Using B cell receptor lineage structures to predict affinity

Duncan K. Ralph[1,*], Frederick A. Matsen IV[1]

1 Fred Hutchinson Cancer Research Center, Seattle, Washington, USA
[*] dralph@fredhutch.org

ABSTRACT

We are frequently faced with a large collection of antibodies, and want to select those with highest affinity for their cognate antigen. When developing a first-line therapeutic for a novel pathogen, for instance, we might look for such antibodies in patients that have recovered. There exist effective experimental methods of accomplishing this, such as cell sorting and baiting; however they are time consuming and expensive. Next generation sequencing of B cell receptor (BCR) repertoires offers an additional source of sequences that could be tapped if we had a reliable method of selecting those coding for the best antibodies. In this paper we introduce a method that uses evolutionary information from the family of related sequences that share a naive ancestor to predict the affinity of each resulting antibody for its antigen. When combined with information on the identity of the antigen, this method should provide a source of effective new antibodies. We also introduce a method for a related task: given an antibody of interest and its inferred ancestral lineage, which branches in the tree are likely to harbor key affinity-increasing mutations? These methods are implemented as part of continuing development of the partis BCR inference package, available at `https://github.com/psathyrella/partis`.

Comments. Please post comments or questions on this paper as new issues at `https://git.io/Jvxkn`.

Introduction

Antibodies are the foundation of both vaccine-induced immunity and many important therapeutics. They stem from B cells through the processes of VDJ rearrangement and somatic hypermutation (SHM), which yield a vast repertoire of B cell receptors (BCRs) within each person. Each clonal family begins from a single naive B cell that has encountered its cognate antigen, which then reproduces in a germinal center (GC), diversifying and evolving as SHM drives affinity maturation. We can probe the BCR repertoire via next generation sequencing (NGS), and with computational methods divide it into groups of clonally-related sequences [1] stemming from the same rearrangement event (clonal families).

Of the many sequences in the BCR repertoire, we are generally interested in those that code for the highest affinity antibodies. There exist several effective methods of finding high affinity B cells experimentally. Both cell sorting and antigen baiting have been used to find a variety of important antibodies [2–4]. These approaches have several challenges. For instance, constructing a stable baiting antigen for certain conformational epitopes can be challenging. However their main limitation is that they require large investments of time and resources.

Deep sequencing of the BCR repertoire yields tens of thousands to millions of sequences from this same pool, some of which undoubtedly correspond to high affinity antibodies. And in many cases, this NGS data already exists, since it is frequently collected as a matter of course when studying, for instance, antibodies isolated using the cell sorting approach. If we had some way of identifying a handful of these sequences that are likely to correspond to high affinity antibodies, then they could be synthesized and tested for their binding properties. This could yield a rich source of novel antibodies.

In most practical cases we would like to choose only antibodies that are effective against a particular pathogen or epitope. Because the evolutionary signatures in sequence data can tell us much more about affinity than they can about specificity, we separate this task into two parts: finding antibodies or families with the desired specificity, and finding antibodies with high affinity regardless of specificity. While de novo epitope prediction from sequence data alone is a rapidly developing field, the challenges are still such that specificity determination is best accomplished

with non-sequence-based information, such as cell sorting or information about vaccine challenge. Because our expertise is solely in analyzing sequence data, the methods in this paper deal only with affinity prediction. However, to give an idea of what is possible, we also review techniques that have been used to enrich for particular specificities in the Discussion.

In our experience, there are two common use cases for methods of choosing antibodies from NGS data, depending on what prior information is available. If we have a previously-isolated antibody of interest, then specificity has already been determined and we want to choose only among the sequences in that antibody's family. In the absence of such an antibody, we are instead choosing from among all families in the repertoire, and our affinity prediction methods must be paired with some form of enriching for the desired specificity. While the second case is more difficult, because both affinity and specificity vary between families, it also holds the promise of much greater rewards, since novel antibodies from previously unknown families could bind with much higher affinity, or to new epitopes.

In this paper we first test a variety of methods to choose individual high-affinity sequences both from among all families in the repertoire, and from within single families. We find that an observed sequence's similarity to its family's amino acid consensus sequence (measured by hamming distance, and abbreviated aa-cdist) is a highly effective predictor of affinity in both of these cases. Because the other, more poorly performing metrics provide independent information not in aa-cdist, we also combined them using several machine learning approaches. We were not, however, successful in training such a method that substantially outperformed aa-cdist, which we believe is likely due to several unique features of the combined tree and sequence spaces.

We next introduce an entirely new metric to predict the change in affinity along each branch of a family's inferred phylogenetic tree (abbreviated Δ-affinity). This new metric, called local branching ratio (lbr), draws on ideas from [5] and [6]. It is designed for situations where we have an antibody of interest, and want to know which branches along its inferred lineage are likely to harbor the most important mutations.

Since we focus on B cells we refer to affinity; however these methods are in fact measuring evolutionary fitness, and are thus also of much wider applicability. For instance, while writing

this paper we discovered that some of the authors of [5] had concurrently found that aa-cdist is predictive of fitness in the context of viral evolution, and also derived a mathematical proof that aa-cdist is in some circumstances the optimal metric [7]. We would also expect lbr to predict fitness-increasing mutations in an inferred viral lineage just as effectively as it predicts affinity increases for antibodies.

These metrics are essentially evolutionary in nature, and thus require at least a handful of sequences from each clonal family. All of these metrics can be calculated with simple options in the freely available partis software package `https://github.com/psathyrella/partis`, which also performs BCR annotation, simulation, clonal family clustering, and per-sample germline inference [1, 8, 9]. However, we note that aa-cdist is simple enough that for users who already have a list of clonal sequences, a simple sequence alignment GUI may suffice.

RESULTS

Affinity and fitness. The metrics that we use to predict affinity and Δ-affinity for each sequence begin by considering that sequence in the context of its clonal family of related sequences. Whether this context is a phylogenetic tree or a sequence alignment, it provides information about the evolutionary history that led to the sequence's development. The metrics use this information to find the sequences stemming from cells that were fittest, in the evolutionary sense, during the GC reaction. Because a key purpose of the GC reaction is to apply selective pressure in order to direct evolution toward antibodies that bind to pathogens with high affinity, fitness is generally correlated with affinity. It is important to note, however, that this fitness also depends on other factors, such as the ability to recruit T cell help. We thus generally refer to "predicting affinity" in this paper, but it should be understood that the metrics are more directly connected to GC reaction fitness, and the extent to which this fitness correlates to affinity must be judged on a case by case basis, particularly in real data. In simulation, we define specific mathematical connections between each sequence's affinity and its expected number of offspring. The two real data samples in this paper, in contrast, measure neutralization concentration rather than affinity, and while these quantities are generally expected to correlate, we cannot say to what extent in any specific instance.

Metrics. We measured the ability of a variety of metrics to predict affinity and Δ-affinity, all of which are summarized in Table 1. Each is also described in the following paragraphs.

As mentioned above, each observed sequence's hamming distance to its family's amino acid consensus (abbreviated **aa-cdist**) is by far the best predictor of affinity (where smaller distance is better). While this metric has been used in an ad hoc way [10], to our knowledge the only attempt to measure its performance was [11], which tested on only a single, small family in the supplemental information. In many cases the calculated consensus sequence itself is not observed, and it is possible that in such cases this unobserved consensus would be a better predictor than the nearest observed sequence, but we have not evaluated this (see Discussion). When calculating the consensus, ties are treated as ambiguous positions, which are then ignored in the hamming distance.

Local branching index (abbreviated **lbi**) uses the family's phylogenetic tree to measure the "branchiness" in the local area around each node [5]. This branchiness is calculated by integrating the total branch length surrounding the node with a decaying exponential weighting factor (S1 Figure). While lbi performs worse than aa-cdist in almost all regions of parameter space, it is nevertheless important because it contains some independent information, provides an obvious path to improving aa-cdist, and serves as the basis for lbr. Our implementation of lbi includes several modifications to the original formulation. We perform an independent optimization of the τ locality parameter, although we arrive at a comparable final value. We also introduce a normalization scheme, which although amounting only to a switch to human-interpretable units, is important because without it lbi calculated in different papers with different τ values cannot be compared.

Because affinity maturation via somatic hypermutation should be, in the long run, an affinity-increasing process, many papers have used the number of somatic mutations as a proxy for affinity (abbreviated **n-shm**, and distinguished from our abbreviation SHM for the process). Unfortunately in practice this metric performs very poorly because it chooses leaves, which harbor many novel mutations that have not yet been evaluated by significant selective pressure. The underlying

idea that affinity increases as you move away from root is valid, however it is also necessary to go some way back toward root from the leaves (indeed this is what aa-cdist accomplishes).

In addition to aa-cdist, we also show results for its nucleotide analog **nuc-cdist**. This is useful mainly as a way to understand the importance of information from the amino acid translation table, and thus differences between aa-cdist and lbi.

An additional metric that we do not evaluate is the multiplicity of each unique sequence. It can be experimentally challenging to disentangle underlying cell numbers from other factors such as primer bias and varying expression levels [12–14]. Thus while higher multiplicity is sometimes used in practice to predict higher fitness [10], we do not pursue it here because many data sets that we encounter do not include the measures necessary for an accurate estimation.

To predict affinity-increasing mutations (Δ-affinity) we mainly focus on the local branching ratio **lbr**. This novel metric uses the same branch length integrals as lbi, but instead of summing in all directions, it compares the branchiness among the node's offspring to that of its parents and siblings (S1 Figure). This ratio of branchiness below vs above the node thus quantifies the possibility that an affinity-increasing mutation occurred along the branch immediately above the node, since such a mutation would increase fitness among the node's offspring.

In order to provide some baseline for the effectiveness of lbr, we also evaluate the change in lbi from the parent node (abbreviated **Δ-lbi**) as a predictor of Δ-affinity. While this functions adequately, it is always significantly worse than lbr.

