i large when B^i_{bound} is large. To use $\text{Pois}(\lambda_i)$ as the progeny distribution we need a function transforming B^i_{bound} to λ_i: $Y(B^i_{\text{bound}}) = \lambda_i$. For this purpose we can use a generalized version of the logistic function since this has the properties we need:

$$\lambda_i = Y(B^i_{\text{bound}}) = \alpha + \frac{K - \alpha}{G + Q\exp(-\beta B^i_{\text{bound}})} \tag{3.6}$$

G is chosen to be the typical value of 1. K is the upper bound of the function and is set to 2, reflecting that the fastest average growth rate is 2^t, with t generations (setting $\max(\lambda) = 2$ is a model choice, but could be changed). α, β, Q and U are fit to fulfill three conditions:

$$Y(0) = 0, \quad Y\left(\frac{f_{\text{full}}}{U}\right) = 1, \quad Y(f_{\text{full}}) = 2 - \epsilon \tag{3.7}$$

The solution is undefined in $Y(f_{\text{full}}) = 2$ because the function is asymptotically growing towards 2, therefore ϵ can be regarded as a small value (e.g. 10^{-3}) so that $Y(f_{\text{full}}) \approx 2$ but having a defined solution. $0 < f_{\text{full}} \leq 1$ is the fraction of bound BCRs (B^i_{bound}) sufficient to make the B cell fully activated, ignoring the negligible contribution of ϵ. At f_{full} the highest fitness is achieved and binding more antigen

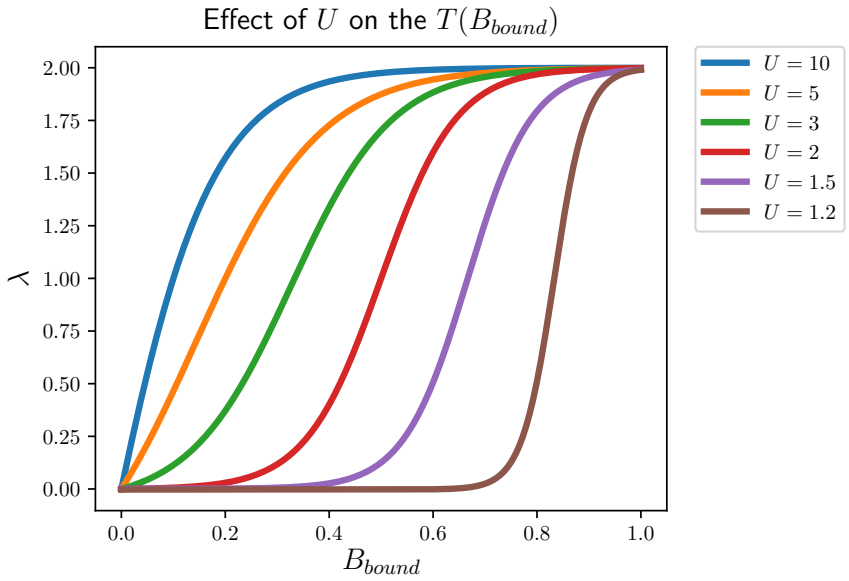


Figure 3-3: Using a constant $f_{\text{full}} = 1$, changing the U parameter in the conditions in (3.7) to achieve a shift of the inflection point, at $\lambda = 1$, on the B_{bound} axis.

will not change the progeny distribution. The constant $U > 1$ in condition 2 can be adjusted to set the value of B^i_{bound} resulting in $\lambda_i = 1$. The interpretation of U is that it is the fraction of BCRs binding antigen necessary to sustain the life of the B cell, but nothing more or less. Using these conditions α, β and Q can be found and the logistic function is fully defined. α can be interpreted as the lower asymptote of the function. β is the steepness of the function, and it is coupled to the Q parameter, which will follow it according to the three conditions in (3.7).

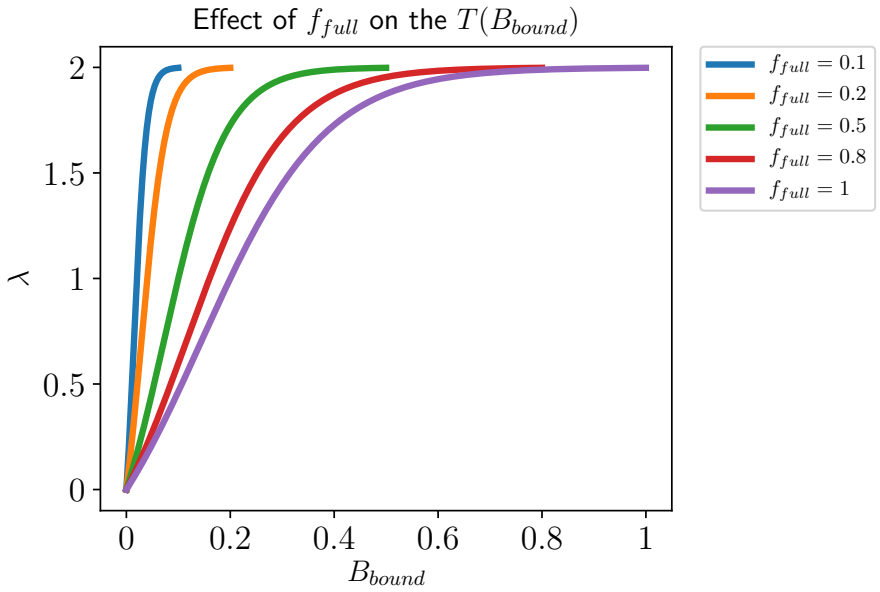


Figure 3-4: Using a constant $U = 5$, changing the f_{full} parameter in the conditions in (3.7) to change the point where B_{bound} reaches the largest λ.

The carrying capacity of a GC

Finally we need to introduce the concept of a carrying capacity of a GC, which is defined as the number cells a GC is able to support in its micro environment. The carrying capacity is determined mainly by the total concentration of antigen, since binding to antigen controls the progeny distribution. BCR affinity is also influencing antigen binding, and therefore must influence the carrying capacity, but at high affinity nearly all antigen is bound, and hence the total antigen concentration is the most influential determinant of GC carrying capacity. At Pois(1) the progeny distribution is only just sustaining the population size of the GC, and given from condition 2 in (3.7) this happens at $\frac{f_{\text{full}}}{U}$. Under the assumption that the population of B cells all have identical BCR sequences, the maximum carrying capacity is:

$$C([A_{\text{total}}]) = U\frac{[A_{\text{total}}] - [A]}{f_{\text{full}}} \tag{3.8}$$

The concentration of unbound antigen is determined by the affinity and concentration of BCRs. It is fair to assume that there are many more BCRs than antigens, so for high affinity BCRs, the majority of antigen should be in a bound state allowing for the approximation of setting $[A] = 0$. This makes it easy to calculate the carrying capacity given (3.8). However in the cases with low affinity the concentration of free antigen cannot be assumed zero and in such cases $[A]$ can be determined through

(3.3).

In the situation where a newly arising mutant has higher affinity than the rest of the population the fraction of BCRs binding antigen on this mutant will approach f_{full}, resulting in a high probability that the clone will expand rapidly to overtake the GC population, also known as a clonal burst. The clonal burst has the characteristic time (average time of take over) being:

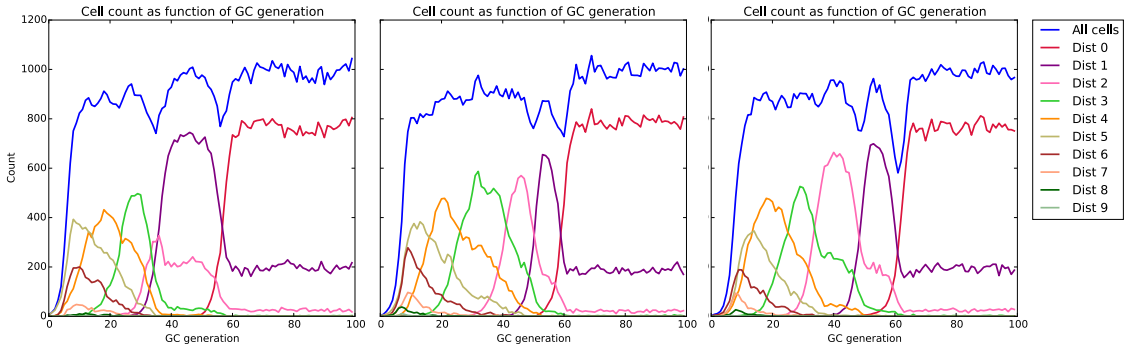
$$T_{\text{burst}} = \log_2 (\text{carrying capacity}) = \log_2 \left(U \frac{[A_{\text{total}}] - [A]}{f_{\text{full}}} \right)$$

3.2.3 Parameter choice

We chose the parameter U in the logistic transformation to take a value to reflect our belief of how the shape of this transformation should look. Our expectation is that initially, when only a few BCRs are bound and stimulation is low, there will be a linear increase of the stimulus with increasing antigen binding. At some point close to f_{full}, this increase in stimulus should level out because the cell can only be stimulated to $\max(\lambda) = 2$. This expected shape is recapitulated by a $U = 5$, see figure 3-3, and therefore this is where we fix U.

The choice of f_{full} determines the BCR occupancy when full activation of proliferation is achieved. f_{full} does not have any known reference value so it is chosen to take a value of 1 because this is mathematically convenient. It turns out that the model is quite robust to different choices of f_{full} and it causes no substantial effect to change it from 1 to 0.05, see figure 3-5. Changing f_{full} will primarily effect the carrying capacity via. (3.8), but by adjusting $[A_{\text{total}}]$ to get the same carrying capacity simulations are indifferent to the choice of f_{full}. Presumably this is because the shape of the logistic transformation $Y(\cdot)$ in (3.6) will remain intact, see figure 3-4. The choice of $f_{\text{full}} = 1$ has the interpretation that when $\frac{1}{U}$ of the BCRs on a B cell are binding antigen it has a Pois(1) progeny distribution and when all BCRs are binding antigen, the progeny distribution increases to Pois$(2 - \epsilon)$.

In the transformation from distance to affinity in (3.5), we have to make a choice about which exponent to use. There is reason to believe that not all mutations are improving affinity with equal proportion and therefore the linear transformation is excluded. Another thing we would like to enforce is to disallow sequences to drift far away from both the mature and naive sequence. A large sequence drift should not be very likely since it would complete abolish the binding of antigen. For this reason it is preferred to have an exponent $k > 1$. We choose $k = 2$ since this puts an extra penalty on simulated sequences that are drifting, without putting an excessive emphasis on the first two steps of approaching the mature sequence, see 3-2.



(a) (b) (c)

Figure 3-5: Simulation with affinity selection for varying magnitudes of f_{full}. $f_{\text{full}} = 1$, $f_{\text{full}} = 0.5$, $f_{\text{full}} = 0.05$ for (a), (b) and (c) respectively. Simulations with $U = 5$ and $[A_{\text{total}}]$ adjusted to obtain a carrying capacity of 1000 cells. Each simulation is run for 100 generations with $t_{\text{naive}} = 10$ and the composition of sequence distances to their closest targets are plotted for each generation.

The logistic function is chosen to yield a maximum λ of 2 since this would correspond to an average max progeny of 2 cells yielding a population exponentially increasing by 2^t, which is a standard exponential population growth. One useful feature of the logistic function is that it has a notion of maximum signal i.e. when more antigen binding does not give more signal. This flattening out is asymptotic approaching 2 which mean that the maximum growth rate of $\lambda = 2$, is not reached completely at f_{full}, so to fit the function we have to allow for this by choosing a small ϵ that makes $Y(f_{\text{full}})$ practically equivalent to 2. Here we choose $\epsilon = \frac{1}{1000}$ meaning that $Y(f_{\text{full}}) = \frac{1999}{1000}$.

Next we need to find some realistic numbers for the constants such as affinity, carrying capacity and B^i_{total}. First lets consider affinity. K_d for a naive sequence is likely in the low micro molar range range of $10^{-6} - 10^{-7}M$, while the mature affinity is in the nano or subnano molar range of $10^{-8} - 10^{-10}M$ [6] (M is used to denote molar concentration). Classically these affinity measurements have been done using hapten induced immunizations, and there is reason to believe that conclusions about affinities will be different for other antigens. Indeed it has been reported that naive sequences in some cases can be high affinity binders and affinity maturation is less useful [28]. Nevertheless several groups have reported 50-100 fold affinity increase due to affinity maturation in both complex and hapten induced immunizations [49], [69], [99]. We choose the naive sequence to be $10^{-7}M$ ($100nM$) and the mature to be $10^{-9}M$ ($1nM$), giving a large span in affinity to select on.

With the introduction of K_d's we need to consider units. Lets start by estimating

the concentration of BCRs per B cell in the GC, also denoted as B^i_{total}. To simplify things we assume that the number of BCRs on each B cell is the same. In each GC there is an estimated 4×10^3 B cells [48], but it has before been estimated to just 1000 [11] which we will use for convenience. On the surface each of these B cells have an estimated 10^4 BCRs [77], [22] (highly uncertain estimate), resulting in 10^7 BCRs per GC. Converting this to moles gives: 10^7 6×10^{-23}moles The GC is reported to have a diameter of $10^{-4}m$ [81]. Assuming the shape is spherical and converting to liters gives:

$$\frac{4}{3}\pi \left(\frac{1}{2} \times 10^{-4}m\right)^3 10^3 \frac{L}{m^3} = \frac{1}{6}\pi \times 10^{-9}L$$

Finally this gives:

$$\frac{10^7\ 6 \times 10^{-23}\text{moles}}{\frac{1}{6}\pi 10^{-9}L} \approx 10^{-6}M \equiv 10^3 nM$$

This means that each B cell contributes approximately $1nM$ to the total concentration of BCRs. Now to simplify things, everything is normalized to nano molar so e.g. the naive/mature affinity is changed to $100nM$ and $1nM$ respectively. All the above described values are tabulated in table 3.2 and used as constant in later simulations of sequences undergoing affinity selection.

Constant	Value	Description	Reference
B^i_{total}	1×10^4	Number of BCRs on each B cell	[22], [77]*
n_t	1000	B cells per GC	[48], [11]
d	$10^{-4}m$	GC diameter	[81]
$\frac{1}{U}$	$\frac{1}{5}$	Fraction of f_{full} necessary to sustain the population	See text
k	2	Exponent of affinity transformation	See text
f_{full}	1	Fraction BCR bound at full response	See text
K_d^{naive}	100nM	Naive affinity	[6]
K_d^{mature}	1nM	Mature affinity	[6]

Table 3.2: Constants used in the model of affinity selection. *There is a lot of uncertainty in this number and depending on the method it is estimated from 10^3 to 10^7.

3.2.4 Implementation

In the solution to the BCR antigen binding equilibrium in (3.3) B cells having the same BCR sequence are included in the same B^i_{total}, and the hence the B^i_{total} will be different when some B cell genotypes are highly abundant while others are less. To simplify bookkeeping now let's expand the index, i, to represent just a single B cell, resulting in the simplification that $[B^1_{\text{total}}] = [B^2_{\text{total}}] = \ldots = [B^i_{\text{total}}]$. In

this notation multiple B cells can have the same BCR sequence, but each will have their own index because they are associated with different B cells. This simplifies bookkeeping because the BCR index is always the same as the B cell index. The change does not affect the solution in (3.3). Now given some total concentration of antigen, A_{total}, the solution to the concentration of free antigen at equilibrium, $[A]$, reduces to finding the real positive root of the polynomial in (3.3). Finding this root is easily be done using Newton's method, or by using a faster method like the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm [88]. Once its quantity is found, we can obtain the concentration of bound antigen by plugging $[A]$ into (3.2.2). Lastly B_{bound} is converted to a progeny distribution parameter λ by using (3.6).

All the above model definitions have the single purpose of adjusting the progeny distribution of a cell through λ_i given some fitness function. Under the neutral model this fitness function is constant, resulting in the same λ_i for all cells while in the affinity selection model λ_i are updated to reflect B cell fitness in a GC reaction. Practically the simulation of sequences can be thought of as a process of constructing a phylogenetic tree, see figure 3-6. In each generation an integer is drawn from each of the progeny distributions for all non terminated leaves on the tree. The leaf can then either terminate or have 1 to $N \in \mathbb{Z}_+$ progeny cells. If the leaf has progeny each will undergo their own mutation process following the S5F model and become leaves in the next generation. One generation is defined as one iteration through all the non terminated leaves on the tree and this is done in random order to avoid any bias. Once every leaf has be evaluated this marks a new generation and the generation time is increased by one. For an overview see figure 3-6.

Using tree terminology, the progeny distribution of a leaf depends on all the states (i.e. sequences) of the non terminated leaves. Then by definition the affinity model needs to be updated, by re-evaluating (3.3) and finding all $[AB]_i$, every single time a new leaf is generated, creating a computational burden at large population sizes. E.g. when simulating at a carrying capacity of 1000 cells, the vast majority of the computations are spent updating $[AB]_i$. An approximate solution is simply to skip some of these updates and rely on the previous determination of λ_i, as an example a system with 1000 cells could be update only at every 100th cell progeny evaluation, resulting in 10 evaluations per generation. Regardless of whether evaluations are skipped, all λ_i's will be updated when a new generation is started. The rationale behind skipping evaluations is that when there is already a large population of cells, then very little will change after a single leaf is evaluated. In fact it turns out that for simulations using a carrying capacity of 1000 cells, there are no distinct difference between updating $[AB]_i$ after every leaf is evaluated or once every 100th leaf progeny

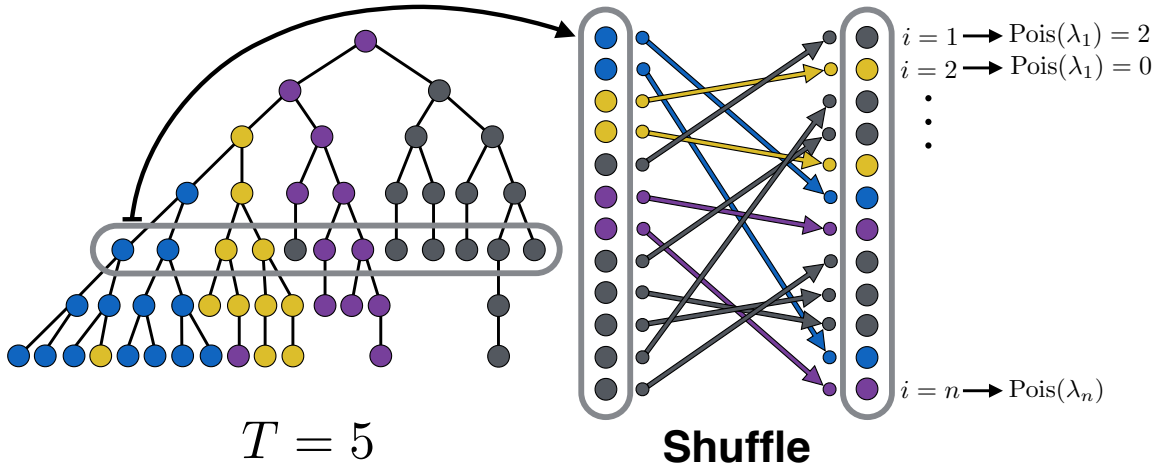
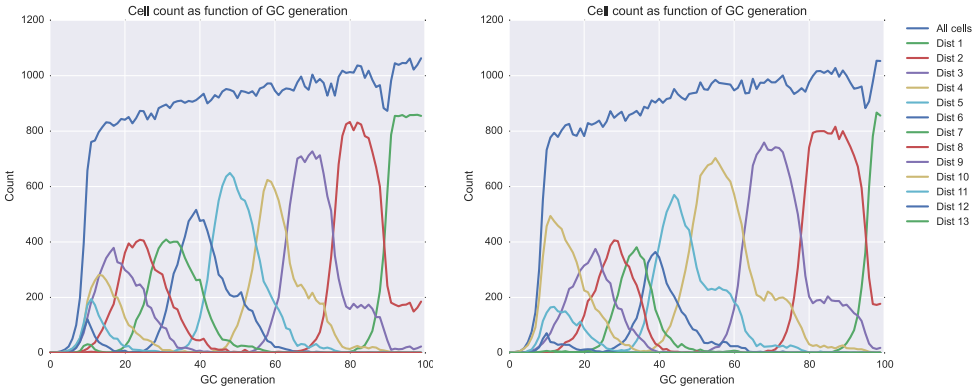


Figure 3-6: Illustration the sampling procedure in a time slice ($T = 5$) of the simulation of a phylogeny undergoing affinity selection. A generation time is defined as the time when all nodes have been sampled and their progeny have been evaluated. At each generation all non-terminated nodes will be evaluated in random order. For neutral selection λ_i is constant and identical for all cells. For simulation with affinity selection λ_i is B cell dependent and re-evaluated every time there is a change in the population of non-terminated nodes.

evaluation, see figure 3-7.