Evaluation framework. The first step toward confidence in any method is measuring its performance in all corners of its parameter space. For the metrics in this paper, that space is constructed from sequences and trees and is vast, complicated, and high-dimensional. In order to measure performance, we began with a previously-described simulation framework [15], extending it to allow a more comprehensive variation of parameters, and rewriting to optimize for speed. We then performed scans across all reasonably plausible values of parameters that could affect performance.

The simulation begins by constructing a naive sequence via VDJ rearrangement. It then chooses a number of "target" sequences at some distance from the naive, each of which represents a potential optimal antibody. Repeated rounds of SHM then apply selective pressure to direct the naive sequence's offspring toward these targets.

We measure performance in a way that emphasizes the practical use case for these metrics: choosing a handful of sequences from deep sequencing data in order to invest substantial resources into synthesis and binding evaluation. While these metrics predict affinity, and are thus correlated with it, quantifying overall correlation with something like the Pearson coefficient would not measure effectiveness for our use case. This is because we only care about the very small fraction of sequences at the highest affinity values, whereas measures of overall correlation count all sequences equally and are thus dominated by low- and medium-affinity sequences. We instead imagine that we have chosen the top few sequences according to the predictive metric, and then ask whether they are also among the top few in affinity. We quantify this by taking the average of their affinity quantiles. We then compare this mean quantile to that of a hypothetical perfect method that simply ranks sequences by their true affinity. The amount by which the mean affinity quantile of sequences chosen by the actual metric deviates from that of the perfect metric, further averaged over quantiles from 75 to 100 (i.e. choosing between 25% and 0% of sequences), forms the basis of our performance evaluation.

We supplement these comprehensive simulation scans with validation results on two small real data samples.

Table 1. **Metrics used in this paper** to predict affinity (top) and Δ-affinity (bottom). Those marked with a * perform much better than the others, and are recommended for all practical use. The sign of the metric's correlation with the predicted quantity is indicated by "$\pm$ corr.".

affinity metrics	shorthand	$\pm$ corr.	description
AA consensus distance *	aa-cdist	-	hamming distance to the clonal family's amino acid consensus sequence
local branching index	lbi	+	total branch length in the local area of the tree
somatic hypermutations	n-shm	+	number of nucleotide somatic hypermutations
nucleotide consensus distance	nuc-cdist	-	nucleotide version of aa-cdist

Δ-affinity metrics	shorthand	$\pm$ corr.	description
local branching ratio *	lbr	+	ratio of total branch length in the tree below the node to above the node
change in lbi	Δ-lbi	+	change in local branching index compared to the parent node

Simulation results. We first show simulation performance for a single parameter scan for both affinity and Δ-affinity prediction (Fig 1 and Fig 2). This scan varies observation time (in units of N generations) from 50 to 3000 while holding other variables constant. This corresponds roughly to varying the mean frequency of somatic nucleotide mutations among observed sequences from 2% to 25%. We find that aa-cdist performs better than all other metrics throughout the scan, typically at 5-10% from perfect. Also of note, nuc-cdist performs similarly to, but slightly worse than, lbi for much of the scan. This is consistent with the hypothesis that lbi does a better job of quantifying branchiness/evolutionary density than does consensus distance, but that this advantage is overwhelmed by the large benefit to including amino acid information in aa-cdist. It is also notable that aa-cdist performs well both when choosing among all families and choosing within single families, while lbi suffers a much larger drop in the more challenging case of choosing among all families. Meanwhile n-shm performs much worse than the other metrics when choosing within a family.

In the Δ-affinity prediction (Fig 2) lbr performs much better than Δ-lbi, on average identifying a branch that is $0.1 - 0.2$ steps in the tree from the branch that actually has an affinity-increasing mutation. This can be thought of as choosing the correct branch eight or nine out of ten times, and being one branch away the other times.

Analogous scans for other parameters are shown separately for the cases of choosing among all families (S2 Figure) and within single families (S3 Figure). These scans correspond to varying the most important simulation parameters: longitudinal sampling, carrying capacity, number of sampled sequences, sequence-to-affinity mapping, and selection strength. The results generally follow the pattern of Fig 1, although we note another case where lbi performs very poorly but aa-cdist works well: low selection strength (neutral evolution). The corresponding plots for Δ-affinity are at `https://doi.org/10.5281/zenodo.3728068`; they are fairly uniform in showing lbr performing significantly better than Δ-lbi, and typically between 0.1 and 0.5 steps from perfect. We also show performance for a variety of different models of cell export from the GC (S4 Figure), as well as the effect of varying the number of "target sequences" (the simulation's representation of hypothetical ideal antibodies, S5 Figure).

Many other variations of these parameter scans, and plots comparing multiple slices for each metric, can be found at `https://doi.org/10.5281/zenodo.3728068`.

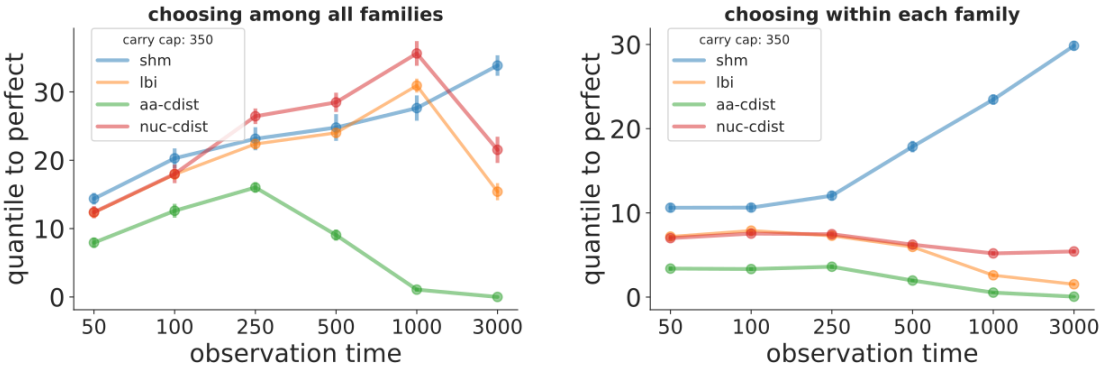


Fig 1. **Simulation performance for affinity prediction vs. observation time** when choosing sequences among all families in the repertoire (left) and within each family (right) for the metrics in Table 1. The y-axis describes how close each metric comes to achieving the best possible performance. For example a value of 5 "quantiles to perfect" for aa-cdist would mean that choosing the top 7% of sequences using aa-cdist would give you sequences that, on average, are in roughly the top 12% of affinity. An observation time of 50 (3000) generations corresponds to a nucleotide SHM frequency of roughly 2% (25%). The longer simulation times correspond to a series of GC reaction/reentry cycles, as would for instance be seen during anti-HIV bNAb development over several years. Each point shows the mean $\pm$ standard error of 30 samples, where each sample consists of 1500 sequences from 10 families. Similar plots across ranges of other simulation parameters can be found in S2 Figure, S3 Figure, S4 Figure, S5 Figure, and at `https://doi.org/10.5281/zenodo.3728068`.

Data results. Validation on real sequence data serves a different, complementary purpose to validation on simulation. Real data validation is more stringent, and thus more useful, in the sense that by definition it perfectly recapitulates the properties of real data. But on the other hand real data is less stringent, and thus less useful, because it is more difficult to produce and thus is only ever available for a very restricted set of parameter values. For instance looking at one real data sample might only explore large, high-mutation trees with strong selection, but ignore performance at other parameter values. This means that designing any method using real data validation alone is extremely risky, since it only provides information about how the method performs

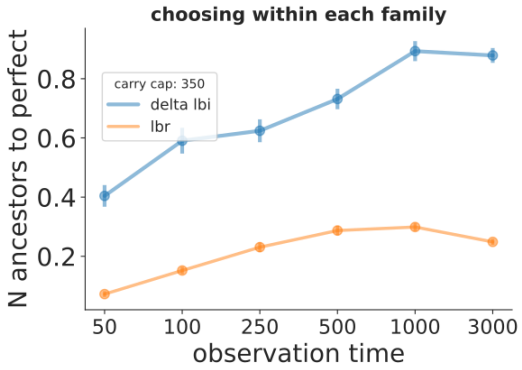


Fig 2. **Simulation performance for Δ-affinity prediction vs. observation time** for the metrics in Table 1. The y-axis describes how close each metric comes to achieving the best possible performance. Here we are predicting the location of affinity-increasing mutations in a lineage of inferred ancestral sequences, and for example a value of 0.5 "N ancestors to perfect" for lbr would mean that if we choose the best inferred ancestral sequence using lbr that we would, on average, be 0.5 branches away from a branch containing an actual affinity-increasing mutation. See Fig 1 caption for details. Similar plots across ranges of other simulation parameters can be found at `https://doi.org/10.5281/zenodo.3728068`.

for the particular combination of parameters in those data sets. In the present case, because affinity is so expensive to measure, real data has a further problem: while there are many papers with both NGS data and affinity information, to our knowledge none of them measure affinity for more than a handful of sequences. This is, nevertheless, certainly better than nothing, and can provide confidence that nothing has gone egregiously wrong in the simulation studies.

Here we have chosen data from two well-known papers on anti-HIV antibodies [16, 17]. There are several aspects of HIV that, a priori, we expected might make it a very challenging environment for these metrics. First, these papers both measure IC_{50} neutralization concentrations rather than affinity. To handle this we simply use $1/\mathrm{IC}_{50}$ in place of affinity, and hope that neutralization and affinity are correlated enough for the metrics to work. Second, HIV's vast diversity both globally and within each subject means that the viruses that applied selective pressure during each antibody's development were in all likelihood quite different from the viruses against which the antibodies' neutralization strengths were tested. These papers also report the geometric mean IC_{50}

over many viruses from the three main global HIV clades, and for the sake of simplicity we use these overall values for validation. As for almost all current work, these papers also use unpaired sequencing, which not only ignores any signal from light chain evolution, but also typically means measuring IC_{50} values for chimeric antibodies with non-native light chain sequences. Finally, we note that these are the only real data samples on which we have run our methods – it would of course have been uninformative to run on many different samples before deciding which to include in this paper.