(a) (b)

Figure 3-7: Simulation with selection comparing (a) with and (b) without skipping recalculation of λ_i at each cell evaluation. In (a) no steps are skipped while, in (b), 99% of all recalculations are skipped (10 updates to a population of 1000 B cells). Simulation parameters are default as in table 3.3, with $\lambda_{\text{mut}} = 0.3$ and $T = 100$.

Target sequences ($\text{Tseq}_i, i = 1, 2, \ldots, t_n$) can be arbitrarily defined by inputting a list of amino acid sequences. Since affinity selection works on the protein level the distance that determines affinity is the Hamming distance between two amino acid

sequences. Either one or multiple target sequences can be used but all are assumed to have equal affinity, K_d^{mature}. In the tests presented here we input the number of targets (t_n) and the amino acid distance from naive seed to target (t_{naive}). Using these parameters the target sequences are simulated by introducing DNA mutations into the naive sequence until it has diverged t_{naive} away from its starting point. The mutations are introduced at DNA level given the neutral branching process described previously and thereby a distance of $t_{\text{naive}} = 10$ does not always correspond to 10 mutations, often the number is higher due to accumulation of synonymous mutations not counting towards protein level distance. The process is repeated until t_n targets have been made. We reason that a good default choice of t_{naive}, to achieve sufficient evolutionary distance, is 10, however the simulation behaviour seems rather indifferent to this, compare 3-5 with 3-7. The default number of targets (t_n) is set to 100 to provoke epistatic effects. All default parameters in the affinity simulation is tabulated in table 3.3. The implementation is available as a simulation subprogram in the codebase of `GCtree` (`github.com/matsengrp/gctree`).

Parameter	Default	Description
λ_{mut}	None	Pois(λ_{mut}) sequence mutability
T	None	Stopping time
cap	1000	Carrying capacity
t_n	100	Number of random target sequences
t_{naive}	10	Distance from naive to target sequence
n	all	Down-sampled number of sequences

Table 3.3: Default parameters used in the affinity selected simulations.

The epistatic effects of having multiple mature targets

A BCR is a highly dynamic structure that can bind a single antigen in many different ways, it is therefore also likely that the GC maturation would end up producing different BCRs if replicating the process. This can be thought of as a multimodal fitness landscape, and it can be emulated by introducing multiple targets for the affinity simulation. Multimodality in the fitness landscape results in epistasis, which is defined as non additive interaction between mutations, and is widely observed in nature e.g. the evolution of influenza nucleoprotein [35]. The manifestations of epistasis can be different, but here is a simple example where the fitness is shown for various genotypes:

$$ab = 1,\ Ab = 1,\ aB = 1,\ AB = 10$$

There are four different genotypes, based on two loci (positions) with two alleles (gene variants), three having a fitness of 1 and one having a fitness of 10. Each position is tolerable to the two states, but only a combination of state A and B improves fitness. In a linear additive, non epistatic, model setting the effect of the intermediate states (Ab and aB) should sum to the effect of the full state:

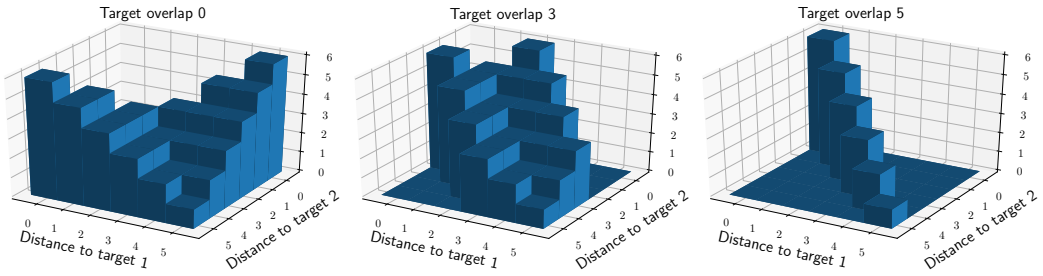
$$ab + (Ab - ab) + (aB - ab) = AB$$

This is just one example of epistasis, where the effect is AND gate like. The effect could also confer deleteriousness to the intermediate:

$$ab = 1,\ Ab = \frac{1}{2},\ aB = \frac{1}{4},\ AB = 10$$

Or any other non additive influence.

In the affinity simulation there is inherent epistasis because the transformation from distance to affinity is non-linear, although we could choose to fix $k = 1$ in (3.5) to obtain linear additive distance effects. Regardless, if simulation is done using multiple different targets it will be epistatic. The shape of the fitness landscape, i.e. the distance between modalities, depends on the distance between targets. Most likely, when two targets are generated, they will be $2 \times t_{\text{naive}}$ apart, but sometimes they have overlapping substitutions squeezing modalities together as illustrated in figure 3-8.



(a) (b) (c)

Figure 3-8: The effect of having two target sequences on the fitness landscape. Two target sequences are created with varying overlap using $t_{\text{naive}} = 5$. The fitness landscape is constructed using a linear distance to affinity function ($k = 1$ in (3.5)). In a) no overlap makes a long distance between the two fitness peaks, in b) peaks are getting closer when targets overlap, and in c) when the overlap is complete the two targets match and the system no longer is epistatic, under a linear distance to affinity function.

Indeed the effects of epistasis can be observed in simulations with multiple targets.

Often sequences are evolving towards a single target and once a few mutations have been accumulated a sequence is "committed" to this evolutionary trajectory. However trajectories can change when a few mutations coincide with another target as observed in figure 3-9. These observations supports the view that the presented affinity simulation is a convoluted model being a good challenge to test the assumptions of inference methods.

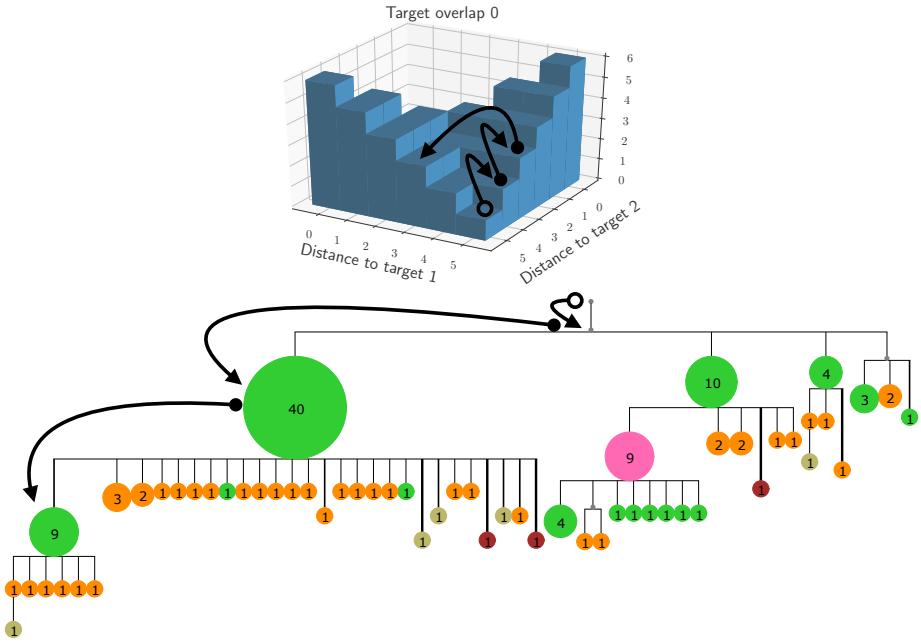


Figure 3-9: Example of epistasis in a simulation run with multiple target sequences. Colors correspond to the affinity of the simulated cells, see figure B-2 in appendix B for run stats. Arrows show an evolutionary trajectory from a low levels in the fitness landscape (starting at the unfilled black circle) to a higher level. Zero amino acid distance branches are collapsed and values inside nodes correspond to the number of B cells. Here we see that the simulation trajectory is following along several targets. There is even a jump between two target sequence trajectories, with the highest frequency node (green 40) yielding a descendant with two amino acid mutations (green 9) that is equally close to another target, resulting in a change in mutational trajectory.

3.3 Results

To test the simulation protocol and whether it recapitulates real world affinity maturation we needed a dataset with a known phylogeny starting from the naive sequence as a root node. It is practically impossible to get such a dataset. However, one of

the current best single-GC data sets was made by Tas et al. [96] with single cells sequencing of B cell isolated from the same GC. The largest Tas et al. clonal family consists of 65 BCR sequences. Some sequences appear in multiple B cells and these are deduplicated and assigned abundances leaving a total of 42 different genotypes as observed in figure 3-10. As a reminder, the phylogeny of the Tas. dataset is *not* known but inferred based on a likelihood ranking of equally parsimonious trees (unpublished data).

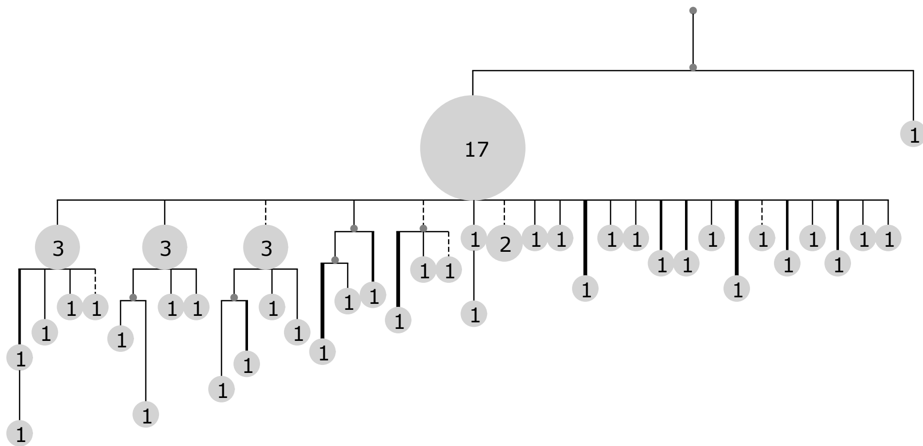


Figure 3-10: Inferred phylogeny for the single GC dataset from Tas et al. [96]. The inference method used is based on likelihood ranking of equally parsimonious trees, unpublished but implemented in the `GCtree` source code as a subprogram. Figure credit William S. DeWitt.

We would like to tune our simulations so as to match real data sets as much as possible. First we need a measure of accumulated SHM, i.e. what is the percentage of mutations in the GC sequences, and how is it distributed across the length of the tree. We plot this mutation distribution as an empirical cumulative distribution function (CDF), see a) in figure 3-11. Next, genotype abundance is an important trait and indicator of clonal bursts. Higher abundance clones are assumed to be more fit and should therefore also yield more offspring. Some offspring will have a slightly different genotype and correspondingly different fitnesses, hence high abundance clones should also have many different genotype descendants. A proxy for counting these descendants is to count the immediate neighbors being just a single Hamming edit away, see b) in figure 3-11. If the assumption holds there should be a positive correlation between abundance and Hamming neighbors.

In figure 3-11 plotting the two above mentioned measures for the Tas dataset, and superimposing the same measures but for 100 simulation runs, shows a consistent good fit between simulations and real data. For this run, parameters were set to $n = 65$, $\lambda_{\text{mut}} = 0.25$, $t_{\text{naive}} = 5$, $T = 35$ and default otherwise. The down-sampling

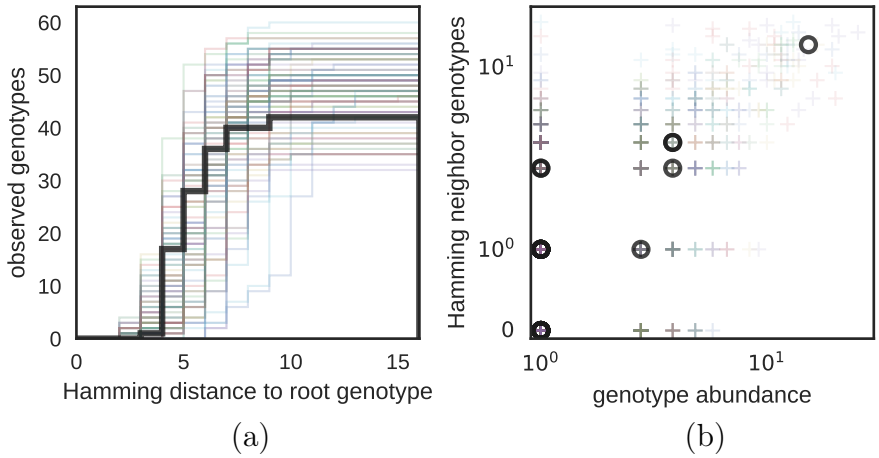


Figure 3-11: Summary statistics for 100 simulations using $\lambda_{\text{mut}} = 0.25$, $t_{\text{naive}} = 5$, $T = 35$ and $n = 65$. Simulations are colored and the Tas dataset is black. In a) the cumulative distribution of mutations (empirical CDF) and b) the number of genotypes in 1 Hamming distance away as function of genotype abundance.

parameter (n) was set to the same number as sampled B cells in the Tas dataset. The seed naive sequence used was a V gene of 264 nt; using the commonly cited SHM rate of 10^{-3} [103] this gives $\lambda_{\text{mut}} = 0.264$ which was rounded to $\lambda_{\text{mut}} = 0.25$. Target distance ($t_{\text{naive}}$) and simulation time ($T$) was adjusted so the simulated sequences had approximately the same minimum Hamming distance to the naive sequence.

During affinity simulations the evolution of the cell population was plotted showing the emergence of new clones with higher affinity and their gradual take over of the cell population until an even more fit clone emerged (see appendix B figure B-1). In all cases of affinity simulation there is a clear progression towards higher affinity as time passes until eventually a cell has reached the target sequence with highest affinity. Topologically simulated trees also capture this notion of clonal bursts with one genotype suddenly being very dominant and yielding many offspring (figure 3-12).

3.4 Discussion and conclusion

Previously many groups have made models of the GC reaction and affinity maturation [76], [87], [10], [105], [11], [78], but none of these have included a definition of the BCR sequences on DNA level. Neutral branching processes can easily be setup to simulation sequences undergoing the same mutational patterns as real BCR sequences, but they have no way of capturing the important clonal bursts happening due to the fitness gain from mutating to a higher affinity BCR. In this work we have addressed this problem through a simple, but yet unexplored way, of integrating affinity related

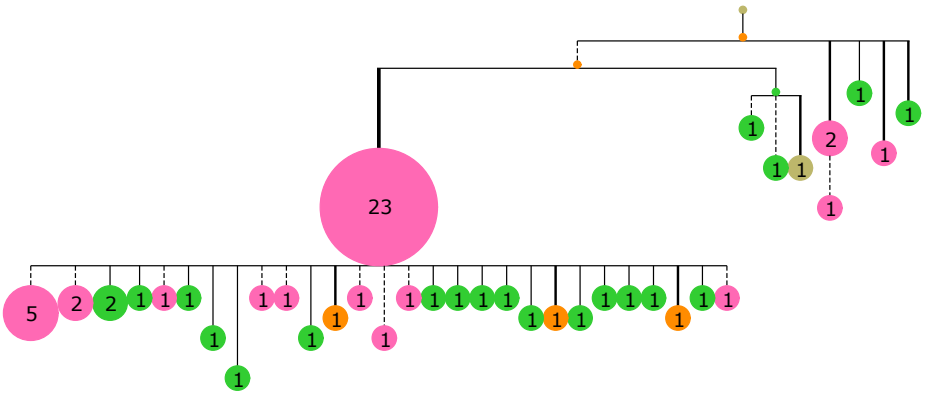


Figure 3-12: Simulated tree using $\lambda_{\text{mut}} = 0.25$, $t_{\text{naive}} = 5$, $T = 35$ and $n = 66$. For simulation statistics and color to affinity mapping see appendix B figure B-1.

fitness into the simulation of BCR sequence evolution. We hope that the simulation framework will be a valuable tool in the assessment of inference methods.

It is interesting to see the similarities between the tree topologies of an affinity simulated tree compared to the inferred tree for the Tas et al. dataset. As it was also noted in the work of Tas et al. [96] this GC has undergone a clonal burst with an apparent dominant genotype (abundance 17 in figure 3-10) having many slightly mutated offspring. Clearly this topological trait is very similar to the affinity simulated tree in figure B-1, and by mapping affinity to each node it can be seen that the clonal burst happened as a result of a mutation causing higher affinity. However, as Tas et al. also observed, the appearance of a high affinity genotype does not guarantee a clonal burst, and as expected this is also true in the affinity simulation. Under the best conditions a B cell has $\lambda = 2$, from which it follows that $\text{Pois}(\textit{termination}|\lambda = 2) = 0.135$, so there is roughly a 14% chance of a high affinity clone turning extinct.