With the final reminder that these are extremely small sample sizes, we find that on real data both aa-cdist (Fig 3) and lbi (S6 Figure) perform roughly in line with expectations from simulation. We find that aa-cdist is generally between 0 and 20 quantiles from perfect, while lbi ranges from 10 to 30.

Discussion

We have demonstrated effective, practical methods for two tasks common to the analysis of B cell receptor deep sequencing data. First, we find that a sequence's similarity to its clonal family's amino acid consensus (aa-cdist) is an excellent predictor of that sequence's affinity, and is a highly effective way to choose a handful of sequences for synthesis and testing. Second, we find that a new metric that we construct called local branching ratio (lbr) is similarly effective for the related task of predicting which branches in a single lineage are likely to harbor affinity-increasing mutations (i.e. predicting Δ-affinity). These metrics are both implemented in the existing partis annotation and clonal family inference package, available at `https://github.com/psathyrella/partis`. By testing performance on simulation generated with a wide variety of parameters, we showed that choosing the best few sequences according to these metrics will likely result in antibodies that are also among the highest affinity. We further find that the number of somatic hypermutations (n-shm), while a frequently-used heuristic affinity predictor [10], performs very poorly. We also showed performance on two very small real data samples, which provide a confidence-boosting cross check. We emphasize that we make no attempt at specificity prediction, since this is better achieved through non-sequence-based methods such as clinical information (e.g. use of

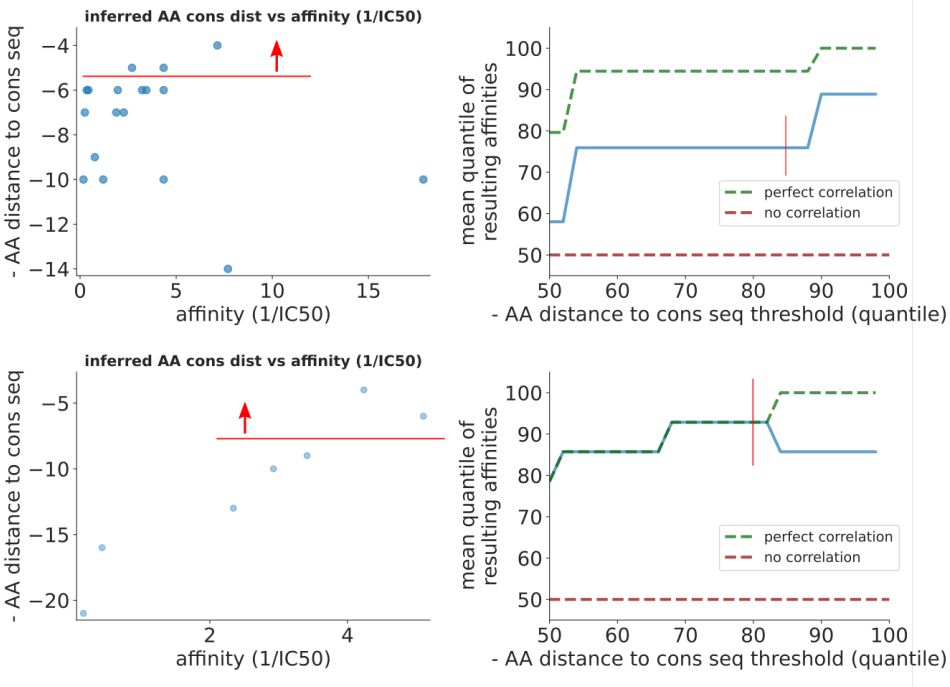


Fig 3. **Performance on real data for aa-cdist** from [16] (top) and [17] (bottom). The scatter plots (left) show the raw correlation between aa-cdist and measured "affinity" (here actually $1/\mathrm{IC}_{50}$), while the quantile plots (right) show the relative affinity ranking for sequences chosen using aa-cdist (blue) compared to perfect (green) and random chance (red). For instance, an x value of 85 on the top right plot corresponds to 75 on the y axis, meaning that if we chose the top 15% of sequences using aa-cdist that these sequences would on average rank in the top 25% by affinity (thin red lines). Another way of viewing this is that, at x of 85, the blue line is about 20% below the green line, meaning performance is 20% worse than perfect; this interpretation corresponds to the values shown in the top row of Fig 2. We also show the equivalent plots for lbi (S6 Figure).

vaccine challenge). Thus while aa-cdist is effective at choosing the highest affinity antibodies both from among all families and within each family, since specificity is typically important, it will often be necessary to combine it with such a non-sequence-based method.

There are many avenues for future improvement to these methods. We currently use the plain consensus, which by considering each site independently ignores information about how sequences evolved together at different sites. By taking a single family-wide consensus, it also does a poor job of handling families with several widely-separated branches. While its performance does

not drop significantly in such cases (this is tested by the variety of target sequence configurations in S5 Figure), there is likely room for improvement. When presented with widely-separated branches, the consensus calculation will in many cases simply ignore all but the largest branch. This is not ideal, since when evolving toward separate targets, the less-numerous branches might hold a significant number of effective antibodies. We could address this by adding locality to the consensus calculation, perhaps by dividing the tree into sublineages and then calculating the consensuses for each sublineage [18]. However we anticipate that a less heuristic and less discrete approach, perhaps borrowing the concept of a decay length τ from lbi, would provide better results. This would also help with cases where widely separated branches result in a spurious consensus nowhere near any real sequence.

The assumption that all sites are equivalent is particularly inaccurate for BCR sequences: not only are the complementarity determining regions (CDRs) more important for binding than framework regions (FWKs), but activation-induced deaminase (AID) activity results in strongly context-dependent mutation patterns. This information could be incorporated into aa-cdist when calculating the hamming distance by assigning different weights to different regions. For instance, hypothetically, perhaps a one amino acid difference in the CDR should count for twice (or half) as much as a difference in FWK. Similarly, perhaps mutations at highly mutable positions should count less toward the distance than those at less mutable positions.

Even if we could combine aa-cdist and lbi, there is potentially significant information not included in either of them. We thus invested a large effort in developing a machine learning approach incorporating many different tree and sequence variables (Table 2). We found the best performance with a decision tree regression (abbreviated dtr), but were unable to significantly and consistently improve on the performance of aa-cdist alone. We believe this failure stems from two factors. First, aa-cdist is much better than any other variable in most regions of parameter space, and it is possible that there simply isn't much information hiding among the other variables. Second, because the relative performance of different metrics varies dramatically between different parts of parameter space, the dtr has to make a very accurate determination of its location in this space before deciding how to use the input variables. But in making this determination

it is limited to inferred variables which provide only a tenuous link to true parameter space. To take one example, the relative usefulness of lbi and n-shm completely reverses between low and high selection strength (blue and orange in bottom right of S3 Figure). While the input variable Fay/Wu H does predict selection strength [19], it is far too noisy to give the dtr an accurate idea where along the x axis it is for a given family. We thus find that the dtr generally recapitulates the performance of the best single metric at each point in parameter space, but is rarely much better, and sometimes a bit worse. Since the best single metric is almost always aa-cdist, this means that aa-cdist is a better choice overall.

Finally, while it is clear that the observed sequence that falls closest to the family consensus is likely the best, we have not tested whether the actual consensus sequence (when it is not observed) would be even better. In our simulations, at long time scales the family consensus is an extremely accurate predictor of the hypothetical optimal antibody toward which selection is pushing (called the target sequence). However, we do not know to what extent this concept of an achievable "optimal" antibody is applicable to real data, although it has some support [20]. Thus while we are confident that results based on this simulation are in general broadly applicable to real data samples, we feel that on this specific point it is less clear that our simulation approach is applicable to all cases. Furthermore, observing a sequence is an excellent indicator that it actually produces a functional and useful antibody. Thus even if the consensus of the observed sequences were perfectly accurate (i.e. the same as the full-family consensus, which is not a good assumption for very small families, see S7 Figure), we have no good way of measuring how frequently it codes for a useful, functional antibody.

In order to improve lbr's prediction of Δ-affinity we would likely see significant improvement by incorporating amino acid information. One way to do this would be to only count nonsynonymous mutations in the branch length integrals. However an approach drawing more explicitly on the analogy to absolute affinity prediction, such as using the change (from parent node) in hamming distance to a local amino acid consensus sequence, would likely yield better results.

An additional concern with lbr is that because it is sensitive to the detailed long range ancestral lineage structure, it requires more accurate phylogenetic inference than lbi (S8 Figure). While lbi

performs well even with the very heuristic (but also very fast) trees that partis makes by default, lbr would benefit from the more sophisticated inference provided by external packages such as RAxML [21]. Another reason to infer trees with a program such as RAxML is that unlike partis, it infers ancestral sequences, which can then be included in the selection metric calculation as if they were observed sequences. This can be very important, for instance, in studies of antibody evolution [22].

Previous work. Because the metrics we have presented do not provide information on specificity, we now review prior work on ways to enrich for antibodies that are specific to a particular pathogen or epitope. These use information from sources beyond the NGS data, and will provide a key component of most practical workflows involving our methods. As described above, cell sorting and baiting approaches provide a very direct way to identify antigen-specific antibodies, and thus when combined with partis seed sequence clustering, also antigen-specific families in the NGS data [22]. One less-direct way to enrich for a particular specificity is to apply a large immune stimulus, such as vaccination. A frequent approach is to limit the analysis to plasma cells from families that expand around 7 days after vaccination [23–29]. In model organisms, it is also possible to cut out tissues where antigen-specific B cells are likely to concentrate (such as lymph nodes or Peyer's patches). With longitudinal sampling, a family's persistence over time can be a strong indicator of specificity in the presence of either chronic infection [30] or the application of multiple vaccinations [26]. With several subjects that have been exposed to the same antigen, we can select shared lineages either using simple sequence similarity [23–25,31] or a Bayesian mixture model incorporating also clonal abundance [32]. With an outside source of antigen-specific sequences (e.g. from cell sorting or public databases), we can choose sequences that are similar [27]. Specificity prediction without any of these extra information sources, i.e. from sequence data alone, is much more difficult. It would involve de novo structure and binding prediction, which are not currently practical, although much recent work focuses on these problems [33–37].

Many papers have focused on finding families that have been subject to strong selection, which when applied to the BCR repertoire is then assumed to correlate to families with higher affinity.

However, we emphasize that the affinity variation within families is much too great for the selection of entire families to be useful in a practical method (i.e. having picked a family with high mean affinity, we still need to pick a sequence from that family). This is why when we choose sequences from the entire repertoire we do not use an intermediate step that chooses entire families. Probably the most common approach to picking highly selected families has been to measure the ratio of nonsynonymous to synonymous mutations. Observed nucleotide mutations are separated into synonymous (S) and nonsynonymous (NS), the ratio NS/S is calculated, and values much larger (smaller) than one indicate positive (negative) selection. A number of corrections [38–41] and optimizations [42] have also been developed to reduce its dependence on the baseline mutation model.

In perhaps the earliest B cell specific effort to pick families [43], the authors calculate a fairly comprehensive variety of simple tree shape metrics on a handful of small trees (e.g. number of nodes, root-tip distance, outgoing degree, trunk length). While some correlations were noted, it was largely a descriptive exercise without strong quantitative conclusions. Work several years later [44] found correlations between some of the same metrics and GC reaction parameters in a mathematical model. However, a later paper [40] noted that neither [43] nor [44] tested under realistic conditions: both assumed 100% sampling depth, i.e. included every cell from the GC reaction history in the final tree. After accounting for realistic sampling, this later paper found that the simple tree metrics lost their predictive power. They then devised a new metric combining a small amount of tree information with NS/S by ignoring terminal mutations (which as described above are much less likely to lead to high affinity antibodies). This is analogous to a combination of lbi, which uses tree structure to measure selection, with aa-cdist, which reduces the influence of tip mutations. A more recent paper [19] found that NS/S is higher in CDR than FWK regions, and calculated the likelihood-based fitness from [5] for several trees, but found no significant relationships between changes in this fitness and NS/S or CDR vs FWK.

A more complex statistic known as Fay/Wu H [45] has also been used in the context of BCR repertoires. It quantifies any excess at high values of the site frequency spectrum, and can be

thought of as measuring the amount of shared mutation in the family, or equivalently the prevalence of selective sweeps. It was used in [19] to determine that families that expand rapidly in response to vaccination are generally positively selected. We independently verified that it indeed identifies highly-selected families (results not shown), and thus included it in the dtr.

A metric called the log offspring number ratio, which provided a starting point for lbr, was introduced in [6]. This metric looks in the tree for pairs of sibling branches where one branch has a mutation and the other does not. It is then calculated as the log of the ratio between the number of offspring in the mutated vs non-mutated branches. The distribution of this value is then calculated separately for NS and S mutations, and a rightward (leftward) shift in the NS distribution indicates positive (negative) selection. It unfortunately is rendered less useful by several issues. First, it counts offspring all the way down the tree, so that ancestors get credit for fitness improvements in all of their offspring, so it cannot be used to find the location of important mutations. It also ignores sibling pairs in which either both edges have mutations, or either branch has multiple mutations, which together can amount to throwing out a large fraction of the tree. Finally, it can only detect in-progress (i.e. not-yet-fixed) incremental selection, and the NS and S mutation rates must be of the same order of magnitude.

The remainder of previous work has focused on choosing single sequences from within a family, and separates into experimental and theoretical studies. Experimental papers typically first choose a family based on the methods above for specificity to the pathogen of interest, then exclude sequences with "bad" features (e.g. highly hydrophobic, or with free cysteines or weird indels), and then rank the remaining sequences by measures such as n-shm, V gene usage, and CDR3 length. On the theoretical side, lbi has undoubtedly enjoyed the most practical use. Introduced in [5], it was originally a quick heuristic replacement for a more complex likelihood-based metric. Its ability to predict fitness in real influenza data has been shown in [46], and it features prominently on the nextstrain [47] web site `https://nextstrain.org/flu/seasonal/h3n2/ha/2y?c=lbi`.

In the only work we're aware of that evaluated aa-cdist, the authors adapted an approach from structural studies called maximum entropy [11]. This models the multiple sequence alignment of a family as a multivariate gaussian, and then takes probability (i.e. nearness to a gaussian's peak) as

a correlate of affinity. This maximum entropy metric performs well in their tests, however its generality is unknown, since it was trained and tested using only one relatively small real data sample. Although they avoid statistical overtraining by training and testing on disjoint parts of the sample, a single sample can only tell us about performance at the one point in parameter space at which it resides. Furthermore, since training consists of optimizing to that particular sample, other regions of parameter space are by construction very likely to have worse performance. Finally, the resulting software does not seem to be publicly available, although the underlying gaussian modeling framework is available at `https://github.com/carlobaldassi/GaussDCA.jl`. Nevertheless, as a cross check in the supplemental information when predicting binding vs non-binding antibodies, they use area under curve (AUC) to compare their method (0.97) to aa-cdist (0.86). This indicates the surprising usefulness of aa-cdist, especially since aa-cdist is certainly not subject to the same potential for overtraining.

A very exciting (and very recent) development is a microfluidics-based approach to experimentally evaluating antibody specificity in high throughput [48]. Of particular relevance to our results, they find no correlation between n-shm and measured affinity.

METHODS

Simulation framework. The simulation of each clonal family in this paper begins with a naive sequence generated by the partis simulation command [8]. Since here we focus on affinity maturation rather than VDJ rearrangement, we refer to that paper for all details on its implementation and validation. This naive sequence is then passed to the bcr-phylo package [15] for GC reaction simulation. For this paper we have extended the original software by adding a number of new parameters to allow for more comprehensive variation, as well as optimizing for speed. The full simulation of a family combining partis and bcr-phylo can be run with the following script: `https://git.io/JvFcW`.

The bcr-phylo simulation begins by generating a number of "target" nucleotide sequences from the naive nucleotide sequence. These targets represent hypothetical optimal antibodies toward

which evolution will be directed, and are chosen at fixed distance from the naive sequence (default 15 amino acid hamming distance). The representation of the GC reaction proceeds generation by generation, beginning with the naive sequence's single cell. In each generation, a number of offspring is chosen for each cell from a Poisson distribution with parameter λ determined by that cell's affinity; if a cell has zero offspring in a generation its lineage ends. The correspondence between nucleotide sequence and affinity is by default determined by amino acid hamming distance to target sequence. For a detailed description refer to the supplement of [15], but the net result is a monotonic increase in mean number of offspring as a cell's sequence draws closer to a target sequence. We also tested a model with a much less discrete distribution of possible affinity values, where the distance for each amino acid pair is rescaled by their BLOSUM similarity (S2 Figure, bottom left). A carrying capacity is implemented within the mapping from affinity to Poisson parameter, with the result that the population initially increases rapidly until plateauing at the carrying capacity. In order to introduce a selection strength parameter (bottom right of S2 Figure and S3 Figure), instead of determining the Poisson parameter directly from affinity, we smear it out (rescale it) by drawing from a normal distribution. The normal distribution's mean and variance are calculated such that for a selection strength of 0, affinity has no effect on offspring number, while for selection strength 1 the Poisson parameter is determined directly by affinity (but since the number of offspring is still drawn from that Poisson, there is still significant stochasticity). Specifically, the parameters for the ith cell's normal distribution are given by $\mu_i = \bar{\lambda} + s(\lambda_i - \bar{\lambda})$ and $\sigma_i = (1-s)\sigma_\lambda$, where s is the selection strength, λ_i is the ith cell's unrescaled (affinity-determined) Poisson parameter, and $\bar{\lambda}$ is the mean and σ_λ the standard deviation of unrescaled Poisson parameters of all cells. Note that σ_i is somewhat arbitrary, but this choice ensures that the variance in rescaled λ over cells is comparable for all values of s. The simulation is terminated after some number of generations, referred to as the observation time.

When each child cell is generated, its nucleotide sequence has some probability to suffer a point mutation. This probability is set based on experimental values, but averages roughly one point

mutation per generation. Synonymous mutations have no effect on affinity, while a nonsynonymous mutation that decreases the distance (whether plain or BLOSUM-scaled hamming) to a target sequence increases affinity, and thus mean number of offspring (Poisson parameter). While context-dependent mutation is implemented in bcr-phylo (using the S5F model [49]), because this feature increases run times by vastly too much to allow the generation of the large samples needed for realistic validation, all samples shown here have this option turned off. In the limited sample sizes that we have run with context dependence turned on (results not shown), neither aa-cdist nor lbi consistently performed either better or worse. The main pattern was that n-shm performed significantly worse with context dependence turned on.

Given the complete simulated tree of cells, we have to decide which we will sample. In order to simplify the many ways of choosing N cells from M generations, we focus on two cases that cover two alternative models for GC cell export: either sampling a fraction of the desired cells every few generations (top rows of S2 Figure and S3 Figure), or sampling all cells at the end (all others). In addition, in order to mimic typical phylogenetic programs that infer ancestral sequences, we then also recursively sample all MRCA sequences starting with the set of sampled cells (although we have also run extensive validation without this option (results not shown), and the only significant effect is due to the change in the fraction of leaf sequences). Different models of GC export also stipulate different levels of bias toward exporting higher affinity cells. We attempt to cover the various possibilities with three options (S4 Figure): uniform random (default), sampling with probability proportional to affinity, and sampling the cells in order of perfectly decreasing affinity.