The distribution of clones over time (e.g. see appendix B figure B-1) reveals that mutations conferring a fitness advantage through higher affinity are quickly found and can quickly overtake the whole GC population. If this represents the events of real affinity maturation it poses the problem, that the probability of completing a maturation trajectory will be defined by the steepness of the fitness function, and not the fitness at the end of a trajectory. This would be a way of leading maturation into a dead end, far lower in fitness than the global optimum, but too high to be reverted backwards towards the naive sequence. Indeed such a mechanism is a reasonable, although not the only, explanation why broadly neutralizing HIV antibodies are so rare, while their typical germline usage is normal [85].

Chapter 4

Ancestral sequence reconstruction of the B cell receptor phylogeny

4.1 Introduction

With the wide availability and cost reductions of high-throughput sequencing (HTS) it is getting commonly applied for studying the immune response through sequencing of the BCR and TCR repertoire. Phylogenetic reconstruction of the evolution of the BCRs have recently gotten increasing attention because of its possible use in studying vaccine response. There is a hope that with better understanding of the evolutionary trajectories of BCRs that the fate of the immune response can be modelled in a probabilistic framework, and that this will lead to new avenues in vaccine design finally addressing vaccination of HIV and broad influenza immunity.

By and large the use of phylogenetics in BCR GC evolution has been made with the standard assumptions of site independence, constant mutation rates etc. using methods like maximum parsimony [96], [4] and maximum likelihood [17], [40]. It is important to note that most algorithms for phylogenetic inference have been made to study population genetics largely evolving under a neutrality or track the evolution of organisms over millions of years. The consequences being that model assumptions have been formulated and validated in a completely different context than the highly selective, small population evolution that BCRs undergo in the GC reaction. In contrast, BCR evolution has some interesting features that violates the assumption of classical phylogenetic methods e.g. the mutation model is DNA context sensitive, the root of the tree is known, high selection pressure on selected sites, sampled ancestors, very high mutation rates etc. We are not aware of any study that does a broad comparison of these commonly used phylogenetic inference tools in a parameter regime

relevant to BCR sequence evolution.

Validation studies are needed in order to understand weaknesses in the existing phylogenetic tool-chain, and opportunities to develop tools specific to the BCR case. Sequence simulation is the cornerstone of all validation studies but unfortunately there has been no publicly available method for simulation of BCR sequence evolution. Multiple articles have described simulations of VDJ recombined sequences [84], [73], [83] and few have also described simulation of B cell maturation [90], [47]. Recombination centric simulations are usually not using a realistic model for simulating the SHM induced mutations in the GC reaction because their purpose is to test VDJ inference, and the few simulation programs emulating maturation are either closed source, does not model central parts of the GC reaction or entirely avoids dealing with sequences. We are using a branching process, with and without selection, that is developed specifically for modelling the outcome of SHM in the GC reaction to test BCR lineage tree reconstruction under complex evolution.

In this paper we benchmark the performance of classical phylogenetic tools when applied to B cell receptor sequences, including tools for phylogenetic tree inference and for ancestral sequence reconstruction (ASR). We do so in a realistic sequence simulation framework that can be used in the future for validating BCR specific methods as they develop.

4.1.1 Ancestral sequence reconstruction

When a phylogeny is inferred using a tree as a model for evolution, the tree will have internal nodes connecting the observations. An internal state is also the inferred common ancestor of a number of leaves and/or branches and therefore sometimes referred to as an ancestral state. Internal nodes are typically unknown, unobserved states, either explicitly defined by a sequence as in a parsimony tree, or defined by a likelihood function in model based phylogenies. When a likelihood function is used there is no longer a well defined ancestral sequence, instead this needs to be inferred as the maximum likelihood (ML) sequence estimate. The ML estimate can be defined in two ways, as either marginal or joint ML reconstruction.

In a marginal reconstruction the ML estimated sequence for one node is independent of the estimate of another i.e. $a_1 = \max(l(a_1|t)), a_2 = \max(l(a_2|t)), \ldots, a_n = \max(l(a_n|t))$, where l is the likelihood function given a tree with branch length and data, t, and a_i is an ancestral sequence on internal node i. This makes it easy to find ancestral sequences by iterating through all internal nodes in arbitrary order.

However, once an ancestral sequence has been fixed to an internal node this changes the likelihood function for the rest of the tree. In fact all internal states

are interdependent and while this is ignored in the marginal reconstruction it is taken into account in joint reconstruction by maximizing the likelihood of all ancestral states at once i.e. $a_1, a_2, \ldots, a_n = \max(l(a_1, a_2, \ldots, a_n|t)), a_2$. It is a non trivial task to minimize this likelihood for many internal nodes, and though Pupko et al. [71] showed how to speed up the joint reconstruction algorithm, most software uses the more simple marginal reconstruction.

While it is clear that joint reconstruction is the more probabilistically correct way of inferring ancestral sequences there is less clarity about whether or not this changes the estimated sequences. The model based methods tested in this work are all using marginal reconstruction and therefore this will also be used in our validation.

4.2 Methods

4.2.1 Measuring correctness of ancestral reconstruction

In the following section we will introduce a benchmark metric for ancestral sequence reconstruction, we call this metric correctness of ancestral reconstruction (COAR). The correctness of a reconstruction compared to the true evolutionary history can be measured by multiple similarity measures e.g. a) topological similarity, b) branch length similarity and c) sequence similarity between inferred and real ancestors. All these measures are inter-dependent e.g. the inferred sequences are affected by the branch lengths and the topology and the branch lengths are determined given a topology etc.

Assume the loss function for ancestral sequence reconstruction is comprised of the three above mentioned terms. The tree topology is the model framework of the phylogeny by which we can extract useful information like relatedness, ancestral sequences, distances and more. In a model based phylogeny branch lengths are a measure for the expected number of substitutions per site so picking the correct branch lengths are important for reconstruction. If the model underlying the phylogeny is a clock-model, where the branch lengths are related to evolutionary time, then the branch lengths are also interpretable, but otherwise interpretation is more difficult, and regardless of their magnitude, the branch lengths are merely numbers adjusted by the underlying phylogenetic model, and therefore the correctness of these are of secondary importance. Lastly, the actual inferred ancestral sequences are determined by a combination of the underlying substitution model, the chosen tree and its branch lengths. While inferring correct tree topology is also important, the correctness of the inferred ancestral sequences are the foremost important objective of a sequence

reconstruction when these sequences are used for real applications involving DNA synthesis, protein expression and lab tests. For this reason the sole purpose of the COAR metric is to capture the correctness of the inferred ancestral sequences. In particular, we would like to have an inferred sequence loss function that makes sense even when the tree is incorrect.

The purpose of COAR is to compare two trees with the same leaves, lets call them the true and inferred tree. When performing ASR the desired result is often to reconstruct the internal nodes in the direct path going from a leaf to the root, as illustrated in figure 4-1. This is extracted by starting at a leaf node and traversing upwards in the tree, parent by parent, until the root is reached. In the following, this list of sequences will be referred to as the ancestral lineage. The ancestral lineage is the objective so lets use this to define COAR, and lets also set COAR to be the expected per-site error in such a reconstruction. Now following the example in figure 4-1. Often there will be small differences in tree topology between the true and inferred trees and these will likely make the number of internal states in the ancestral lineages differ. This makes comparison difficult because two lists of different length cannot be element wise compared. The lists could be made equal lengths by adding gaps but a systematic way of adding these would be needed.

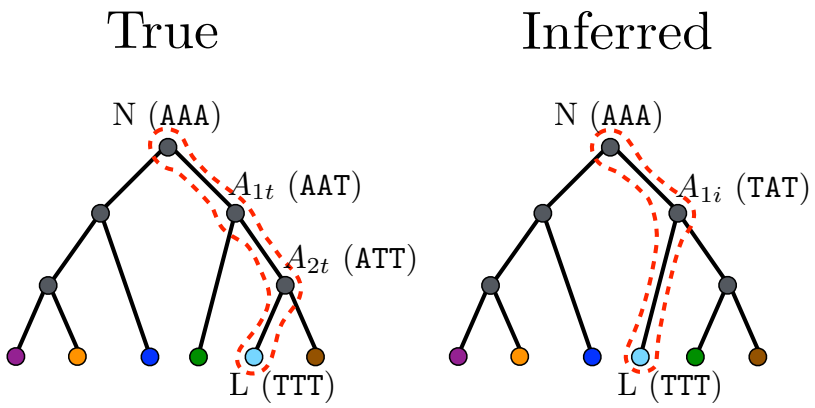


Figure 4-1: True vs. inferred tree with colored leaves and grey ancestral states. Reconstruction from the light blue leaf is marked by a dashed red line and annotated with genotypes in parenthesis. N is the naive sequence, L is the leaf sequence and the As are ancestors $1, 2, ..., n$ with either true or inferred marked by t or i, respectively, appended to the subscript. The inferred tree has misplaced the branch leading to the light blue node, resulting in a missing ancestor sequence. The missing ancestor is treated as a missing realization in the inferred mutation process.

The basis of COAR is a list comparison progressing element wise through the list i.e. element 1 in list 1 compared to element 1 in list 2, next element 2 in list 1 compared to element 2 in list 2 etc. For lists of similar length the list comparison

is easy, it will simply be the cumulated distance between the elements in the two list at the same position, and this corresponds to the sum of Hamming distances between inferred and true ancestors. When lists are not equally long one or more gaps must be introduced into one of the lists; we choose to do so in such a way that the list similarity is maximized. This is an alignment problem and the optimal solution has been described and solved by Needleman and Wunsch [65]. We restrict the global alignment result so that it has to start at the root of the tree and end at the leaf sequence because these two states are known for both the true and inferred phylogenies. We further restrict the Needleman-Wunsch algorithm so that gaps are only allowed to be introduced in the shortest of the two lists being aligned to force the maximum number of true vs. inferred node comparisons. With this restriction on gaps, the number of gaps is determined solely by the differences in list lengths and as long as the gap penalty is less than or equal to zero the gaps will always be at the same positions. Practically this means that the magnitude of the gap penalty only serves the purpose of introducing an extra penalty for inferring a wrong topology and can be adjusted to put more, or less, emphasis on tree topological correctness. A larger gap penalty puts more emphasis on correctness of tree topology.

One interpretation of the COAR value is that it is the distance between the true and inferred mutation histories as shown in figure 4-2. In this representation of an ancestral lineage the root and the leaf are two fixed states with a continuous mutation process running between them. The internal nodes in the ancestral lineage are discrete states in the continuous process, and while on the true tree this corresponds to a single cell, on the inferred tree they need not correspond to actual observed genotypes. Instead we can think about internal nodes on an inferred ancestral lineage as realizations along the continuous mutation process defined by the inferred tree. The COAR value is then a similarity measured between the true cell genotype and the inferred realizations, each sampled from the true and inferred mutation processes respectively, and now in the case of a mismatch between the number of realizations and cells, a gap will be introduced in the alignment to compensate. In this view more sequences in the inferred ancestral lineage means more realizations along the mutational path and vice versa. Those extra realizations could in fact be correctly inferred sequences with no cell observations on the true phylogeny. For this reason we set the gap penalty to be zero so that the COAR value will only reflect the difference in ancestral lineages and be indifferent to the possible differences in tree topology. The correctness of the tree topology can the be assessed separately by other metrics such as Robinson-Foulds distance.

Using the aligned ancestral lineages it is now possible to derive a score, simi-

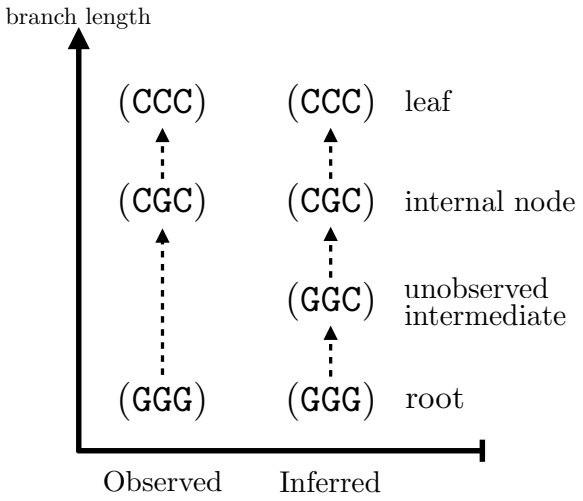


Figure 4-2: One interpretation of the COAR value is that it is the distance between the true and inferred mutation histories, here shown by the true and inferred ancestral lineage nodes of an example phylogeny. The true ancestral lineage (left side) represents actual observed cells where the genotype is a known constant. The inferred ancestral lineage (right side) represents the estimated genotypes at branching points along the inferred topology. In some cases there is a mis-correspondence between observed cells in the true phylogeny and the branching points in the inferred tree. These are treated as missing realizations and ignored in the alignment of the two mutation histories.

lar to a sequence alignment score, and then normalizing the alignment score to the smallest possible alignment score will give our measure for correctness of ancestral reconstruction.

$$\text{COAR}_i = \frac{\text{alignscore}(\text{leaf}_i)}{\text{alignscore}_{\min}(\text{leaf}_i)}$$

Where alignscore is the score of the alignment between the true and inferred ancestral lineages and in the denominator, alignscore_{min}, is the smallest possible score given the number of sequences in the ancestral lineages. The alignment score is defined in terms of penalties, so all values are negative, alignscore ≤ 0. Likewise the smallest possible alignment score is negative thereby canceling out the negatives to make a positive COAR.

COAR is defined in the range from 0 to 1, where 0 is a perfect ancestral sequence reconstruction and 1 is the worst possible. The COAR value is comparable across different trees, methods and datasets because of this normalization, and if the score is proportional to the sequence distance, COAR can be interpreted directly as the average per site error in the inferred ancestral lineage sequences. COAR for a single ancestral lineage can be expanded to the tree level by calculating the mean COAR

value for the whole tree:

$$\text{mean(COAR)} = \sum_{i=1}^{L_N} \frac{\text{alignscore}(\text{leaf}_i)}{\text{alignscore}_{\min}(\text{leaf}_i)} \Big/ L_N$$

Where L_N is the number of leaves on the tree.

Calculating COAR values - example with a known root

As an example of how the COAR metric works we will present a small example, summarized in figure 4-1 with the light blue leaf chosen for lineage reconstruction and the true and inferred ancestral lineages marked in each tree with red dashed lines. The example is on the phylogeny of a B cell receptor (BCR) clonal family in which case the root sequence is a known state called the naive sequence. Assume that the true phylogeny is known with corresponding ancestral sequences along the tree. Given the leaves of this tree and their sequences a phylogeny can be inferred using any method e.g. maximum parsimony, maximum likelihood, Bayesian methods etc. We make the restriction that only one inferred tree can be evaluated i.e. if multiple equally parsimonious trees exist one should be chosen at random, and if a Bayesian method is used the maximum posterior tree or a random tree weighted according to the posterior distribution should be chosen.

Now take a leaf sequence on the tree and reconstruct its ancestral lineage by extracting the parent, the parent's parent, etc. until the root is reached. Results for the trees in figure 4-1 are tabulated in table 4.1. This ordered list of sequences constitute the reconstructed ancestral lineage for the chosen leaf and it always start at the root and ends with the leaf sequence. Both the true and inferred tree may have any number of sequences in this list, however there must be as minimum 2, the root and the leaf. In this example the root state is known to be the naive BCR sequence, so we are imposing the restriction on the alignment that it must start with the root and end at the leaf, and that these are known states do not count towards the COAR value.

	True	Inferred
Naive (N)	AAA	AAA
A_1	AAT	TAT
A_2	ATT	-
Leaf (L)	TTT	TTT

Table 4.1: Reconstructed ancestral lineage for true and inferred trees as shown in figure 4-1.

In cases of a wrongly inferred topology the true and inferred lists of ancestral lineage sequences can have different lengths, so we need a way of finding the best possible alignment between these lists. We know the start and end of this alignment since that is defined as the shared root and leaf sequences, but the sequences between are free to be shifted up or down to maximize the alignment fit. As described above, we adapt the Needleman and Wunsch dynamic program [65] to solve the alignment problem in squared time complexity. A notable difference to the original Needleman-Wunsch algorithm is that it was intended to align two sequences of characters, like DNA or amino acids, while in this application instead of aligning a list of sequence characters a list of whole sequences are aligned. Here we work through an example in detail.

The first step in the alignment algorithm is to calculate a score matrix of all pairwise sequence comparisons. For this example we use the negative of the Hamming distance between sequences as a score and therefore populating the score matrix is a simple procedure. However the score function can be extended to reflect different situations, like larger penalty for non-synonymous rather than synonymous mutations. By using simple Hamming distances, we will in this example be weighting all mismatches equally. The score matrix is tabulated in table 4.2. We use negative scores to reflect that mismatches represents a loss.

	N	A_{1t}	A_{2t}	L
N	0	-1	-2	-3
A_{1i}	-2	-1	-2	-1
L	-3	-2	-1	0

Table 4.2: Score matrix based on all pairwise distances between the sequence in figure 4-1.

Now we are ready to initializing the alignment grid central to the Needleman-Wunsch algorithm. Initialization is started by inserting the scores of pure gap characters i.e. first row and first column, see table 4.3. Since the naive sequence is known we require the two root sequences to align by setting this gap penalty to negative infinity. Similarly we disallow introduction of gaps in the longest of the sequence lists by penalizing these as negative infinity as seen in table 4.4. Setting certain gap penalties to negative infinity is a simple way of dealing with disallowed gaps and it works well in an implementation.

Then the alignment grid is filled up, starting with the cell that has left, top and diagonal cells filled (marked by $\twoheadrightarrow$ in table 4.3) and continuing to the rightmost cell.

	-	N	A_{1t}	A_{2t}	L
-	0	-Inf	-Inf	-Inf	-Inf
N	-Inf	⇾			
A_{1i}	-Inf				
L	-Inf				

Table 4.3: The starting alignment grid, initialized with negative infinite gap penalties to disallow gap opening in the beginning of the alignment. The grid is filled up from left to right row by row, starting in the cells with left, top and diagonal cells filled (marked by ⇾).

Cells are filled up by the following equation:

$$C_{i,j} = max\left\{(C_{i-1,j} + gp_{\text{down}}); (C_{i,j-1} + gp_{\text{right}}); (C_{i-1,j-1} + S_{i-1,j-1})\right\}$$

Cells are iterated through in the order $i = 1, 2, 3, 4$ and $j = 1, 2, 3, 4, 5$, where $C_{i,j}$ is the ith row and jth column cell in the grid, $S_{i-1,j-1}$ is the score of aligning the ith, jth elements found by look-up in the score matrix (table 4.2) and gp_{down} and gp_{right} is the gap penalty of the moving down and right respectively. In this example the longest sequence list is the true ancestral lineage and therefore gaps are only allowed to be introduced into the inferred list of sequences. The penalty for introducing gaps in the true list of sequences is set to negative infinity to reflect this and therefore $gp_{\text{down}} = -\text{Inf}$ and $gp_{\text{right}} = 0$.