An unavoidable feature of our approach is that we must simulate a vastly larger number of sequences than we want to sample. Since every cell in the evolutionary history of the family must be simulated, large carrying capacities and observation times dramatically increase the required time and memory. For example, each point in Fig 1 uses only 4.5×10^4 final sampled sequences, but requires actually simulating 3×10^8 sequences. And the biggest dtr training sample, with 2.5×10^6 final sampled sequences, required simulating roughly 7×10^{10} sequences (Table 3).

When we simulate families with high SHM frequencies, for simplicity we treat them as a single long-running GC reaction rather than multiple sequential reactions. Antibodies with 20-30% SHM are generally assumed to have undergone many cycles of GC completion and reentry over a number of years (although antibodies with 20% SHM have been observed after only 28 days [50]). Treating this instead as a single long-running GC reaction, however, is probably a good approximation, because GC completion followed by reentry of memory B cells is similar to a strong selective sweep [51]. This means that GC completion and reentry is largely equivalent to increasing either the observation time or selection strength.

While we have great confidence that our simulation framework effectively recreates the important features and variation of BCR evolution, we would prefer to also validate on a completely independent package. Unfortunately, while there are many packages that come close to simulating what we need [52–55], to our knowledge they all lack necessary features. Most frequently, they do not simulate complete nucleotide sequences, and instead draw a cell's affinity either directly from some distribution, or based on a few key residues.

Evaluation framework. While we described the basics of our performance evaluation above, there are a lot of steps between a metric vs affinity scatter plot and the quantiles in Fig 1. The question we're trying to answer is, if we take the best few sequences according to our metric, how high can we expect those sequences to rank in affinity? In a metric vs affinity scatter plot (left side of Fig 3, or top row of Fig 4) this means taking the highest few points by y, and seeing where they rank in affinity (x). We calculate these rankings as the quantile in affinity, averaged over the chosen sequences. For instance the sequences in the top 4% of affinity have quantiles from 96%-100%, and thus mean quantile 98%. The top 4% of sequences according to some metric, on the other hand, might have affinity quantiles spread between 85%-100%, which could give a mean of 92%. This example both neglects ties and assumes infinite sample size; in practice both are important, which results in jumps and horizontal lines as in Fig 3. This is one reason why it's important to compare to the mean quantile of a hypothetical perfect metric, rather than to a constant value of 100. We plot this resulting mean affinity quantile for thresholds from the 50th to 100th quantile (i.e. choosing from half to none of the sequences, right side of Fig 3, or middle row of Fig 4).

In order to compare performance over many different parameter choices, we need to summarize this plot with one number. We do this by defining the deviation from perfect as the difference between the metric's performance and that of a hypothetical perfect method (distance between blue and green lines). We then average this quantity from the 75th to 100th quantiles. This average, reported as a mean $\pm$ standard error over many statistically independent samples, is what appears in Fig 1.

When predicting Δ-affinity, we cannot simply report the mean Δ-affinity quantile of the chosen sequences because we want to account for being close to, but not exactly on, the correct branch. We instead imagine moving upward on the tree from the node of interest until reaching a branch containing an affinity-increasing mutation. We report the number of steps (i.e. ancestors) that were necessary, so if we're exactly right this number is 0. Searching only upward reflects the fact that a mutation can only affect the fitness of nodes below it, and thus a high lbr value at a node immediately above an important mutation is likely due to random chance rather than a signal of selection. Nodes with high lbr that are several steps below such a mutation, on the other hand, simply reflect the fact that increased fitness typically takes several generations to manifest itself as an increase in observed offspring. In other words searching downward would improve the apparent performance of our metric, but only by counting as successes cases that were successfully predicted only through random chance. Another reason we do not also search in the downward direction is that in a practical sense it is much more useful to know that the important mutation is above a node than below it. We could imagine in the lab testing one or a few branches above a node, but because of the bifurcating nature of trees there would be far too many potential branches below (not to mention adding the ambiguity of potentially going up and then down, i.e. how to count cousins). One potential issue with this step-counting approach is that it gives equal credit for being off by long and short branches. We thus also performed extensive validation using the total branch length traversed, rather than number of steps (results not shown). The performance was generally similar, and is also probably less relevant since during inference we don't control how long the branches are. For instance any metric would appear to do worse on sparsely-sampled trees with long branches.

Distance to family consensus sequence (aa-cdist and nuc-cdist). One nice feature of consensus distance metrics is that, unlike lbi and lbr, there is no inference inaccuracy: they are a direct result from observed sequences. However, these observed sequences will in some cases not accurately represent the entire family. In order to quantify this inaccuracy, we calculated the full-family consensus sequence, and then compared it to consensus sequences calculated with smaller subsets of the family (S7 Figure). While the nucleotide consensus can be quite inaccurate (top), we only calculate it in order to inform comparison between aa-cdist and lbi. The amino acid consensus, on the other hand, is quite accurate, reaching an error of one position (out of around 130) only for sampled families smaller than 10 sequences and very early times (i.e. when almost no selection has yet occurred, S7 Figure bottom left).

local branching index (lbi). The authors of [5] introduce lbi as an approximate metric to supplement their more complex likelihood-based measure. However, they find that the two perform very similarly, so the full likelihood calculation is probably more useful for building intuition than practical use. The lbi score for each node in the tree is computed as the total of all nearby branch lengths, weighted by an exponential function of the distance away from that location with scale τ (S1 Figure).

τ optimization. The decay length τ determines the size of the local area that impacts each node's lbi. The authors of [5] suggest using 0.0625 times the average pairwise distance between sequences (elsewhere they use the coalescence time $T_C/\sqrt{\log N}$ and $T_C/15$, but since we're not using coalescent models this is less useful here). While these estimates are based on a thorough optimization, they result in a τ that depends on which sequences are sampled, which precludes comparisons between families, as well as between samples in different papers. We thus perform an optimization from a somewhat different perspective, although we end up with a comparable value.

We begin by noting that the sequence length has a profound effect on the distances over which trees branch, and for BCR sequences it is constant across families and samples. The minimum possible branch length, corresponding to one point mutation, is equal to the inverse sequence length $1/\ell_{\text{seq}}$. Thus from first principles we expected this to be a reasonable guess for τ. However, we also

want a value that gives optimal performance, so we measured performance vs τ in a number of parameter scans similar to Fig 1. The strongest dependence was vs number of sampled sequences as a fraction of carrying capacity (S9 Figure, top row). While the most important message from this plot is probably that lbi performance does not depend very strongly on τ, there is a pronounced peak in performance at $1/\ell_{\text{seq}} \simeq 1/400 = 0.0025$, especially when choosing within families (top right), where lbi is most important. We get a similar value if we (roughly) calculate the recommendation from [5] for 10% SHM: $0.0625 \times 0.10 = 0.00625$. Thus we recommend using $\tau = 1/\ell_{\text{seq}}$ for general use, which for BCR sequences is about 0.0025.

As mentioned above, because sequence multiplicity is experimentally difficult to measure accurately, we do not currently use it, since any spurious multiplicities could easily overwhelm the information from unique sequences, which is likely to be more accurate. However, in cases where they are reliable, multiplicities would be an extremely useful source of additional information. If a cell has three sampled offspring, for instance, that is a strong indication of fitness regardless of whether the offspring all have identical BCR sequences. We have thus implemented a version of lbi that incorporates multiplicity information for each node, although it is turned off by default. It works by adding additional dummy branches above each node that has multiplicity greater than 1. For instance a node with multiplicity 3 will have 3 branches connecting it to its parent, rather than 1. This represents the case where the three observed sequences all "split off" at the top of the branch to its parent. In reality a split point somewhere in the middle of the branch would likely be more accurate, but we think that the current approach is a reasonable first approximation.

lbi normalization. In general, both the absolute magnitude of lbi and its value relative to other sequences are meaningful. For influenza virus evolution, however, which is the case of most interest to [5], only the relative value is useful, since there is only one global influenza "clonal family", and we know that at least some of these viruses will survive in the future (i.e. we definitely want to "choose" some of them). They thus normalize lbi to the maximum value within the tree [47], discarding all information on its magnitude. Our case, however, is quite different: we are trying to determine how good an antibody is likely to be, so we care very much about the magnitude.

Magnitude tells us whether this is a really branchy bit of the tree, not just whether it's branchier than the rest of the tree.

We go further than using the magnitude of lbi, however, and normalize this value relative to two universal and intuitively meaningful minimum and maximum values. One reason is that raw lbi can only be compared to other values that were calculated with the same τ: a comparison to calculations that used a different τ, or simply don't report τ, is meaningless.

We thus look for some theoretically meaningful minimum and maximum values of lbi, and set them to 0 and 1. Note that these don't have to be the actual smallest and largest possible values in order to be useful: for instance the Centigrade temperature scale definitions of 0 and 100 are entirely analogous.

To find a maximum value, we construct a synthetic tree representing the "very branchy" case, then search among its nodes for the maximum value. We generate such a tree by bifurcating after every point mutation (i.e. after a branch length of $1/\ell_{\mathrm{seq}}$), then find the node in the tree with maximum lbi. In order to avoid dependence on the depth of this tree, we start with small N generations, and increase depth until (hopefully) reaching an asymptote. If it diverges, i.e. does not reach an asymptote, the exercise is not meaningful (indeed when we N-furcate for $N > 2$ it never converges, results not shown). Calculating this numerically, we find that the maximum value converges to an asymptote for τ less than around $1/\ell_{\mathrm{seq}}$, but diverges for larger values (S10 Figure). Luckily τ optimization resulted in a comparable recommendation. While this bifurcating tree is of course only one of many possible choices, its purpose is to serve as some reasonable benchmark: when looking at a node with lbi of 1, it's very useful to know that its local area is roughly as branchy as a tree that bifurcates every $1/\ell_{\mathrm{seq}}$.