The grid is filled and results store in table 4.4. The final alignment score is the number in the rightmost bottom cell.

	-	N	A_{1t}	A_{2t}	L
-	0	-Inf	-Inf	-Inf	-Inf
N	-Inf	0	0	0	0
A_{1i}	-Inf	-Inf	-1	-1	-1
L	-Inf	-Inf	-Inf	-2	-1

Table 4.4: The filled alignment grid, ready for tracing back the best alignment. The rightmost bottom cell has the score for the best alignment.

The last step is to traceback the best path through the alignment grid and store this alignment. The traceback starts from the leaf sequence, in the right bottom corner, and ends with the naive sequence in the left top corner. A diagonal step is equivalent to a sequence match, a left move is introducing a gap character in the inferred list and a move up is introducing a gap in the true list. The path is found by moving backwards following the same path that was used to fill up the grid, and therefore using the same equation as to fill it, just reversed with moving from bottom

right cell to top left, $i = 4, 3, 2, 1$ and $j = 5, 4, 3, 2, 1$:

$$\text{move}_{i,j} = \text{which}\,\{C_{i,j} = [(C_{i-1,j} + gp_{\text{down}}), (C_{i,j-1} + gp_{\text{right}}), (C_{i-1,j-1} + S_{i-1,j-1})]\}$$

Notice that this path has already been generated when the alignment grid was filled up and could be cached. The resulting alignment and the penalty for each position is tabulated in table 4.5.

Lastly this value is normalized by the smallest legal alignment score assuming that all matched positions contained sequences with maximal distances i.e. no similarity. This normalized number is the COAR value and it runs between 0 to 1. In the presented example we only calculated the COAR value for the reconstructed ancestral lineage from one leaf node but doing the calculations on all leaves on the tree and taking the average would give the mean COAR value for the whole tree.

True Inferred	N N	A_{1t} A_{1i}	A_{2t} -	T T
Penalty	0	-1	0	0
Max penalty	0	-3	0	0
COAR	-1/-3=0.333			

Table 4.5: The resulting alignment and the penalty for each positions.

4.2.2 Other validation metrics

Most recent common ancestor distance

As an alternative to COAR we are also using a measure named the "most recent common ancestor" (MRCA) metric. In this more simple measure of correctness of ASR the MRCA ancestral sequence between two leaves are found and compared in the true vs. inferred phylogeny. By iterating through all pairwise combinations of leaves the MRCA distance can be found as the sum:

$$\text{MRCA}_{\text{dist}} = \sum_{i=1}^{N} \sum_{j=i+1}^{N} \text{dist}(T_{\text{MRCA}}(l_i, l_j), I_{\text{MRCA}}(l_i, l_j))$$

All the pairwise combinations of leaves l_i and l_j $(i \neq j)$ to find their true (T_{MRCA}) and inferred (I_{MRCA}) MRCA. The distance between the two ancestral sequences are found with the function $\text{dist}(\cdot, \cdot)$ and summed over all pairs. In this work we use the Hamming distance as distance function.

Similar to COAR, ancestral sequences close to the root will have more influence on

the result than sequences close to the tips. The major difference is that MRCA is not scaled and not representing the result of a lineage reconstruction and can therefore the value has no meaningful interpretation aside from measuring performance of different method on identical data.

Robinson–Foulds distance

The tree topology is the model that explains the order of branching events and relatedness of leaves, and therefore inferred ancestral sequences will always be tied to the topology. Hence validating the ASR should also express some degree of correctness of the tree. Regardless it is interesting to have a method solely expressing the correctness of inferred tree topology so for this purpose we use the Robinson–Foulds (RF) distance [79]. Briefly the RF distance is defined as the number of different tree partitions between two trees. Tree partitions are made by cutting a branch connecting two nodes, all possible cuts are performed and the resulting split of taxa are sorted and recorded into two columns. The after sorting the list of partitions they are compared true vs. inferred tree and for each partition mismatch the RF distance is increased by 1.

4.2.3 Algorithms tested

With the exception of a few methods there has been little published about BCR phylogeny specific software. Barak et al. made some vaguely defined rules written into an algorithm they named IgTree [4]. The description of IgTree is similar to that of maximum parsimony (MP) and it is unclear whether there is any difference at all. For that reason we use the MP algorithm `dnapars` from PHYLIP [70] as a substitute for IgTree and other methods e.g. Tas et al. [96].

In the paper from Doria-Rose et al. [17] they use the general time reversible (GTR) model implemented in MEGA5 [95]. Similarly Kepler [46] was using `dnaml` from the PHYLIP package to infer BCR phylogenetic trees. To test these commonly used maximum likelihood trees we use `dnaml`.

Two different models have been defined specifically for BCR evolution on protein level, the AB amino acid substitution matrix from Mirsky et al. [61] and HLP17 codon model from Hoehn et al. [40]. Both are implemented with codonPhyML [30]. The AB substitution matrix is just a standard amino acids substitution rate matrix but fitted to substitutions observed in BCR sequences, yielding a symmetric rate matrix that assumes equilibrium. But recently Sheng et al. found large discrepancies between their findings and the AB substitution matrix [89], promoting us to exclude

this from our evaluation. Instead we test the HLP17 model as it is implemented in the software IgPhyML (github.com/kbhoehn/IgPhyML). The HLP17 model is based on a GY94 model [34] but adding parameters to account for AID hot/cold spots and thereby attempting to address the context sensitivity of SHM.

Lastly we also test the performance of an unpublished method that uses a likelihood based ranking of equally parsimonious trees, we call this method GCtree, for genotype collapsed tree. Briefly, the GCtree likelihood function is based on the assumption that the germinal center reaction can be modelled as a binary Galton-Watson process [37]. Some genotypes have many observations while others have fewer, the Galton-Watson process will favour a high abundance genotype to be the parent for a low abundance genotype. Each MP tree is evaluated and ranked according to its fit and then the highest likelihood tree is picked out as the GCtree inferred tree. The GCtree method assumes getting the correct genotype abundances and will be sensitive to experimental errors in these.

4.2.4 Simulated data

Simulations were performed using both the neutral branching process and the affinity simulation previously described in chapter 3. Both simulation, inference and evaluation was wrapped into an SCons script [27] using nestly [59] to loop through all combinations of requested parameters. This script is available in the codebase of `GCtree` (github.com/matsengrp/gctree). SCons commands used can be found in appendix C.

4.3 Results - comparing algorithms for B cell phylogenetic reconstruction

The most suitable dataset for clonal family phylogeny is data collected from a single GC like one in Tas et al. [96]. But unlike single cell sequences most data is generated from bulk RNA extract from peripheral blood mononuclear cell (PBMC). To adjust the simulation parameters to recapitulate a given dataset we both used the single cell Tas dataset (see appendix B), and an HTS dataset generated from PBMCs from HIV infected individuals. In these HTS datasets cluster were done using partis resulting in hundreds of clonal families. To only isolate HIV relevant clonal families a seed sequence was used that was known from lab work to be HIV-responsive and only sequences in the same clonal family were extracted. These seed sequences were generated by single cell screening of the memory fraction of the PBMCs, followed by

Sanger sequencing of the cells expressing HIV neutralizing antibodies. The size of the clonal families varied substantially as shown in figure 4-3, but they serve as a more realistic dataset case than the single cell data from Tas et al. for many cases.

Simulation parameters were adjusted in order to recapitulate two measures: a) distribution of sequence across the length of the tree and b) how "spread out" the tree leaves were. Measure a) is captured by plotting the CDF of the distance to the naive/root sequence ((a) in figure 4-3 and 4-4). Measure b) is captured by plotting the CDF of the nearest neighbor distance ((b) in figure 4-3 and 4-4). The neutral simulation with a high λ will naturally vary a lot in early population size an therefore had no problem fitting into the large span of possible summary statistics defined by two chosen clonal family representatives from the HTS data, see figure 4-3. The nature of the affinity simulation forces it to have a rather constant population size defined by its carrying capacity. Instead the GC was down-sampled to 150 sequence and the variance in the number of observed sequences comes from the fact that some sequences are identical at DNA level and deduplicated, see figure 4-4.

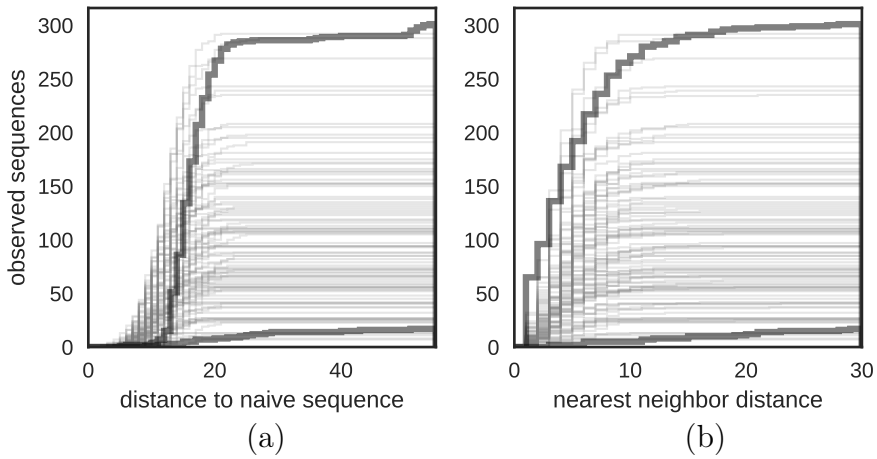


Figure 4-3: Summary statistics for the unique sequences simulated under a neutral model fitted to HTS data. The two thick dark grey lines represents the characteristics of two clonal families extracted by partis seed clustering on HTS data. Smaller light grey lines are showing 1 of the 100 simulated datasets. Non default parameters used: $T = 5$, $\lambda = 2.5$, $\lambda_{\text{mut}} = 3$

Looking a RF distance for the neutral simulation there are only small differences and only the distribution of RF distances for MP is shift up compared to the others (significant $p < 0.05$ in Mann–Whitney U (MWU) test), see figure 4-5. Likewise the quartiles are slightly shifted upward for MP in the comparison of MRCA distances. We consider COAR the most important metric because this has a direct interpretation inspired by the purpose of the tree, and for COAR there seems to be no significant

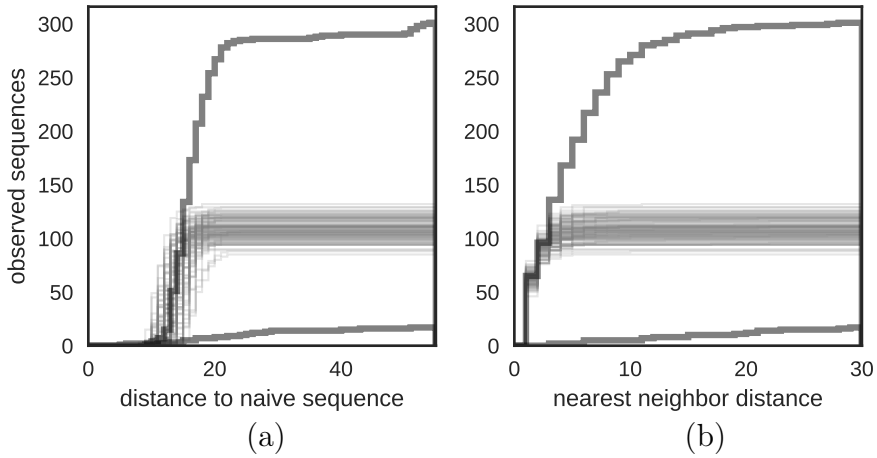


Figure 4-4: Summary statistics for the unique sequences simulated under the affinity model fitted to HTS data. The two thick dark grey lines represents the characteristics of two clonal families extracted by partis seed clustering on HTS data. Smaller light grey lines are showing 1 of the 100 simulated datasets. Non default parameters used: $T = 90$, $n = 150$, $\lambda_{\text{mut}} = 0.25$

difference between the methods when simulating under neutral conditions. The average COAR values are ranging from 1.76E-04 to 2.37E-04 (appendix A table A.2) corresponding to approx. 10% chance of one nucleotide error per reconstructed sequence in a lineage from leaf to root. Similar but somewhat large differences are observed for neutral simulation using parameters fitted to the Tas dataset, see figure B-4 in appendix B. Should we rank the methods based on all metrics from best to worst the rank would be: GCtree, dnaml, IgPhyML, Parsimony. But the first three are virtually the same. The Tas dataset fitted simulations also experience a similar range of expected errors in ASR, though slightly lower because of the fewer sequences simulated, see table A.1 in appendix A.

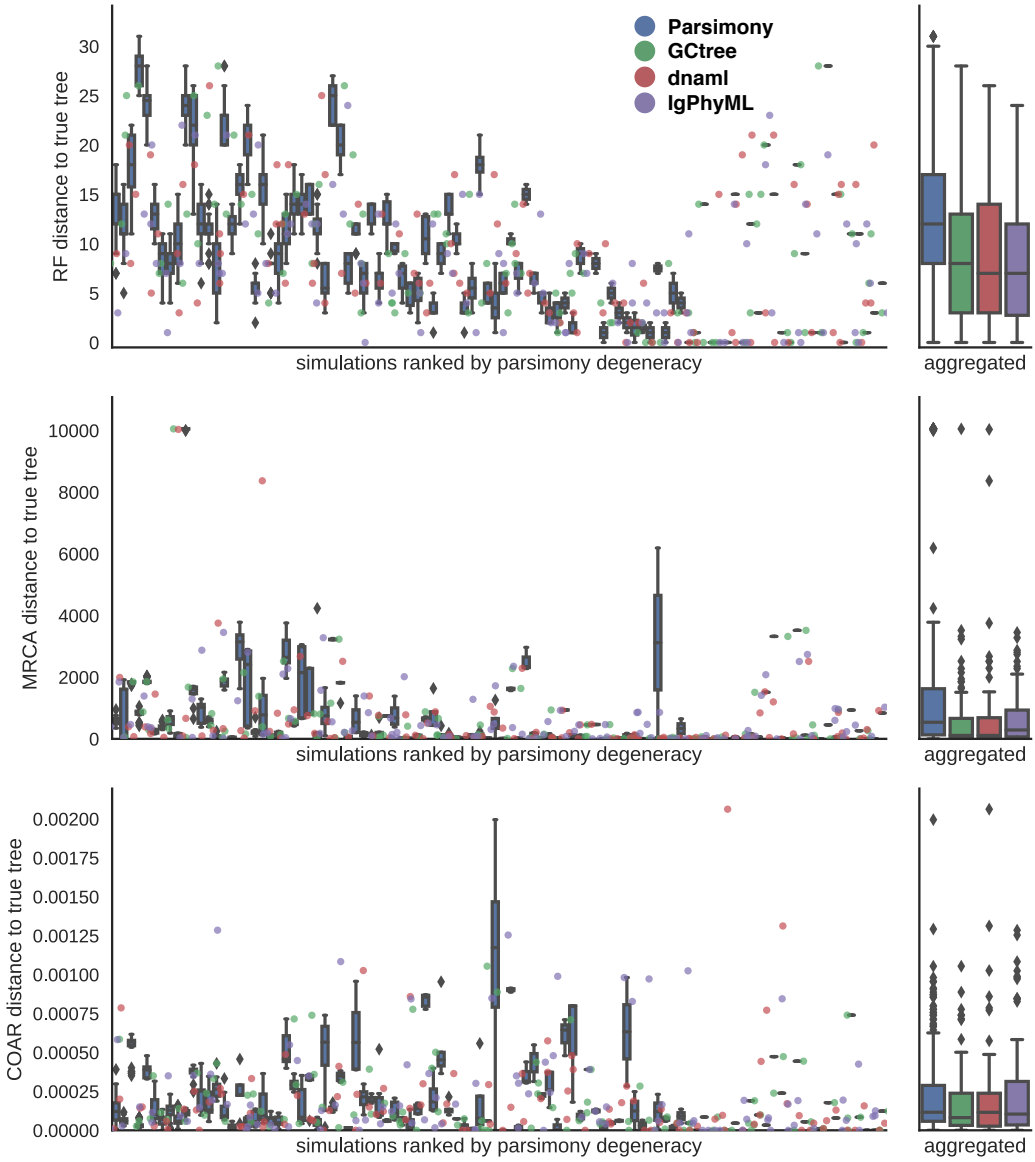


Figure 4-5: Simulation with 100 repeats of a neutral branching process. Each simulation is plotted in a single column ranked according to the number of equally parsimonious trees for the simulation. The ensemble of equally parsimonious trees are shown as a box plot while the other methods are plotted as jittered dots. On the right the aggregated result is shown.

For the affinity simulations we allow genotypes to reappear in another clade during simulations i.e. homoplasy is allowed. This prevents us from calculating the RF distance but not from the other metrics. However for both MRCA and COAR there is no significant differences resulting in a tie between all the inference methods, see figure 4-6. Similar characteristics are observed for affinity simulations fitted to the Tas dataset, although with slightly worse performance of MP and IgPhyML, see figure B-8 in appendix B. Again the best to worse rank would be: GCtree, dnaml, IgPhyML, Parsimony, but with a small effect size separating them.

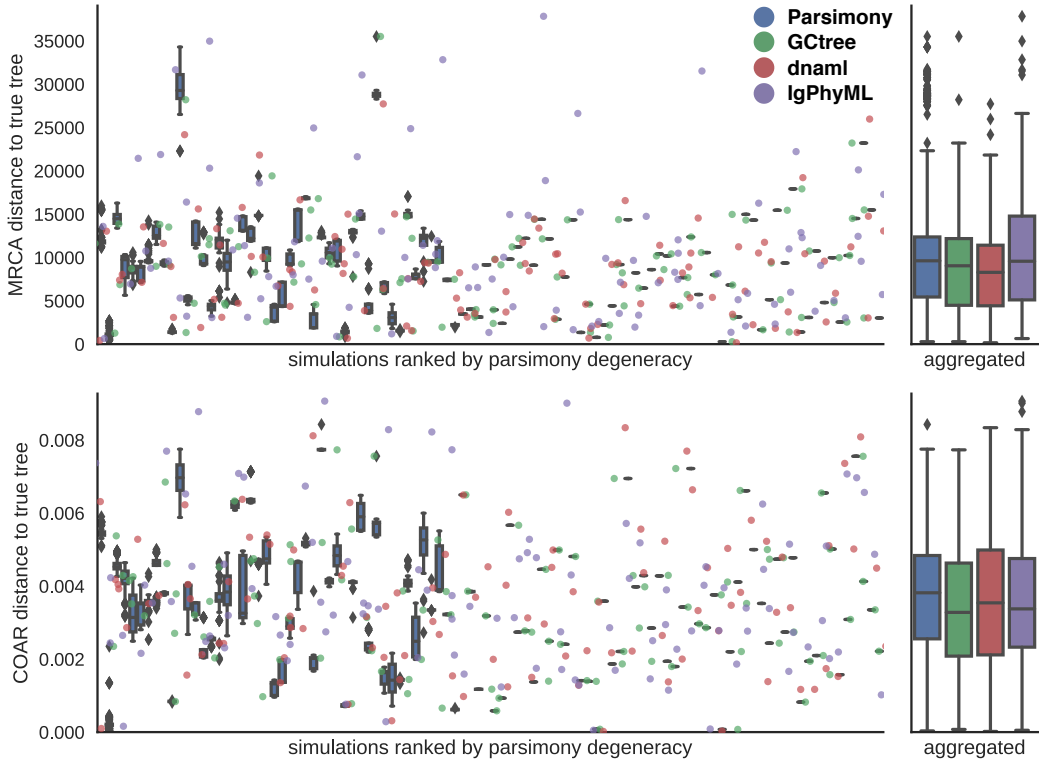


Figure 4-6: Simulation with 100 repeats of the affinity simulation. Each simulation is plotted in a single column ranked according to the number of equally parsimonious trees for the simulation. The ensemble of equally parsimonious trees are shown as a box plot while the other methods are plotted as jittered dots. On the right the aggregated result is shown.