Finding a minimum value is easier: we construct the (deep) caterpillar tree and find the minimum lbi among its nodes. This minimum is simply τ, a result which can also be obtained by performing a trivial analytic integration. We thus normalize lbi such that this minimum value is 0, and the maximum value (previous paragraph) is 1. Note that leaves in shallow trees can thus have values less than 0, and dense trees can result in values greater than 1.

Dummy branches. An inherent bias in the lbi calculation is that nodes near the edge of the tree (i.e. near root or leaves) are systematically biased low. While there is no way to address the underlying reason for this – by definition we don't know what happens before root or after leaves – we can test some corrections. Calculating lbi on a tree that ends abruptly at leaves and root amounts to using the (implausible) prior that root sprang forth from nothing, and that every leaf died immediately upon sampling (or rather, would have died even without being sampled). We can instead use the much more plausible prior that the status quo at these times continued to $\pm$infinite time by adding long "dummy" branches above root and below each leaf. Unfortunately this does not turn out to improve performance (results not shown). We suspect that this is because we actually know another key piece of information that is not encoded in lbi: most novel mutations are deleterious. In other words the fact that lbi is biased low for leaves is compensated for by the fact that for external reasons we know that leaves are typically low fitness.

Relative fitness. We also note that, by design, lbi measures a cell's fitness relative to cells alive at the same time (i.e. cells against which it was actually competing). Thus in any situations that involve sampling sequences from many different time points (e.g. sampling lots of ancestral sequences) performance suffers in an absolute sense. This can be ameliorated in practice simply by paying attention to the location of nodes within a tree, and keeping in mind that affinity likely increases for much of the length of the tree.

local branching ratio (lbr). We calculate lbr with the same integrals as lbi, so much of that discussion applies here as well. The difference is that instead of adding up the branch lengths in all directions, we instead divide those below the node by those above and beside it (S1 Figure). This results in a very sharp distinction between the effects of mutations above the parent vs those below.

As for lbi, we performed an optimization of the decay length τ, but with quite different results (S9 Figure). First, the variation in performance is much more significant. Second, there is no peak in performance: instead it improves monotonically as τ increases to very large values. This makes sense since larger τ means that more of the tree is visible, which adds information. For lbi, however,

there is a countervailing effect: resolution as to which node we're focusing on is also determined by τ, which disfavors large τ. But the resolution of lbr is determined by the sharp distinction between numerator and denominator, so τ can increase without penalty. Another way of explaining why lbr improves with larger τ is that if a fitness-increasing mutation has just occurred, the chances of having many offspring in the first generation is likely not very close to 1 even if it is strongly advantageous (for example, let's say it's $1/3$). But as we proceed down the tree, the chance of any of the next few generations having many offspring is much closer to one, for instance after five generations it would be $1 - (1/3)^5 = 0.99$. We thus (somewhat arbitrarily) choose to use τ of $20 \times 1/\ell_{\text{seq}}$for lbr.

While the same normalization procedure could be followed as for lbi, there is no point, because lbr is an inherently unitless ratio.

In the case of lbi above, we concluded that adding "dummy" branches to correct for biases did not make sense. For lbr, however, we have an additional consideration: lbr begins to diverge for nodes near the root, since there is no branch length above the root (in the denominator). Because this is highly undesirable, we add a very long dummy branch above the root node during lbr calculation.

Tree inference methods. When evaluating the simulation performance of metrics that depend on trees (lbi, lbr, and Δ-lbi), we use only the true simulated tree. This is because we want to separate the ability of the metrics to predict affinity or Δ-affinity from the performance of a particular phylogenetic inference method. We then separately evaluate the robustness of the lbi and lbr calculations to use of the fast but relatively inaccurate tree inference that partis uses by default (S8 Figure). We find in this test that tree inference matters very little for lbi, but is important for lbr. This makes sense, since lbi depends on shorter range comparisons between similar sequences that are likely easier to capture with a heuristic tree method, while lbr is sensitive to the longer range details of ancestral inference. It would be useful to also make this comparison for other phylogenetic inference programs and over the huge variety of tree characteristics encoded by our simulation parameters. As these programs get more accurate, however, they also get much slower, so this will require a substantial time investment.

The cases we can envision where better phylogenetic inference is important will usually involve only a few families, and trees can thus be inferred separately using programs such as linearham [56] or RAxML [21]. These trees would then be passed to partis for selection metric calculation using the `--treefname` option of the `get-selection-metrics` action (see `https://git.io/JfeGk`). Most commonly this would be necessary after an antibody of interest has already been chosen. In such a case we would want to infer ancestral sequences in that antibody's family (which partis cannot currently do), and then include these ancestors in the selection metric calculation. Such a workflow was followed in [22] (albeit without the selection metric calculation). This used the linearham package for accurate Bayesian inference of both trees and naive sequences (see `https://github.com/matsengrp/linearham`), and the Olmsted package for visualization (see `https://github.com/matsengrp/olmsted`), which are both highly recommended.

The fast but heuristic tree inference method that is currently included in partis combines the history of its clustering algorithm with the FastTree inference program. Because this clustering proceeds via hierarchical agglomeration [1], the history of the clustering process itself constitutes a tree. While this tree is based only on sequence similarity measures (either inferred naive hamming distance or the partis VDJ rearrangement likelihood) rather than a model of sequence evolution, in many cases the result will be quite similar. The biggest inaccuracy in this approach is that the agglomeration frequently merges many clusters together in a single step, resulting in a large multifurcation in the tree. Typically the largest of these happens in the initial clustering step, when it merges together input sequences with very similar inferred naive sequences (for details and thresholds see [1]). We thus refine the clustering-based tree by removing any subtree whose root has more than two offspring, and replace it with a subtree inferred by FastTree [57]. Because FastTree forces any observed ancestral sequences to be very short leaves hanging off of the corresponding internal node, we then also collapse any such leaf with length less than $0.5 \times 1/\ell_{\text{seq}}$. Note that this method does not allow for ancestral sequence inference, which for detailed studies of single families will be quite important.

The speed of this calculation is entirely dependent on how much of the tree needs to be inferred via FastTree, since the non-FastTree parts come along for free from the already-completed clustering. But on typical BCR NGS samples, the only appreciable time is taken for the largest few families, and a family of for instance several thousand sequences can be expected to take a few minutes.

Decision tree regression (dtr). The fact that we have several metrics that perform adequately suggests very strongly that we combine them using some form of machine learning. We focused on decision tree regressors, and found that gradient boosting generally performed better than other methods. We tried many combinations of input variables, but found no benefit to reducing their number from the full set in Table 2. We trained different dtr versions both for choosing among all families and for choosing within each family, as well as for predicting both affinity and Δ-affinity.

Because we need the dtr to perform well for all possible combinations of parameters, we must construct a training sample consisting of families exhibiting a huge variety of characteristics, which requires an extremely large number of families. The types of samples used for the parameter scans, such as Fig 1, are useless for this, since they hold all but one or two parameters constant. We thus made several highly variable samples (Table 3), each generated by first choosing a distribution for each parameter. To simulate a family, a value for each parameter is then drawn from that parameter's distribution. We generated between two and five independent samples for each set of parameter distributions, i.e. the dtr was trained on one sample of the indicated size, while there was at least one other sample for testing that had identical underlying parameters, but was generated with a different random seed. We also tested on all other combinations of parameter distributions, as well as the slice/scan samples from e.g. Fig 1. Finally, we also tested a comprehensive variety of the parameters describing dtr training (e.g. number of estimators, decision tree depth, and pruning details). Note that we test on training samples only in order to evaluate statistical overtraining, i.e. to ensure that performance is comparable on samples that differ from the training sample only by random seed; we never report or talk about the "performance" on a training sample.

Here we summarize our conclusions; full results can be found at https://doi.org/10.5281/zenodo.3728068. We managed to create a dtr that in most regions of parameter space roughly recapitulates the performance of the best single metric (usually aa-cdist, but in some places lbi), but improves upon it only slightly (perhaps a few percent). Because of the significant complication introduced by the use of any machine learning method, and their lack of interpretability, we think they are only worthwhile in cases where they improve performance by much more than this. This is of course influenced by the fact that we have one metric performing best almost everywhere; if different regions of parameter space required different single metrics, the dtr would be much more attractive.

It is difficult, if not impossible, to definitively attribute the dtr's failure to improve performance as much as we expected, since we cannot directly measure whether these expectations were reasonable. However, we think it is highly likely that there is significant additional information in the more poorly-performing variables, and we are also quite confident that a number of features of our use case make it particularly challenging for a machine learning approach. The metrics lbi and aa-cdist are far from perfectly correlated (Fig 4, at bottom), and while this discordance is of course not entirely due to useful independent information, lbi clearly contains tree and locality information that is not included in aa-cdist. One challenging feature of our case is that the inferred tree quantities that serve as input variables (e.g. n-shm, Fay/Wu H) are very noisy predictors of the true tree parameters are (e.g. observation time, selection strength, carrying capacity). To take one example, the relative performance of lbi and n-shm completely reverses between low and high selection strength (blue and orange in bottom right of S3 Figure), but Fay/Wu H is far too poor an estimator of selection strength to give the dtr an accurate idea where along the x axis it is for a given family. It is likely that more sophisticated tree inference would improve performance by allowing the use of more fundamental tree variables as input. It is also of course entirely possible that a different machine learning approach would be able to improve on our efforts; however given the exemplary performance of aa-cdist alone, we do not feel that this is a high priority.

Table 2. **Decision tree regression (dtr) input variables**. The among-families dtr uses both per-sequence (top) and per-family (bottom) metrics; while the within-family dtr uses only the former.

per-sequence metrics	description
aa-cdist	see Table 1
nuc-cdist	see Table 1
lbi	see Table 1
lbr	see Table 1
n-shm	see Table 1
n-shm-aa	amino acid distance to family naive sequence
edge dist.	min. distance in tree to either root or tip

per-family metrics	description
Fay/Wu H	measures selection via excess in site frequency spectrum [45]
nuc cons seq SHM	nucleotide distance from family consensus sequence to naive sequence
aa cons seq SHM	amino acid distance from family consensus sequence to naive sequence
mean n-shm	mean number of nucleotide mutations among sequences in the family
max lbi	maximum lbi value in the family
max lbr	maximum lbr value in the family

Table 3. **Parameters used for dtr training samples.** Versions v0-v2 sampled parameters for each family from uniform distributions of the indicated mean and half-width, while v3 sampled with equal probability from the indicated discrete values.

label	N families per sample	N samples	carry cap.	obs times	N sampled	selection strength
v0	1000	5	1500 ± 1000	150 ± 75	150 ± 100	0.75 ± 0.25
v1	50000	5	1500 ± 1000	150 ± 75	30 ± 7.5	0.75 ± 0.25
v2	300000	2	1500 ± 1000	150 ± 75	20 ± 7.5	0.75 ± 0.25
v3	50000	2	250,500,900,1000,1100,1500,5000	75,100,150,200,1000	15,30,75,150,500	0.5,0.9,0.95,1.0

Acknowledgements

This research was supported by National Institutes of Health grants R01 GM113246, R01 AI146028, R01 AI138709, and R01 AI120961. The research of Frederick Matsen was supported in part by a Faculty Scholar grant from the Howard Hughes Medical Institute and the Simons Foundation. The research of Duncan Ralph was supported by a 2018-2019 University of Washington/Fred Hutch Center for AIDS Research Young Investigator award (P30 AI027757).

We would like to thank Laura Doepker, Mackenzie Shipley, and Julie Overbaugh for collaborations that motivated this work, as well as Kristian Davidsen and Will DeWitt for helpful discussions. We are indebted to Peter Ralph for invaluable help in editing the manuscript.

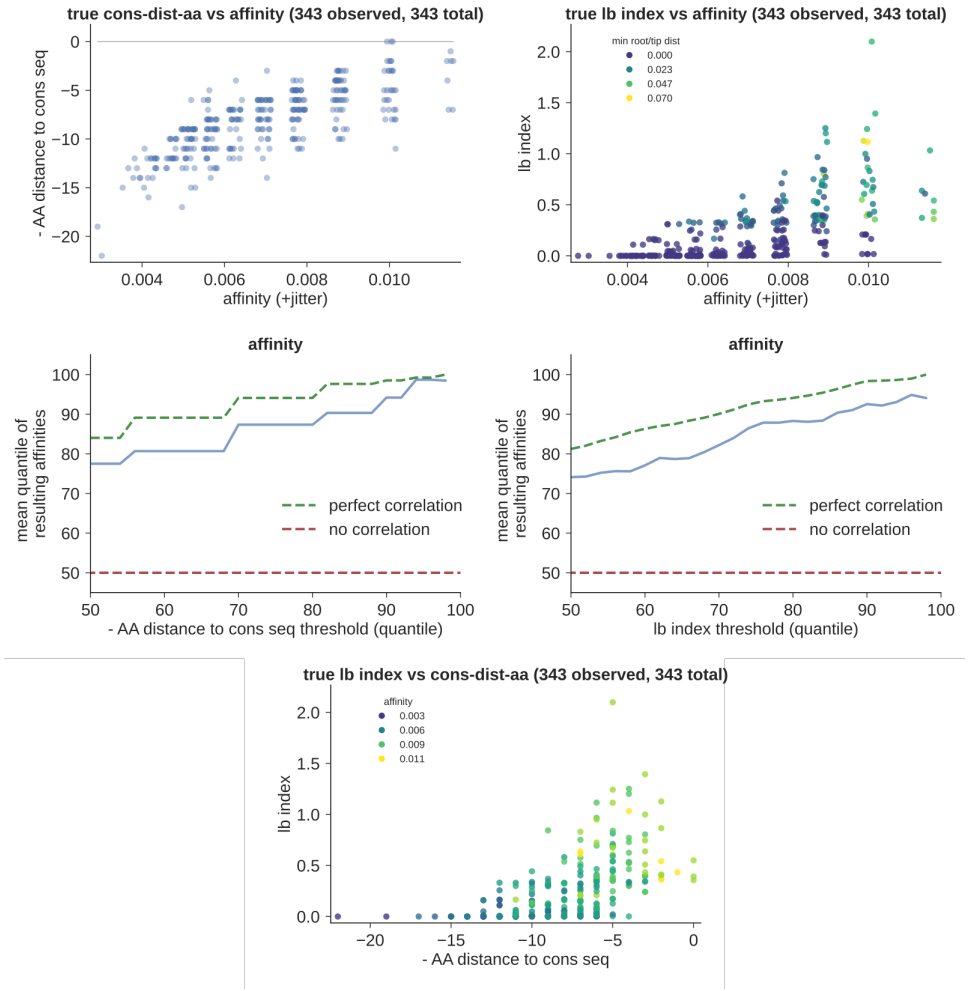


Fig 4. **Simulation performance of aa-cdist (left) and lbi (right) on a single, representative family.** First showing scatter plots of each metric vs affinity (top), and then the quantile performance plots (middle, see text and Fig 1 for explanation). The lbi scatter plot is colored by the distance to the "edge" of the tree, i.e. minimum distance to either tip or root: lbi is less accurate (and biased low) for nodes near the edge of the tree (darker color). At bottom, we show the correlation between aa-cdist and lbi, colored by affinity.

References

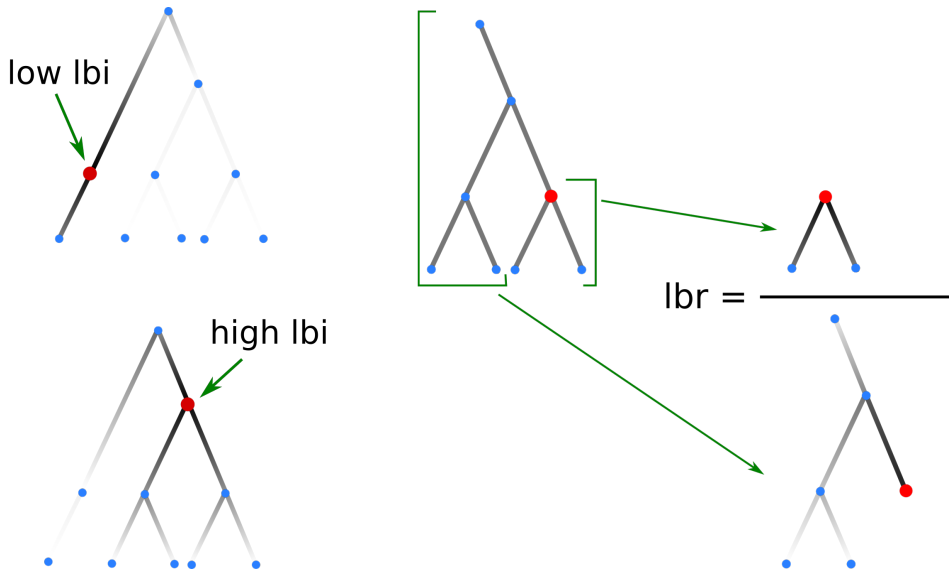
1. Ralph DK, Matsen IV FA. Likelihood-Based Inference of B Cell Clonal Families. PLOS Comput Biol. 2016 Oct;12(10):e1005086.

2. Kong L, Ju B, Chen Y, He L, Ren L, Liu J, et al. Key gp120 Glycans Pose Roadblocks to the Rapid Development of VRC01-Class Antibodies in an HIV-1-Infected Chinese Donor. Immunity. 2016 Apr;44(4):939–950.
3. Simonich CA, Williams KL, Verkerke HP, Williams JA, Nduati R, Lee KK, et al. HIV-1 Neutralizing Antibodies with Limited Hypermutation from an Infant. Cell. 2016 Jun;166(1):77–87.
4. Galson JD, Kelly DF, Truck J. Identification of Antigen-Specific B-Cell Receptor Sequences from the Total B-Cell Repertoire. Crit Rev Immunol. 2015;35(6):463–478.
5. Neher RA, Russell CA, Shraiman BI. Predicting evolution from the shape of genealogical trees. Elife. 2014 Nov;3.
6. Liberman G, Benichou J, Tsaban L, Glanville J, Louzoun Y. Multi Step Selection in Ig H Chains is Initially Focused on CDR3 and Then on Other CDR Regions. Front Immunol. 2013 Sep;4:274.
7. Barrat-Charlaix P, Neher RA. Limited predictability of amino acid substitutions in seasonal influenza viruses. (forthcoming preprint/personal communication);.
8. Ralph DK, Matsen FA 4th. Consistency of VDJ Rearrangement and Substitution Parameters Enables Accurate B Cell Receptor Sequence Annotation. PLoS Comput Biol. 2016 Jan;12(1):e1004409.
9. Ralph DK, Matsen FA 4th. Per-sample immunoglobulin germline inference from B cell receptor deep sequencing data. PLoS Comput Biol. 2019 Jul;15(7):e1007133.
10. Nivarthi UK, Tu HA, Delacruz MJ, Swanstrom J, Patel B, Durbin AP, et al. Longitudinal analysis of acute and convalescent B cell responses in a human primary dengue serotype 2 infection model. EBioMedicine. 2019 Mar;41:465–478.
11. Asti L, Uguzzoni G, Marcatili P, Pagnani A. Maximum-Entropy Models of Sequenced Immune Repertoires Predict Antigen-Antibody Affinity. PLoS Comput Biol. 2016 Apr;12(4):e1004870.
12. Carlson CS, Emerson RO, Sherwood AM, Desmarais C, Chung MW, Parsons JM, et al. Using synthetic templates to design an unbiased multiplex PCR assay. Nat Commun. 2013;4:2680.
13. DeWitt W, Lindau P, Snyder T, Vignali M, Emerson R, Robins H. Replicate immunosequencing as a robust probe of B cell repertoire diversity. 2014 Oct;.
14. DeWitt WS, Lindau P, Snyder TM, Sherwood AM, Vignali M, Carlson CS, et al. A Public Database of Memory and Naive B-Cell Receptor Sequences. PLoS One. 2016 Aug;11(8):e0160853.
15. Davidsen K, Matsen F. Benchmarking tree and ancestral sequence inference for B cell receptor sequences; 2018.
16. Landais E, Murrell B, Briney B, Murrell S, Rantalainen K, Berndsen ZT, et al. HIV Envelope Glycoform Heterogeneity and Localized Diversity Govern the Initiation and Maturation of a V2 Apex Broadly Neutralizing Antibody Lineage. Immunity. 2017 Nov;47(5):990–1003.e9.
17. Wu X, Zhou T, Zhu J, Zhang B, Georgiev I, Wang C, et al. Focused evolution of HIV-1 neutralizing antibodies revealed by structures and deep sequencing. Science. 2011 Sep;333(6049):1593–1602.
18. Giles BM, Ross TM. A computationally optimized broadly reactive antigen (COBRA) based H5N1 VLP vaccine elicits broadly reactive antibodies in mice and ferrets. Vaccine. 2011 Apr;29(16):3043–3054.