4.4 Discussion and conclusion

We have presented the COAR metric as an objective function tailored for evaluating ASR on a non-fixed inferred phylogeny. COAR is different from other tree metrics by being robust towards tree topology errors irrelevant for ASR and by being directly

interpretable as the expected per site error of the sequences in a reconstructed lineage. Using COAR and other tree metrics we went on to show that reconstruction of sequences along an ancestral lineage is robust and only rarely accumulate errors, regardless of the inference method tested. In a worst-case scenario we found an expected one nucleotide error per reconstructed lineage sequence, based on affinity simulation with 4-8 sequences (including the a priori know root and leaf) in a lineage, see b) in figure B-9 appendix B.

We use two kinds of simulation, with and without selection. A neutral branching process serve as a good baseline but is not realistic compared to the fast and stringent selection process which is undertaken in the GC reaction. The simulation with added affinity selection attempts to address this issue by including a semi-mechanistic model for BCR-antigen affinity, challenging the inference methods with both epistasis and homoplasy. It is therefore not surprising to see that inference made on affinity simulations obtain higher COAR values than neutral simulation under comparable conditions. What we do find surprising is that all inference methods seems to be equally effected by the imposed affinity selection. We would have expected a codon based model like IgPhyML to have an advantage compared to the nucleotide based models because affinity selection is strictly acting on protein level, and hence there is a preference for synonymous over non-synonymous mutations that is an integrated part of a codon model but not accounted for in a nucleotide model. IgPhyML should also have an advantage over the other methods due to its model incorporating AID hot/cold spot motifs. Although IgPhyML is marginalizing over motifs that span multiple codons to achieve site independence and this might water out some signal, it should still capture some of the signal from mutational context bias. However, again this does not seem to improve the inference by IgPhyML over the other methods.

Looking at typical trees simulated from both the neutral and affinity selected process (see trees in appendix B) it looks like trees are relatively shallow i.e. the number of internal nodes from the leaf to the root is small. This of course makes it easier to get a low COAR score because there are only few states to reconstruct. In the affinity model trees are simulated over many generations but still appear shallow because cells with low affinity are lost during the competitive selection of maturation. On a tree this "memory loss" of lower affinity variants will result in a tree with a long trunk extending from the root to a most recent common ancestor of a bushy canopy of leaves, similar to observations made by Yaari et al. [109] (visualized in b) in figure B-9 in appendix B). This trait could explain why is there is no apparent benefit of integrating AID motifs into an inference algorithm like IgPhyML, the effect could in fact be positive but too small to be detected in our ASR validation using shallow trees.

Hoehn et al. describing IgPhyML provides results of their own simulation study based on a fixed tree topology inferred from clonal BCR data ranging from approx. 300 to 1000 taxa. With these trees containing more taxa than our simulations, they show that integration of AID hotspots does increase ASR correctness slightly compared to the GY94 codon model. Still, they only get an average of 3% improvement (at h=2) over the regular GY94 model confirming our findings of high similarity between methods.

The overall performance of all the evaluated inference methods is very similar. There is a consistent ranking that puts GCtree first, dnaml and IgPhyML second and finds Parsimony last, but it must be stressed that the difference is very small and that the performance can be regarded as practically equivalent. GCtree does stand out as the most consistently better performing algorithm but it is also dependent on genotype abundance data, which is not readily available for most BCR data sets. Abundance data could potentially be extracted from standard HTS data, but because of sequencing errors and primer biases these abundances are not expected to be reliable. But with the introduction of UMIs it is expected most future datasets will have much more reliable abundance information, giving the opportunity to integrate abundance data in methods such as GCtree to direct phylogenetic inference.

4.5 Conclusion

The problem of inferring BCR phylogenies appears to be insensitive to the method used on the simulation regimes tested. With the GCtree inference being significantly better than the other tested method, we suggest that this is the first choice if reliable abundance information exists. Otherwise, if GCtree cannot be used because of missing abundance information, the methods tested are practically equivalent suggesting that users should care mostly about which tool provide the most convenient solution to inferring their BCR phylogeny. It remains to be tested whether different sampling conditions, such as time series sampling, will alter the results. Likewise it is undetermined if joint reconstruction will do any improvement over marginal reconstruction.

Chapter 5

Perspectives

Pretty much no difference. Maybe IgPhyML is over-parameterized. - Or selection is strong enough to make the tree look shallow in mutational landscape (long trunk and bushy canopy) - This could be assessed by taking multiple time points on the tree

Why don't we see any difference? For affinity selection this could be because that information is lost during the selection steps i.e. many mutations are getting selected against.

It would be interesting is investigate the effect of using joint reconstruction or just assess the likelihood difference between maximum likelihood and the second highest likelihood. If these are close joint reconstruction might change the results substantially but if they are always far away then jointly reconstructing all ancestral sequences are very likely to be the same, in which case joint reconstruction would be a waste of computations.

We could also let us inspire by the Hoehn paper and infer a distribution of tree topologies for real data and then simulate the sequences along these trees.

Future studies should include more methods e.g. PRANK.

Use more complicated simulation methods like hyphasma with the same idea of making a target sequence and then start maturing towards this. This would be even more realistic simulation of sequences and might even make the mechanistic model in hyphasma better.

Bayesian phylogenetic pairing of heavy and light chains.

HIV vaccine design and also vaccine design in general.

Appendix A

Tables

	COAR		MRCA		RF	
	Mean	STDV	Mean	STDV	Mean	STDV
Parsimony	1.67E-04	2.32E-04	38.6	65.2	3.59	2.40
GCtree	7.57E-05	2.01E-04	26.1	84.0	1.46	1.87
dnaml	2.01E-05	9.08E-05	30.9	103	0.49	0.98
IgPhyML	9.95E-05	1.74E-04	35.0	101	2.39	2.02

Table A.1: Mean and standard deviation (STDV) for 100 simulations under the neutral model fitted to the Tas. dataset. Plotted in B-4.

	COAR		MRCA		RF	
	Mean	STDV	Mean	STDV	Mean	STDV
Parsimony	1.93E-04	2.03E-04	1116	1804	13.0	7.04
GCtree	1.76E-04	2.23E-04	577	1229	8.96	7.22
dnaml	2.37E-04	4.34E-04	602	1415	8.6	6.6
IgPhyML	2.29E-04	2.31E-04	873	1832	7.69	6.14

Table A.2: Mean and standard deviation (STDV) for 100 simulations under the neutral model fitted to HTS data. Plotted in B-6.

	COAR		MRCA	
	Mean	STDV	Mean	STDV
Parsimony	8.62E-04	1.10E-03	310	189
GCtree	7.34E-04	9.06E-04	274	277
dnaml	6.21E-04	8.89E-04	185	193
IgPhyML	7.12E-04	8.93E-04	280	228

Table A.3: Mean and standard deviation (STDV) for 100 simulations under the affinity model fitted to the Tas. dataset. Plotted in B-8.

	COAR		MRCA	
	Mean	STDV	Mean	STDV
Parsimony	3.66E-03	1.70E-03	9770	5838
GCtree	3.44E-03	1.79E-03	8899	5848
dnaml	3.64E-03	1.99E-03	8648	5497
IgPhyML	3.74E-03	2.09E-03	11055	8271

Table A.4: Mean and standard deviation (STDV) for 100 simulations under the affinity model fitted to HTS data. Plotted in B-10.

Appendix B

Figures

B.1 Affinity simulation trees with stats

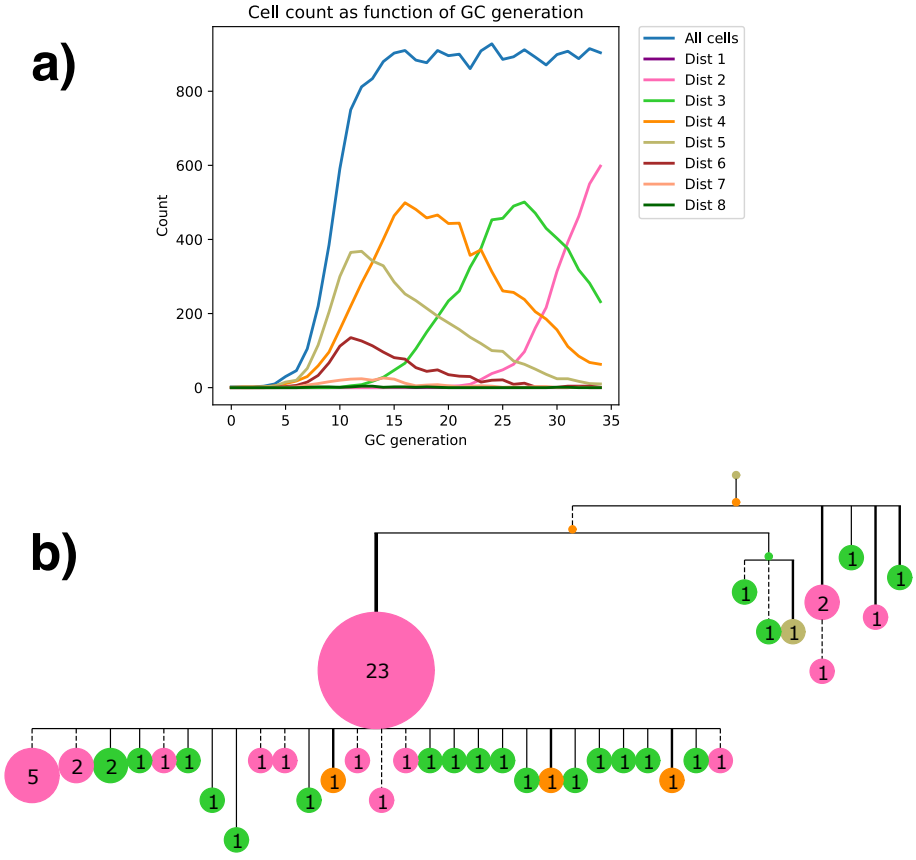
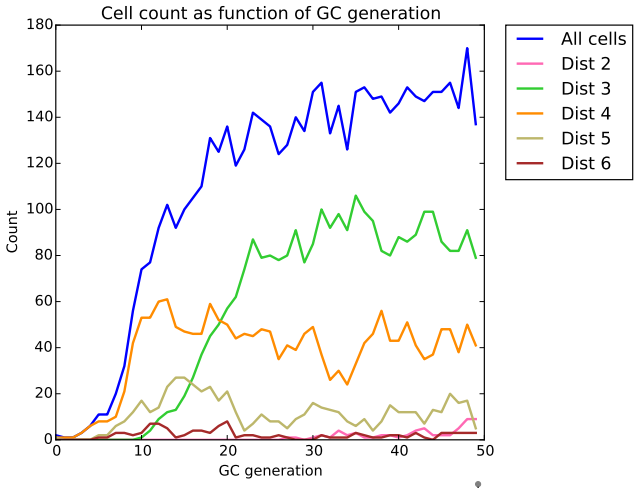


Figure B-1: Summary statistics for the simulation similar to a single cell GC in figure 3-12. a) run stats with color codes corresponding to affinity (through smallest distance to a target), b) resulting tree with colors matching those in a).

B.2 Affinity simulation with visual epistasis

(a)



(b)

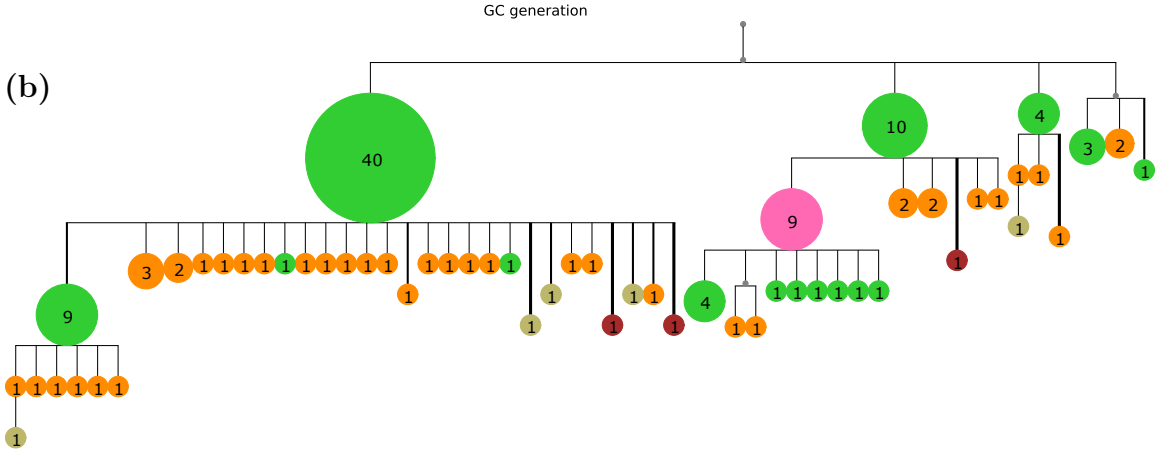


Figure B-2: Summary statistics for the simulation showing switch in target sequences trajectories described in figure 3-9. a) run stats, b) resulting tree.

B.3 Simulation comparison to data

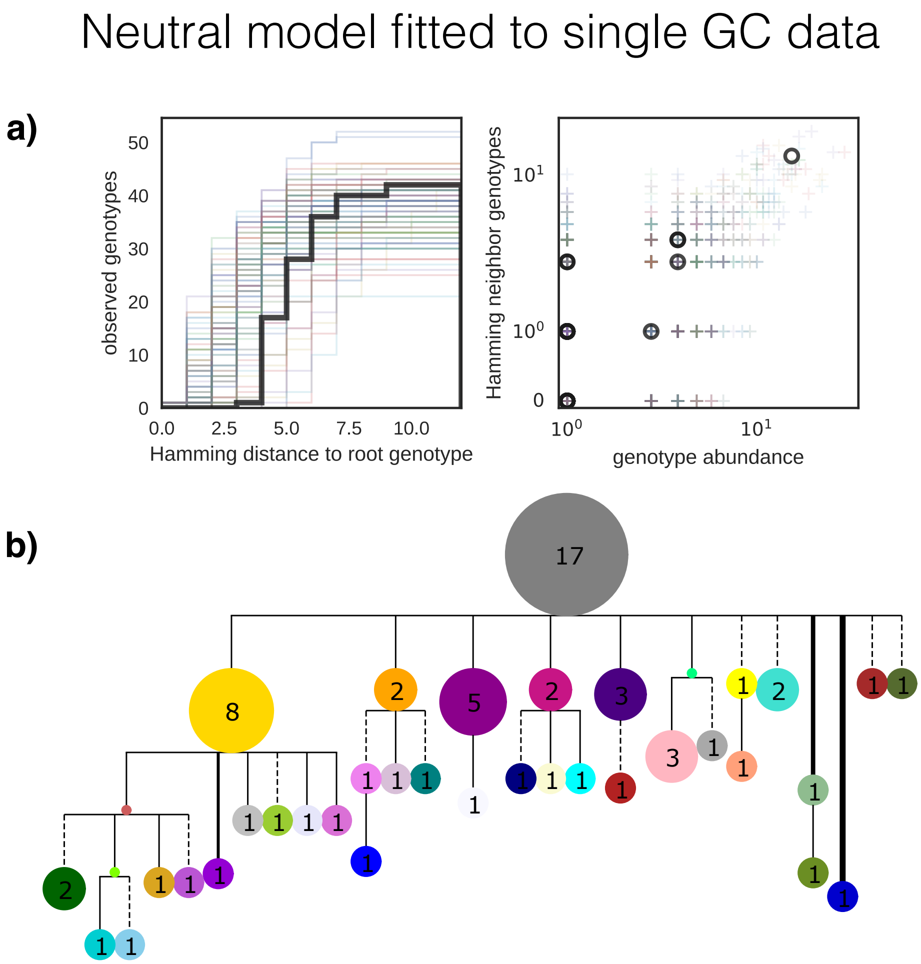


Figure B-3: Neutral branching process with parameters fit to single cell data. In a) summary statistics of how well the simulations fit data (simulation in colors, data in black). In b) a typical tree topology from the simulation run.

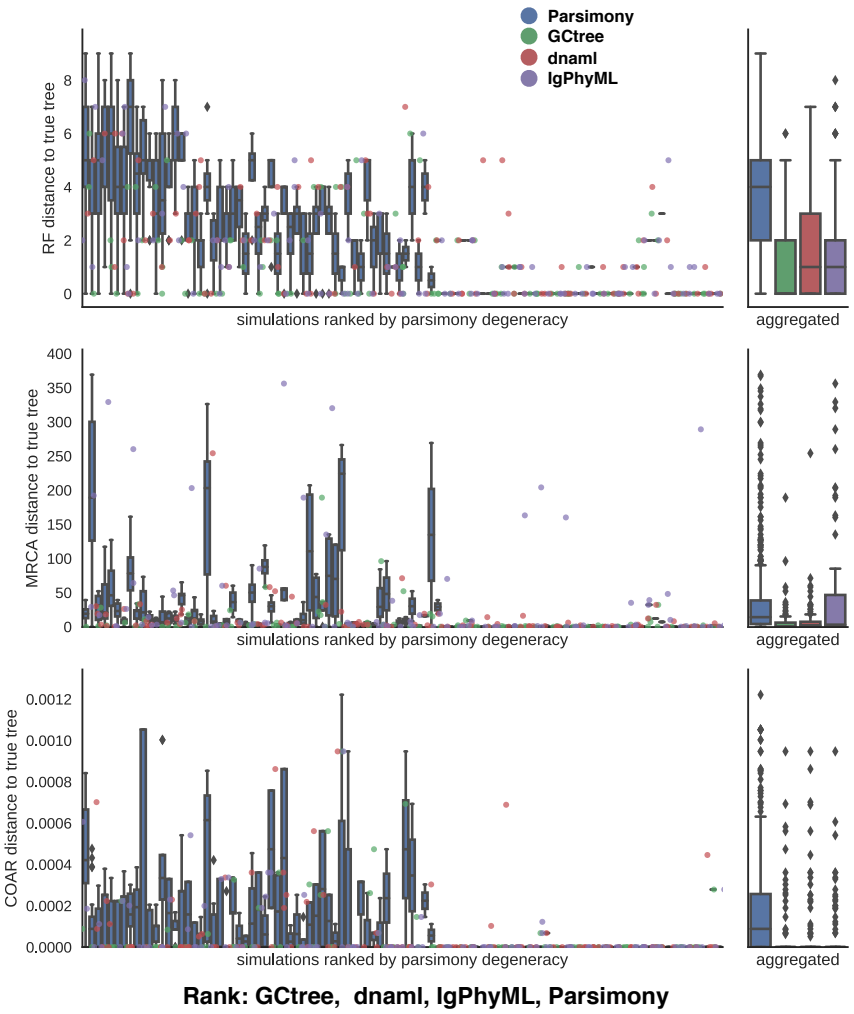


Figure B-4: Performance of different inference method over the 100 simulations shown in B-3. Standard box plot format with the box covering the two middle quartiles (Q2=25% to Q3=75% percentile), whiskers extends these and extra 1.5 times the interquartile range and points outside this are plotted individually. The median is indicated by a black line. A rank of best to worst, is subjectively decided based on the metrics plotted and with importance of the metrics determined by the rank; COAR, MRCA, RF.

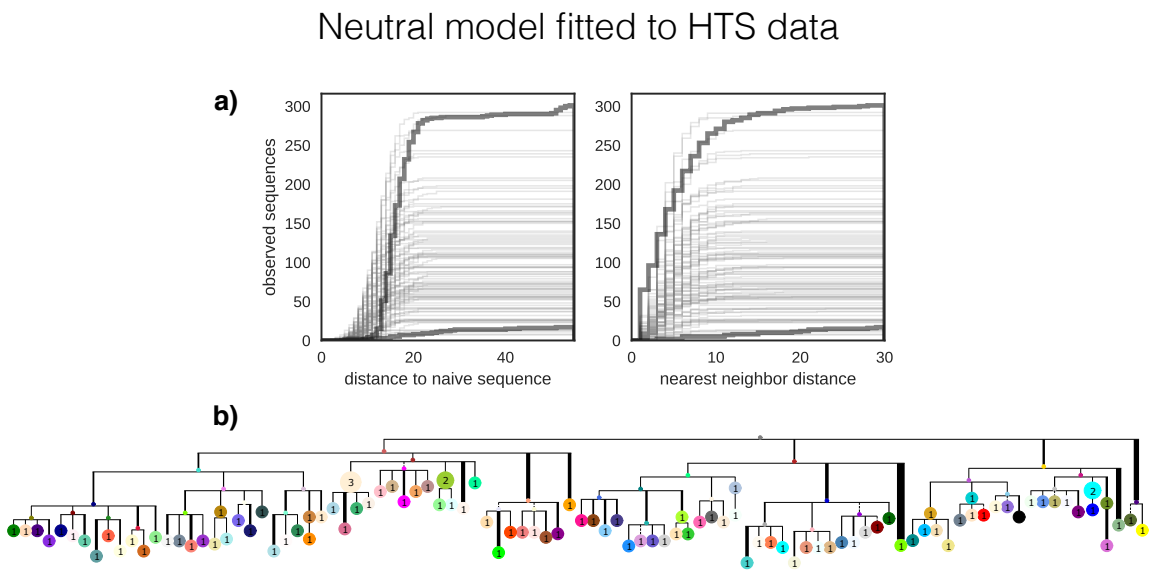


Figure B-5: Neutral branching process with parameters fit to HTS data. In a) summary statistics of how well the simulations fit data (simulations in grey shade, data in dark grey). In b) a typical tree topology from the simulation run.

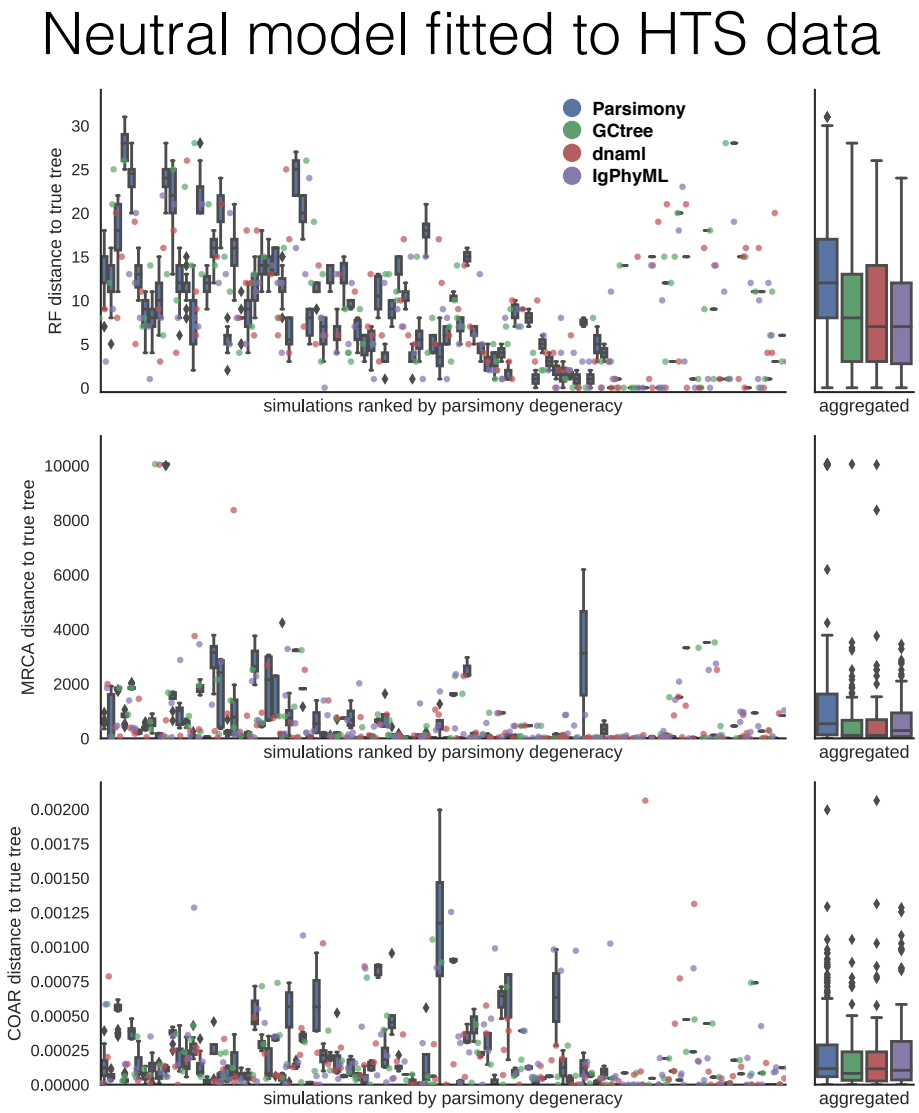


Figure B-6: Performance of different inference method over the 100 simulations shown in B-5. Standard box plot format with the box covering the two middle quartiles (Q2=25% to Q3=75% percentile), whiskers extends these and extra 1.5 times the interquartile range and points outside this are plotted individually. The median is indicated by a black line. A rank of best to worst, is subjectively decided based on the metrics plotted and with importance of the metrics determined by the rank; COAR, MRCA.

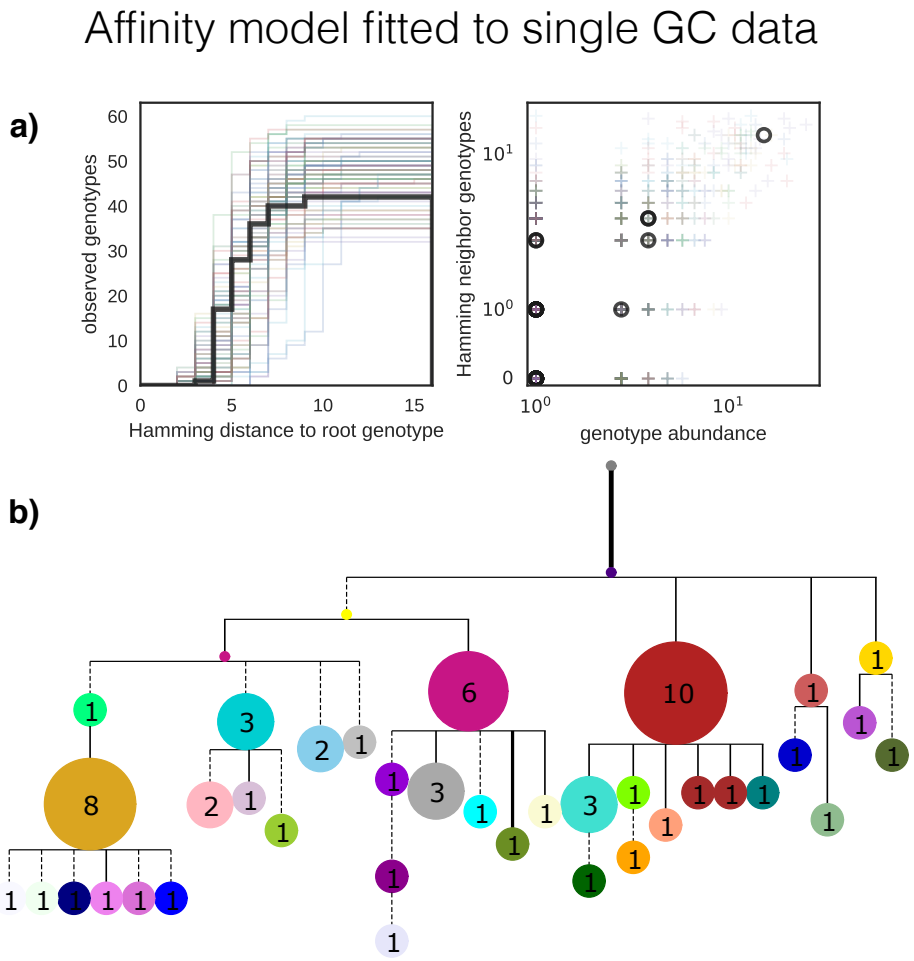


Figure B-7: Affinity simulation with parameters fit to single cell data. In a) summary statistics of how well the simulations fit data (simulation in colors, data in black). In b) a typical tree topology from the simulation run.

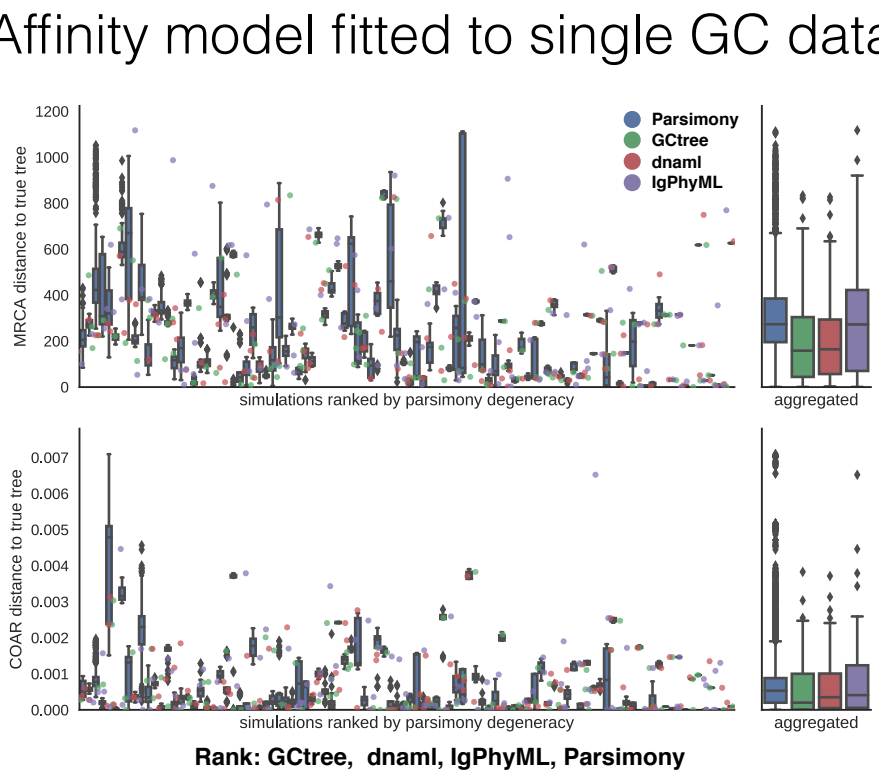


Figure B-8: Performance of different inference method over the 100 simulations shown in B-7. Standard box plot format with the box covering the two middle quartiles (Q2=25% to Q3=75% percentile), whiskers extends these and extra 1.5 times the interquartile range and points outside this are plotted individually. The median is indicated by a black line. A rank of best to worst, is subjectively decided based on the metrics plotted and with importance of the metrics determined by the rank; COAR, MRCA, RF.

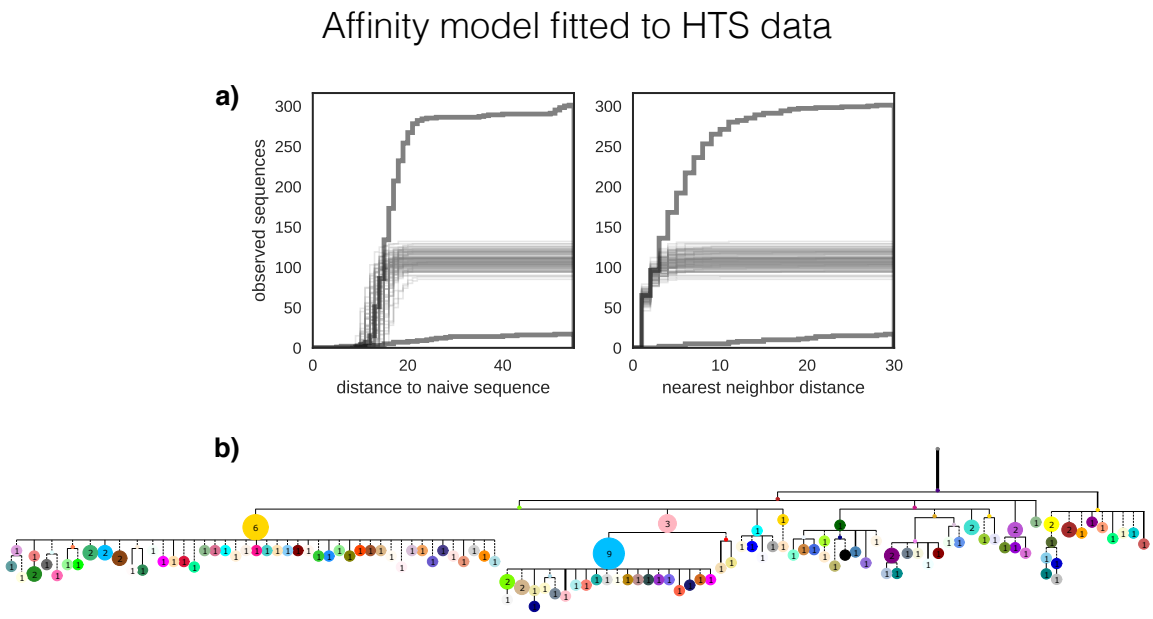


Figure B-9: Affinity simulation with parameters fit to HTS data. In a) summary statistics of how well the simulations fit data (simulations in grey shade, data in dark grey). In b) a typical tree topology from the simulation run.

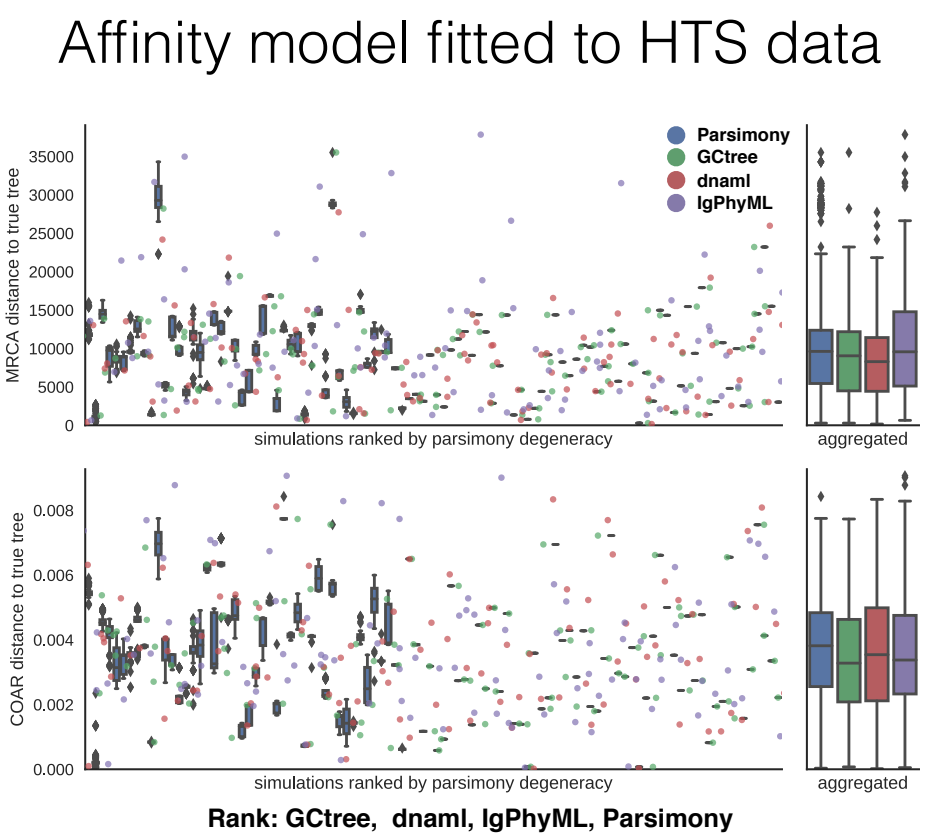


Figure B-10: Performance of different inference method over the 100 simulations shown in B-9. Standard box plot format with the box covering the two middle quartiles (Q2=25% to Q3=75% percentile), whiskers extends these and extra 1.5 times the interquartile range and points outside this are plotted individually. The median is indicated by a black line. A rank of best to worst, is subjectively decided based on the metrics plotted and with importance of the metrics determined by the rank; COAR, MRCA.

Appendix C

Source code

We find it unfortunate that the majority of previously published simulation methods does not offer publicly available software. Therefore, to facilitate as much transparency and usability of the method presented in this work, we have chosen to share all our code as open source through a Git repository. Hopefully this will encourage non developers to use and test our methods and potentially inspire to applications in fields that we have not yet investigated.

Methods for neutral, as well as affinity, simulations (used and described in chapter 3) are all collected in the same code repository along with SCons wrappers to run comparisons between inference methods (used in validation study in chapter 4). This is all available in the `GCtree` code base: github.com/matsengrp/gctree

In addition, the SCons commands used in the validation study, used to create the statistics in Appendix A and the figures in Appendix B, are listed below:

```
scons --simulate --igphyml --gctree --dnaml --outdir=HTS_aff_sim --frame=1
    --naive=CAGGTGCAGCTGGTGCAGTCTGGGGCTGAGGTGAAGAAGCCTGGGGCCTCAGTGAA
    GGTCTCCTGCAAGGCTTCTGGATACACCTTCACCGGCTACTATATGCACTGGGTGCGACAGGCCCCTGGACAA
    GGGCTTGAGTGGATGGGATGGATCAACCCTAACAGTGGTGGCACAAACTATGCACAGAAGTTTCAGGGCA
    GGGTCACCATGACCAGGGACACGTCCATCAGCACAGCCTACATGGAGCTGAGCAGGCTGAGATCTGACGACA
    CGGCCGTGTATTACTGTGCGAGAGGGCCATTCCCGAATTACTATGGTACGGGGAGTTATTGGGGGGGTTTTGA
    CTACTGGGGCCAGGGAACCCTGGTCACCGTCTCCTCA --experimental=<PATH TO FASTA FILE
    WITH EXPERIMENTAL GC SEQUENCES> --naiveIDexp=naive0 --lambda0=0.25 --
    selection --target_dist=10 --target_count=1000 --verbose --T=90 --carry_c
    ap=10000 --skip_update=1000 --nsim=100 --n=150 --quick &> HTS_aff_sim.log

scons --simulate --igphyml --gctree --dnaml --outdir=HTS_neut_sim --frame=1
    --nsim=100 --T=5 --lambda=2.5 --lambda0=3 --naive=CAGGTGCAGCTGGTGCA
    GTCTGGGGCTGAGGTGAAGAAGCCTGGGGCCTCAGTGAAGGTCTCCTGCAAGGCTTCTGGATACACCTTCA
```

```
    CCGGCTACTATATGCACTGGGTGCGACAGGCCCCTGGACAAGGGCTTGAGTGGATGGGATGGATCAACCCTAA
    CAGTGGTGGCACAAACTATGCACAGAAGTTTCAGGGCAGGGTCACCATGACCAGGGACACGTCCATCAGCACA
    GCCTACATGGAGCTGAGCAGGCTGAGATCTGACGACACGGCCGTGTATTACTGTGCGAGAGGGCCATTCCCGA
    ATTACTATGGTACGGGGAGTTATTGGGGGGGTTTTGACTACTGGGGCCAGGGAACCCTGGTCACCGTCTCCTC
    A --experimental=<PATH TO FASTA FILE WITH EXPERIMENTAL GC SEQUENCES> --na
    iveIDexp=naive0 --quick &> HTS_neut_sim.log

scons --simulate --gctree --igphyml --dnaml --frame=1 --outdir=single_cell_a
    ff_sim --naive=ggacctagcctcgtgaaaccttctcagactctgtccctcacctgttctgtcactggcg
    actccatcaccagtggttactggaactggatccggaaattcccagggaataaacttgagtacatggggtacat
    aagctacagtggtagcacttactacaatccatctctcaaaagtcgaatctccatcactcgagacacatccaag
    aaccagtactacctgcagttgaattctgtgactactgaggacacagccacatattactgt --experiment
    al=<PATH TO FASTA FILE WITH EXPERIMENTAL GC SEQUENCES> --naiveIDexp=na
    ive0 --lambda0=0.25 --selection --target_dist=5 --target_count=100 --
    verbose --T=35 --nsim=100 --n=65 &> single_cell_aff_sim.log

scons --simulate --gctree --igphyml --dnaml --frame=1 --outdir=
    single_cell_neut_sim --nsim=100 --N=100 --n=70 --lambda=1.5 --lambda0=.25
     --naive=ggacctagcctcgtgaaaccttctcagactctgtccctcacctgttctgtcactggcgactcca
    tcaccagtggttactggaactggatccggaaattcccagggaataaacttgagtacatggggtacataagcta
    cagtggtagcacttactacaatccatctctcaaaagtcgaatctccatcactcgagacacatccaagaacca
    gtactacctgcagttgaattctgtgactactgaggacacagccacatattactgt --experimental=<P
    ATH TO FASTA FILE WITH EXPERIMENTAL GC SEQUENCES> --naiveIDexp=naive0 &>
    single_cell_neut_sim.log
```

Appendix D

Table of abbreviations

Abbreviation	Full name
AID	Activation-induced cytidine deaminase
ASR	Ancestral sequence reconstruction
BC	Barcode
BCR	B cell receptor
BFGS	Broyden–Fletcher–Goldfarb–Shanno
CDF	Cumulative distribution function
CDR	Complementarity defining region
COAR	Correctness of ancestral reconstruction
CSTR	Continuously stirred tank reactor
DZ	Dark zone
FDC	Follicular dendritic cell
FR	Framework region
GC	Germinal center
GCtree	Genotype collapsed tree
GTR	General time reversible
HDI	High density interval
HMM	Hidden Markov model
HTS	High-throughput sequencing
LZ	Light zone
MAC	Macrophage
MAP	Maximum a posteriori
ML	Maximum likelihood
MP	Maximum parsimony
MHCI	Major histocompatibility complex class I
MHCII	Major histocompatibility complex class II
MRCA	Most recent common ancestor
MWU	Mann–Whitney U

Table D.1: List of abbreviations.

Abbreviation	Full name
NGS	Next generation sequencing
NK	Natural killer
NNI	Nearest neighbor interchange
PBMC	Peripheral blood mononuclear cell
PMN	Polymorphonuclear leucocyte
RF	Robinson–Foulds
SRP	Subtree pruning and regrafting
TCR	T cell receptor
Tfh	T follicular helper
TBR	Tree bisection and reconnection
SHM	Somatic hypermutation
VDJ	Variable, diversifying and joining

Table D.2: List of abbreviations, continued.

Bibliography

[1] Simon Andrews et al. Fastqc: a quality control tool for high throughput sequence data, 2010.

[2] Irene Balelli, Vuk Milišić, and Gilles Wainrib. Multi-type galton-watson processes with affinity-dependent selection applied to antibody affinity maturation. Sep 2016.

[3] Oliver Bannard and Jason G. Cyster. Germinal centers: programmed for affinity maturation and antibody diversification. *Current opinion in immunology*, 45:21–30, Jan 2017.

[4] M Barak, NS Zuckerman, H Edelman, R Unger, and R Mehr. IgTree (c) : Creating immunoglobulin variable region gene lineage trees. *Journal of Immunological Methods*, 338(1-2):67–74, 2008.

[5] Facundo D Batista and Naomi E Harwood. The who, how and where of antigen presentation to b cells. *Nature Reviews Immunology*, 9(1):15–27, 2009.

[6] Claudia Berek and Cesar Milstein. Mutation drift and repertoire shift in the maturation of the immune response. *Immunological reviews*, 96(1):23–41, 1987.

[7] Scott D Boyd, Jason D Merker, James L Zehnder, and Andrew Z Fire. High-throughput sequencing for diagnosis, prognosis and monitoring of lymphoid malignancies. *Blood*, 112(11):3779–3779, 2008.

[8] Adrian W Briggs, Stephen J Goldfless, Sonia Timberlake, Brian J Belmont, Christopher R Clouser, David Koppstein, Devin Sok, Jason Vander A Heiden, Manu V Tamminen, Steven H Kleinstein, Dennis R Burton, George M Church, and Francois Vigneault. Tumor-infiltrating immune repertoires captured by single-cell barcoding in emulsion. *bioRxiv*, 2017.

[9] Peter J Campbell, Erin D Pleasance, Philip J Stephens, Ed Dicks, Richard Rance, Ian Goodhead, George A Follows, Anthony R Green, P Andy Futreal, and Michael R Stratton. Subclonal phylogenetic structures in cancer revealed by ultra-deep sequencing. *Proceedings of the National Academy of Sciences*, 105(35):13081–13086, 2008.

[10] Sidhartha Chaudhury, Jaques Reifman, and Anders Wallqvist. Simulation of b cell affinity maturation explains enhanced antibody cross-reactivity induced by the polyvalent malaria vaccine ama1. *Journal of immunology*, 193(5):2073–2086, Sep 2014.

[11] Lauren M. Childs, Edward B. Baskerville, and Sarah Cobey. Trade-offs in antibody repertoires to complex antigens. *Philosophical transactions of the Royal Society of London. Series B, Biological sciences*, 370(1676), Sep 2015.

[12] Cyrus Chothia and Arthur M Lesk. Canonical structures for the hypervariable regions of immunoglobulins. *Journal of molecular biology*, 196(4):901–917, 1987.

[13] David R Cox. Regression models and life-tables. In *Breakthroughs in statistics*, pages 527–541. Springer, 1992.

[14] Ang Cui, Roberto Di Niro, Jason A Vander Heiden, Adrian W Briggs, Kris Adams, Tamara Gilbert, Kevin C O'Connor, Francois Vigneault, Mark J Shlomchik, and Steven H Kleinstein. A model of somatic hypermutation targeting in mice based on high-throughput ig sequencing data. *The Journal of Immunology*, 197(9):3566–3574, 2016.

[15] MO Dayhoff, RM Schwartz, and BC Orcutt. 22 a model of evolutionary change in proteins. In *Atlas of protein sequence and structure*, volume 5, pages 345–352. National Biomedical Research Foundation Silver Spring, MD, 1978.

[16] William S DeWitt, Paul Lindau, Thomas M Snyder, Anna M Sherwood, Marissa Vignali, Christopher S Carlson, Philip D Greenberg, Natalie Duerkopp, Ryan O Emerson, and Harlan S Robins. A public database of memory and naive b-cell receptor sequences. *PloS one*, 11(8):e0160853, 2016.

[17] Nicole A. Doria-Rose, Chaim A. Schramm, Jason Gorman, Penny L. Moore, Jinal N. Bhiman, Brandon J. DeKosky, Michael J. Ernandes, Ivelin S. Georgiev, Helen J. Kim, Marie Pancera, Ryan P. Staupe, Han R. Altae-Tran, Robert T. Bailer, Ema T. Crooks, Albert Cupo, Aliaksandr Druz, Nigel J. Garrett, Kam H. Hoi, Rui Kong, Mark K. Louder, Nancy S. Longo, Krisha McKee, Molati Nonyane, Sijy O'Dell, Ryan S. Roark, Rebecca S. Rudicell, Stephen D. Schmidt, Daniel J. Sheward, Cinque Soto, Constantinos Kurt Wibmer, Yongping Yang, Zhenhai Zhang, Nisc Comparative Sequencing, James C. Mullikin, James M. Binley, Rogier W. Sanders, Ian A. Wilson, John P. Moore, Andrew B. Ward, George Georgiou, Carolyn Williamson, Salim S. Abdool Karim, Lynn Morris, Peter D. Kwong, Lawrence Shapiro, and John R. Mascola. Developmental pathway for potent V1V2-directed HIV-neutralizing antibodies. *Nature*, 2 March 2014.

[18] James Dunbar and Charlotte M Deane. Anarci: antigen receptor numbering and receptor classification. *Bioinformatics*, 32(2):298–300, 2016.

[19] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[20] Herman N Eisen and Gregory W Siskind. Variations in affinities of antibodies during the immune response. *Biochemistry*, 3(7):996–1008, 1964.

[21] Yuval Elhanati, Zachary Sethna, Quentin Marcou, Curtis G Callan, Thierry Mora, and Aleksandra M Walczak. Inferring processes underlying b-cell repertoire diversity. *Phil. Trans. R. Soc. B*, 370(1676):20140243, 2015.

[22] Rieckmann et al. 2017. Protein copy number estimates found by mass spec. Web resource for mass spec. data, January 2017. http://www.immprot.org/.

[23] Joseph Felsenstein. Cases in which parsimony or compatibility methods will be positively misleading. *Systematic zoology*, pages 401–410, 1978.

[24] Joseph Felsenstein. The number of evolutionary trees. *Systematic Biology*, 27(1):27–33, 1978.

[25] Joseph Felsenstein. Evolutionary trees from dna sequences: a maximum likelihood approach. *Journal of molecular evolution*, 17(6):368–376, 1981.

[26] Walter M Fitch. Toward defining the course of evolution: minimum change for a specific tree topology. *Systematic Biology*, 20(4):406–416, 1971.

[27] Sergey Fomel and Gilles Hennenfent. Reproducible computational experiments using scons. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1257. IEEE, 2007.

[28] Gregory M Frank, Davide Angeletti, William L Ince, James S Gibbs, Surender Khurana, Adam K Wheatley, Edward E Max, Adrian B McDermott, Hana Golding, James Stevens, et al. A simple flow-cytometric method measuring b cell surface immunoglobulin avidity enables characterization of affinity maturation to influenza a virus. *mBio*, 6(4):e01156–15, 2015.

[29] George Georgiou, Gregory C Ippolito, John Beausang, Christian E Busse, Hedda Wardemann, and Stephen R Quake. The promise and challenge of high-throughput sequencing of the antibody repertoire. *Nature biotechnology*, 32(2):158–168, 2014.

[30] Manuel Gil, Marcelo Serrano Zanetti, Stefan Zoller, and Maria Anisimova. Codonphyml: fast maximum likelihood phylogeny estimation under codon substitution models. *Molecular biology and evolution*, page mst034, 2013.

[31] Veronique Giudicelli and Marie-Paule Lefranc. Ontology for immunogenetics: the imgt-ontology. *Bioinformatics*, 15(12):1047–1054, 1999.

[32] Jacob Glanville, Tracy C Kuo, H-Christian von Büdingen, Lin Guey, Jan Berka, Purnima D Sundar, Gabriella Huerta, Gautam R Mehta, Jorge R Oksenberg, Stephen L Hauser, et al. Naive antibody gene-segment frequencies are heritable and unaltered by chronic lymphocyte ablation. *Proceedings of the National Academy of Sciences*, 108(50):20066–20071, 2011.

[33] Jacob Glanville, Wenwu Zhai, Jan Berka, Dilduz Telman, Gabriella Huerta, Gautam R Mehta, Irene Ni, Li Mei, Purnima D Sundar, Giles MR Day, et al. Precise determination of the diversity of a combinatorial antibody library gives insight into the human immunoglobulin repertoire. *Proceedings of the National Academy of Sciences*, 106(48):20216–20221, 2009.

[34] Nick Goldman and Ziheng Yang. A codon-based model of nucleotide substitution for protein-coding dna sequences. *Molecular biology and evolution*, 11(5):725–736, 1994.

[35] Lizhi Ian Gong, Marc A Suchard, and Jesse D Bloom. Stability-mediated epistasis constrains the evolution of an influenza protein. *Elife*, 2:e00631, 2013.

[36] Namita T Gupta, Kristofor D Adams, Adrian W Briggs, Sonia C Timberlake, Francois Vigneault, and Steven H Kleinstein. Hierarchical clustering can identify b cell clones with high confidence in ig repertoire sequencing data. *The Journal of Immunology*, 198(6):2489–2499, 2017.

[37] Theodore E Harris. *The theory of branching processes*. Courier Corporation, 2002.

[38] Michael D Hendy and David Penny. Branch and bound algorithms to determine minimal evolutionary trees. *Mathematical Biosciences*, 59(2):277–290, 1982.

[39] Steven Henikoff and Jorja G Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences*, 89(22):10915–10919, 1992.

[40] Kenneth Hoehn, Gerton Lunter, and Oliver Pybus. A phylogenetic codon substitution model for antibody lineages. 28 September 2016.

[41] Annemarie Honegger and Andreas Plückthun. Yet another numbering scheme for immunoglobulin variable domains: an automatic modeling and analysis tool. *Journal of molecular biology*, 309(3):657–670, 2001.

[42] Jinghe Huang, Byong H Kang, Elise Ishida, Tongqing Zhou, Trevor Griesman, Zizhang Sheng, Fan Wu, Nicole A Doria-Rose, Baoshan Zhang, Krisha McKee, et al. Identification of a cd4-binding-site antibody to hiv that evolved near-pan neutralization breadth. *Immunity*, 45(5):1108–1121, 2016.

[43] Jaime Huerta-Cepas, François Serra, and Peer Bork. Ete 3: Reconstruction, analysis, and visualization of phylogenomic data. *Molecular biology and evolution*, 33(6):1635–1638, 2016.

[44] Ning Jiang, Jiankui He, Joshua A Weinstein, Lolita Penland, Sanae Sasaki, Xiao-Song He, Cornelia L Dekker, Nai-Ying Zheng, Min Huang, Meghan Sullivan, et al. Lineage structure of the human antibody repertoire in response to influenza vaccination. *Science translational medicine*, 5(171):171ra19–171ra19, 2013.

[45] Kazutaka Katoh and Daron M Standley. Mafft multiple sequence alignment software version 7: improvements in performance and usability. *Molecular biology and evolution*, 30(4):772–780, 2013.

[46] Thomas B Kepler. Reconstructing a b-cell clonal lineage. i. statistical inference of unobserved ancestors. *F1000Res.*, 2:103, 3 April 2013.

[47] Steven H Kleinstein, Yoram Louzoun, and Mark J Shlomchik. Estimating hypermutation rates from clonal tree data. *The Journal of Immunology*, 171(9):4639–4649, 2003.

[48] FGM Kroese, W Timens, and P Nieuwenhuis. Germinal center reaction and b lymphocytes: morphology and function. In *Reaction Patterns of the lymph node*, pages 103–148. Springer, 1990.

[49] Masayuki Kuraoka, Aaron G. Schmidt, Takuya Nojima, Feng Feng, Akiko Watanabe, Daisuke Kitamura, Stephen C. Harrison, Thomas B. Kepler, and Garnett Kelsoe. Complex antigens drive permissive clonal selection in germinal centers. *Immunity*, 44(3):542–552, Mar 2016.

[50] Peter D Kwong, Gwo-Yu Chuang, Brandon J DeKosky, Tatyana Gindin, Ivelin S Georgiev, Thomas Lemmin, Chaim A Schramm, Zizhang Sheng, Cinque Soto, An-Suei Yang, et al. Antibodyomics: bioinformatics technologies for understanding b-cell immunity to hiv-1. *Immunological Reviews*, 275(1):108–128, 2017.

[51] James W Larrick, Lena Danielsson, Carol A Brenner, Ellen F Wallace, Magnus Abrahamson, Kirk E Fry, and Carl AK Borrebaeck. Polymemse chain reaction using mixed primers: Cloning of human monoclonal antibody variable region genes from single hybridoma cells. *Nature Biotechnology*, 7(9):934–938, 1989.

[52] Andreas H Laustsen, Mikael Engmark, Christopher Clouser, Sonia Timberlake, Francois Vigneault, José María Gutiérrez, and Bruno Lomonte. Exploration of immunoglobulin transcriptomes from mice immunized with three-finger toxins and phospholipases a2 from the central american coral snake, micrurus nigrocinctus. *PeerJ*, 5:e2924, 2017.

[53] M-P Lefranc. Nomenclature of the human immunoglobulin heavy (igh) genes. *Experimental and clinical immunogenetics*, 18(2):100–116, 2001.

[54] Marie-Paule Lefranc, Christelle Pommié, Manuel Ruiz, Véronique Giudicelli, Elodie Foulquier, Lisa Truong, Valérie Thouvenin-Contet, and Gérard Lefranc.

Imgt unique numbering for immunoglobulin and t cell receptor variable domains and ig superfamily v-like domains. *Developmental & Comparative Immunology*, 27(1):55–77, 2003.

[55] Hanjie Li, Congting Ye, Guoli Ji, Xiaohui Wu, Zhe Xiang, Yuanyue Li, Yonghao Cao, Xiaolong Liu, Daniel C Douek, David A Price, et al. Recombinatorial biases and convergent recombination determine interindividual tcrβ sharing in murine thymocytes. *The Journal of Immunology*, 189(5):2404–2413, 2012.

[56] Shuo Li, Marie-Paule Lefranc, John J Miles, Eltaf Alamyar, Véronique Giudicelli, Patrice Duroux, J Douglas Freeman, Vincent DA Corbin, Jean-Pierre Scheerlinck, Michael A Frohman, et al. Imgt/highv quest paradigm for t cell receptor imgt clonotype diversity and next generation repertoire immunoprofiling. *Nature communications*, 4, 2013.

[57] Man Liu, Jamie L Duke, Daniel J Richter, Carola G Vinuesa, Christopher C Goodnow, Steven H Kleinstein, and David G Schatz. Two levels of protection for the b cell genome during somatic hypermutation. *Nature*, 451(7180):841–845, 2008.

[58] Andrew CR Martin. Protein sequence and structure analysis of antibody variable domains. *Antibody engineering*, pages 33–51, 2010.

[59] Connor O McCoy, Aaron Gallagher, Noah G Hoffman, and Frederick A Matsen. nestly—a framework for running software with nested parameter choices and aggregating results. *Bioinformatics*, 29(3):387–388, 2012.

[60] Jonathan R McDaniel, Brandon J DeKosky, Hidetaka Tanno, Andrew D Ellington, and George Georgiou. Ultra-high-throughput sequencing of the immune receptor repertoire from millions of lymphocytes. *Nature protocols*, 11(3):429–442, 2016.

[61] Alexander Mirsky, Linda Kazandjian, and Maria Anisimova. Antibody-specific model of amino acid substitution for immunological inferences from alignments of antibody sequences. *Molecular biology and evolution*, page msu340, 2014.

[62] K Murphy, P Travers, M Walport, and C Janeway. Immunobiology. 7th. *New York: Garland Science*, 2008.

[63] Anand Murugan, Thierry Mora, Aleksandra M Walczak, and Curtis G Callan. Statistical inference of the generation probability of t-cell receptors from sequence repertoires. *Proceedings of the National Academy of Sciences*, 109(40):16161–16166, 2012.

[64] Spencer V Muse and Brandon S Gaut. A likelihood approach for comparing synonymous and nonsynonymous nucleotide substitution rates, with application to the chloroplast genome. *Molecular biology and evolution*, 11(5):715–724, 1994.

[65] Saul B Needleman and Christian D Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[66] National Institutes of Health, Elvin Abraham Kabat, et al. *Sequences of proteins of immunological interest*. National Institutes of Health, 1983.

[67] Phuong Pham, Samir A. Afif, Mayuko Shimoda, Kazuhiko Maeda, Nobuo Sakaguchi, Lars C. Pedersen, and Myron F. Goodman. Structural analysis of the activation-induced deoxycytidine deaminase required in immunoglobulin diversification. *DNA repair*, 43:48–56, Jul 2016.

[68] Phuong Pham, Ronda Bransteitter, John Petruska, and Myron F Goodman. Processive aid-catalysed cytosine deamination on single-stranded dna simulates somatic hypermutation. *Nature*, 424(6944):103–107, 2003.

[69] Tri Giang Phan, Didrik Paus, Tyani D Chan, Marian L Turner, Stephen L Nutt, Antony Basten, and Robert Brink. High affinity germinal center b cells are actively selected into the plasma cell compartment. *Journal of Experimental Medicine*, 203(11):2419–2424, 2006.

[70] DOTREE Plotree and DOTGRAM Plotgram. Phylip-phylogeny inference package (version 3.2). *cladistics*, 5(163):6, 1989.

[71] Tal Pupko, Itsik Pe, Ron Shamir, and Dan Graur. A fast algorithm for joint reconstruction of ancestral amino acid sequences. *Molecular Biology and Evolution*, 17(6):890–896, 2000.

[72] Duncan K Ralph and Frederick A Matsen IV. Consistency of vdj rearrangement and substitution parameters enables accurate b cell receptor sequence annotation. *PLoS Comput Biol*, 12(1):e1004409, 2016.

[73] Duncan K Ralph and Frederick A Matsen IV. Likelihood-based inference of b cell clonal families. *PLoS Comput Biol*, 12(10):e1005086, 2016.

[74] Ryan N Randall, Caelan E Radford, Kelsey A Roof, Divya K Natarajan, and Eric A Gaucher. An experimental phylogeny to benchmark ancestral sequence reconstruction. *Nature Communications*, 7, 2016.

[75] Sai T Reddy, Xin Ge, Aleksandr E Miklos, Randall A Hughes, Seung Hyun Kang, Kam Hon Hoi, Constantine Chrysostomou, Scott P Hunicke-Smith, Brent L Iverson, Philip W Tucker, et al. Monoclonal antibodies isolated without screening by analyzing the variable-gene repertoire of plasma cells. *Nature biotechnology*, 28(9):965–969, 2010.

[76] Polina Reshetova, Barbera D. C. van Schaik, Paul L. Klarenbeek, Marieke E. Doorenspleet, Rebecca E. E. Esveldt, Paul-Peter Tak, Jeroen E. J. Guikema, Niek de Vries, and Antoine H. C. van Kampen. Computational model reveals limited correlation between germinal center b-cell subclone abundancy

and affinity: Implications for repertoire sequencing. *Frontiers in immunology*, 8, 2017.

[77] Jan C Rieckmann, Roger Geiger, Daniel Hornburg, Tobias Wolf, Ksenya Kveler, David Jarrossay, Federica Sallusto, Shai S Shen-Orr, Antonio Lanzavecchia, Matthias Mann, et al. Social network architecture of human immune cells unveiled by quantitative proteomics. *Nature Immunology*, 2017.

[78] Philippe A Robert, Ananya Rastogi, Sebastian C Binder, and Michael Meyer-Hermann. How to simulate a germinal center. *Germinal Centers: Methods and Protocols*, pages 303–334, 2017.

[79] David F Robinson and Leslie R Foulds. Comparison of phylogenetic trees. *Mathematical biosciences*, 53(1-2):131–147, 1981.

[80] Sebastien Roch. Branching processes. Course work web page, September 2015. https://www.math.wisc.edu/~roch/mdp/roch-mdp-chap5.pdf.

[81] T. Romppanen. A morphometrical method for analyzing germinal centers in the chicken spleen. *Acta pathologica et microbiologica Scandinavica. Section C, Immunology*, 89(4):263–268, Aug 1981.

[82] Aaron M Rosenfeld, Wenzhao Meng, Eline T Luning Prak, and Uri Hershberg. Immunedb: a system for the analysis and exploration of high-throughput adaptive immune receptor sequencing data. *Bioinformatics*, 33(2):292–293, 2017.

[83] Daniel E Russ, Kwan-Yuet Ho, and Nancy S Longo. Htjoinsolver: Human immunoglobulin vdj partitioning using approximate dynamic programming constrained by conserved motifs. *BMC bioinformatics*, 16(1):170, 2015.

[84] Yana Safonova, Alla Lapidus, and Jennie Lill. Igsimulator: a versatile immunosequencing simulator. *Bioinformatics*, page btv326, 2015.

[85] Cathrine Scheepers, Ram K Shrestha, Bronwen E Lambson, Katherine JL Jackson, Imogen A Wright, Dshanta Naicker, Mark Goosen, Leigh Berrie, Arshad Ismail, Nigel Garrett, et al. Ability to develop broadly neutralizing hiv-1 antibodies is not restricted by the germline ig gene repertoire. *The Journal of Immunology*, 194(9):4371–4378, 2015.

[86] Gitit Shahaf, Michal Barak, Neta S Zuckerman, Naamah Swerdlin, Malka Gorfine, and Ramit Mehr. Antigen-driven selection in germinal centers as reflected by the shape characteristics of immunoglobulin gene lineage trees: a large-scale simulation study. *Journal of theoretical biology*, 255(2):210–222, 2008.

[87] Gitit Shahaf, Michal Barak, Neta S. Zuckerman, Naamah Swerdlin, Malka Gorfine, and Ramit Mehr. Antigen-driven selection in germinal centers as reflected by the shape characteristics of immunoglobulin gene lineage trees: a

large-scale simulation study. *Journal of theoretical biology*, 255(2):210–222, Nov 2008.

[88] David F Shanno. On broyden-fletcher-goldfarb-shanno method. *Journal of Optimization Theory and Applications*, 46(1):87–94, 1985.

[89] Zizhang Sheng, Chaim A Schramm, Rui Kong, James C Mullikin, John R Mascola, Peter D Kwong, Lawrence Shapiro, NISC Comparative Sequencing Program, et al. Gene-specific substitution profiles describe the types and frequencies of amino acid changes during antibody somatic hypermutation. *Frontiers in Immunology*, 8, 2017.

[90] MJ Shlomchik, P Watts, MG Weigert, and S Litwin. Clone: a monte-carlo computer simulation of b cell clonal expansion, somatic mutation, and antigen-driven selection. In *Somatic Diversification of Immune Responses*, pages 173–197. Springer, 1998.

[91] Alexandros Stamatakis, Paul Hoover, Jacques Rougemont, and Susanne Renner. A rapid bootstrap algorithm for the raxml web servers. *Systematic biology*, 57(5):758–771, 2008.

[92] Joel NH Stern, Gur Yaari, Jason A Vander Heiden, George Church, William F Donahue, Rogier Q Hintzen, Anita J Huttner, Jon D Laman, Rashed M Nagra, Alyssa Nylander, et al. B cells populating the multiple sclerosis brain mature in the draining cervical lymph nodes. *Science translational medicine*, 6(248):248ra107–248ra107, 2014.

[93] Marc A Suchard and Benjamin D Redelings. Bali-phy: simultaneous bayesian inference of alignment and phylogeny. *Bioinformatics*, 22(16):2047–2048, 2006.

[94] Kazuhiro Suzuki, Irina Grigorova, Tri Giang Phan, Lisa M Kelly, and Jason G Cyster. Visualizing b cell capture of cognate antigen from follicular dendritic cells. *Journal of Experimental Medicine*, 206(7):1485–1493, 2009.

[95] Koichiro Tamura, Daniel Peterson, Nicholas Peterson, Glen Stecher, Masatoshi Nei, and Sudhir Kumar. Mega5: molecular evolutionary genetics analysis using maximum likelihood, evolutionary distance, and maximum parsimony methods. *Molecular biology and evolution*, 28(10):2731–2739, 2011.

[96] Jeroen MJ Tas, Luka Mesin, Giulia Pasqual, Sasha Targ, Johanne T Jacobsen, Yasuko M Mano, Casie S Chen, Jean-Claude Weill, Claude-Agnès Reynaud, Edward P Browne, et al. Visualizing antibody affinity maturation in germinal centers. *Science*, 351(6277):1048–1054, 2016.

[97] Simon Tavaré. Some probabilistic and statistical problems in the analysis of dna sequences. *Lectures on mathematics in the life sciences*, 17:57–86, 1986.

[98] MA Turchaninova, A Davydov, OV Britanova, M Shugay, V Bikos, ES Egorov, VI Kirgizova, EM Merzlyak, DB Staroverov, DA Bolotin, et al. High-quality full-length immunoglobulin profiling with unique molecular barcoding. *Nature Protocols*, 11(9):1599–1616, 2016.

[99] Helle D Ulrich, Emily Mundorff, Bernard D Santarsiero, Edward M Driggers, Raymond C Stevens, and Peter G Schultz. The interplay between binding energy and catalysis in the evolution of a catalytic antibody. *Nature*, 389(6648):271–275, 1997.

[100] Jason A Vander Heiden, Panos Stathopoulos, Julian Q Zhou, Luan Chen, Tamara J Gilbert, Christopher R Bolen, Richard J Barohn, Mazen M Dimachkie, Emma Ciafaloni, Teresa J Broering, et al. Dysregulation of b cell repertoire formation in myasthenia gravis patients revealed through deep sequencing. *The Journal of Immunology*, 198(4):1460–1473, 2017.

[101] Jason A Vander Heiden, Gur Yaari, Mohamed Uduman, Joel NH Stern, Kevin C O'Connor, David A Hafler, Francois Vigneault, and Steven H Kleinstein. presto: a toolkit for processing high-throughput sequencing raw reads of lymphocyte receptor repertoires. *Bioinformatics*, page btu138, 2014.

[102] Gabriel D Victora and Luka Mesin. Clonal and cellular dynamics in germinal centers. *Current opinion in immunology*, 28:90–96, 2014.

[103] Gabriel D Victora and Michel C Nussenzweig. Germinal centers. *Annual review of immunology*, 30:429–457, 2012.

[104] Shenshen Wang, Jordi Mata-Fink, Barry Kriegsman, Melissa Hanson, Darrell J Irvine, Herman N Eisen, Dennis R Burton, K Dane Wittrup, Mehran Kardar, and Arup K Chakraborty. Manipulating the selection forces during affinity maturation to generate cross-reactive hiv antibodies. *Cell*, 160(4):785–797, 2015.

[105] Shenshen Wang, Jordi Mata-Fink, Barry Kriegsman, Melissa Hanson, Darrell J. Irvine, Herman N. Eisen, Dennis R. Burton, K. Dane Wittrup, Mehran Kardar, and Arup K. Chakraborty. Manipulating the selection forces during affinity maturation to generate cross-reactive hiv antibodies. *Cell*, 160(4):785–797, Feb 2015.

[106] Jr. William Green. course materials for 10.37 chemical and biological reaction engineering. MIT OpenCourseWare (http://ocw.mit.edu), June 2017. https://ocw.mit.edu/courses/chemical-engineering/10-37-chemical-and-biological-reaction-engineering-spring-2007/lecture-notes/lec05_02212007_g.pdf.

[107] C Wilson, R V Agafonov, M Hoemberger, S Kutter, A Zorba, J Halpin, V Buosi, R Otten, D Waterman, D L Theobald, and D Kern. Kinase dynamics. using ancient protein kinases to unravel a modern cancer drug's mechanism. *Science*, 347(6224):882–886, 20 February 2015.

[108] Xueling Wu, Tongqing Zhou, Jiang Zhu, Baoshan Zhang, Ivelin Georgiev, Charlene Wang, Xuejun Chen, Nancy S Longo, Mark Louder, Krisha McKee, Sijy O'Dell, Stephen Perfetto, Stephen D Schmidt, Wei Shi, Lan Wu, Yongping Yang, Zhi-Yong Yang, Zhongjia Yang, Zhenhai Zhang, Mattia Bonsignori, John A Crump, Saidi H Kapiga, Noel E Sam, Barton F Haynes, Melissa Simek, Dennis R Burton, Wayne C Koff, Nicole A Doria-Rose, Mark Connors, NISC Comparative Sequencing Program, James C Mullikin, Gary J Nabel, Mario Roederer, Lawrence Shapiro, Peter D Kwong, and John R Mascola. Focused evolution of HIV-1 neutralizing antibodies revealed by structures and deep sequencing. *Science*, 333(6049):1593–1602, 16 September 2011.

[109] Gur Yaari, Jennifer IC Benichou, Jason A Vander Heiden, Steven H Kleinstein, and Yoram Louzoun. The mutation patterns in b-cell immunoglobulin receptors reflect the influence of selection acting at multiple time-scales. *Phil. Trans. R. Soc. B*, 370(1676):20140242, 2015.

[110] Gur Yaari, Jason A Vander Heiden, Mohamed Uduman, Daniel Gadala-Maria, Namita Gupta, Joel NH Stern, Kevin C O'Connor, David A Hafler, Uri Laserson, Francois Vigneault, et al. Models of somatic hypermutation targeting and substitution based on synonymous mutations from high-throughput immunoglobulin sequencing data. *Frontiers in immunology*, 4, 2013.

[111] Masao Yamada, Robert Wasserman, Betty Anne Reichard, Sara Shane, Andrew J Caton, and Giovanni Rovera. Preferential utilization of specific immunoglobulin heavy chain diversity and joining segments in adult human peripheral blood b lymphocytes. *J Exp med*, 173(2):395–407, 1991.

[112] Jian Ye, Ning Ma, Thomas L Madden, and James M Ostell. Igblast: an immunoglobulin variable domain sequence analysis tool. *Nucleic acids research*, page gkt382, 2013.

[113] Leng-Siew Yeap, Joyce K Hwang, Zhou Du, Robin M Meyers, Fei-Long Meng, Agnė Jakubauskaitė, Mengyuan Liu, Vinidhra Mani, Donna Neuberg, Thomas B Kepler, Jing H Wang, and Frederick W Alt. Sequence-Intrinsic mechanisms that target AID mutational outcomes on antibody genes. *Cell*, 2015.

[114] Leng-Siew Yeap, Joyce K. Hwang, Zhou Du, Robin M. Meyers, Fei-Long Meng, Agnė Jakubauskaitė, Mengyuan Liu, Vinidhra Mani, Donna Neuberg, Thomas B. Kepler, and et al. Sequence-intrinsic mechanisms that target aid mutational outcomes on antibody genes. *Cell*, 163(5):1124–1137, Nov 2015.

[115] Li Zhang, Jason Cham, Alan Paciorek, James Trager, Nadeem Sheikh, and Lawrence Fong. 3d: diversity, dynamics, differential testing–a proposed pipeline for analysis of next-generation sequencing t cell repertoire data. *BMC bioinformatics*, 18(1):129, 2017.

[116] Jiang Zhu, Gilad Ofek, Yongping Yang, Baoshan Zhang, Mark K Louder, Gabriel Lu, Krisha McKee, Marie Pancera, Jeff Skinner, Zhenhai Zhang, et al. Mining the antibodyome for hiv-1–neutralizing antibodies with next-generation sequencing and phylogenetic pairing of heavy/light chains. *Proceedings of the National Academy of Sciences*, 110(16):6470–6475, 2013.