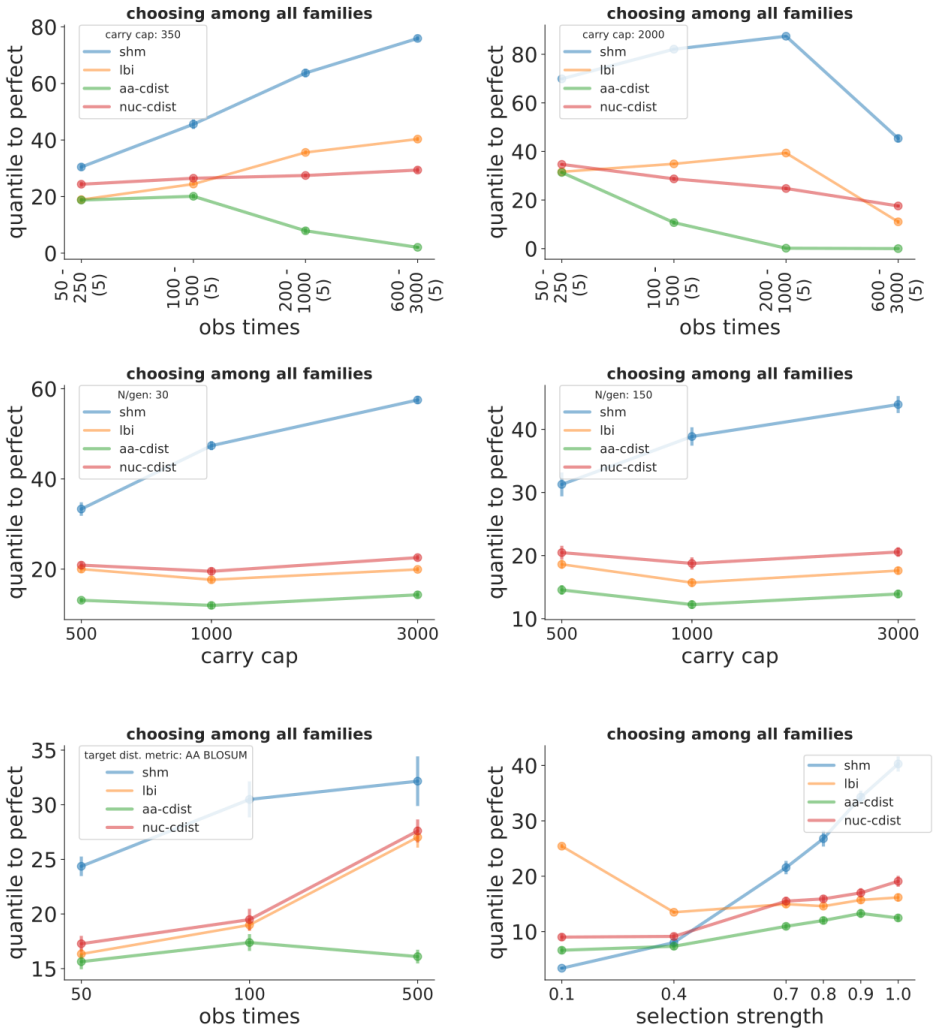
19. Horns F, Vollmers C, Dekker CL, Quake SR. Signatures of selection in the human antibody repertoire: Selective sweeps, competing subclones, and neutral drift. Proc Natl Acad Sci U S A. 2019 Jan;116(4):1261–1266.
20. Desikan R, Antia R, Dixit NM. The weakest link bridging germinal center B cells and follicular dendritic cells limits antibody affinity maturation. 2020 Feb;.
21. Stamatakis A. RAxML version 8: a tool for phylogenetic analysis and post-analysis of large phylogenies. Bioinformatics. 2014 May;30(9):1312–1313.
22. Simonich CA, Doepker L, Ralph D, Williams JA, Dhar A, Yaffe Z, et al. Kappa chain maturation helps drive rapid development of an infant HIV-1 broadly neutralizing antibody lineage. Nat Commun. 2019 May;10(1):2190.
23. Trück J, Ramasamy MN, Galson JD, Rance R, Parkhill J, Lunter G, et al. Identification of antigen-specific B cell receptor sequences using public repertoire analysis. J Immunol. 2015 Jan;194(1):252–261.
24. Galson JD, Trück J, Fowler A, Clutterbuck EA, Münz M, Cerundolo V, et al. Analysis of B Cell Repertoire Dynamics Following Hepatitis B Vaccination in Humans, and Enrichment of Vaccine-specific Antibody Sequences. EBioMedicine. 2015 Dec;2(12):2070–2079.
25. Galson JD, Trück J, Clutterbuck EA, Fowler A, Cerundolo V, Pollard AJ, et al. B-cell repertoire dynamics after sequential hepatitis B vaccination and evidence for cross-reactive B-cell activation. Genome Med. 2016 Jun;8(1):68.
26. Vollmers C, Sit RV, Weinstein JA, Dekker CL, Quake SR. Genetic measurement of memory B-cell recall using antibody repertoire sequencing. Proc Natl Acad Sci U S A. 2013 13 Aug;110(33):13463–13468. Available from: `http://dx.doi.org/10.1073/pnas.1312146110`.
27. Jackson KJL, Liu Y, Roskin KM, Glanville J, Hoh RA, Seo K, et al. Human responses to influenza vaccination show seroconversion signatures and convergent antibody rearrangements. Cell Host Microbe. 2014 Sep;16(1):105–114.
28. Jiang N, He J, Weinstein JA, Penland L, Sasaki S, He XS, et al. Lineage structure of the human antibody repertoire in response to influenza vaccination. Sci Transl Med. 2013 Jun;5(171):171ra19.
29. Tan YC, Blum LK, Kongpachith S, Ju CH, Cai X, Lindstrom TM, et al. High-throughput sequencing of natively paired antibody chains provides evidence for original antigenic sin shaping the antibody response to influenza vaccination. Clin Immunol. 2014 Mar;151(1):55–65.
30. Wang C, Liu Y, Xu LT, Jackson KJL, Roskin KM, Pham TD, et al. Effects of aging, cytomegalovirus infection, and EBV infection on human B cell repertoires. J Immunol. 2013 Nov;192(2):603–611.
31. Parameswaran P, Liu Y, Roskin KM, Jackson KKL, Dixit VP, Lee JY, et al. Convergent antibody signatures in human dengue. Cell Host Microbe. 2013 Jun;13(6):691–700.
32. Fowler A, Galson JD, Trück J, Kelly DF, Lunter G. Inferring B cell specificity for vaccines using a Bayesian mixture model. BMC Genomics. 2020 Feb;21(1):176.
33. Spassov VZ, Flook PK, Yan L. LOOPER: a molecular mechanics-based algorithm for protein loop prediction. Protein Eng Des Sel. 2008 Feb;21(2):91–100.

34. Hamer R, Luo Q, Armitage JP, Reinert G, Deane CM. i-Patch: interprotein contact prediction using local network information. Proteins. 2010 Oct;78(13):2781–2797.
35. Krawczyk K, Liu X, Baker T, Shi J, Deane CM. Improving B-cell epitope prediction and its application to global antibody-antigen docking. Bioinformatics. 2014 Aug;30(16):2288–2294.
36. Riesselman AJ, Ingraham JB, Marks DS. Deep generative models of genetic variation capture the effects of mutations. Nat Methods. 2018 Oct;15(10):816–822.
37. Akbar R, Jeliazkov JR, Robert PA, Snapkov I, Pavlović M, Slabodkin A, et al. A finite vocabulary of antibody-antigen interaction enables predictability of paratope-epitope binding; 2019.
38. Uduman M, Yaari G, Hershberg U, Stern JA, Shlomchik MJ, Kleinstein SH. Detecting selection in immunoglobulin sequences. Nucleic Acids Res. 2011 Jul;39(Web Server issue):W499–504.
39. Yaari G, Uduman M, Kleinstein SH. Quantifying selection in high-throughput Immunoglobulin sequencing data sets. Nucleic Acids Res. 2012 Sep;40(17):e134.
40. Uduman M, Shlomchik MJ, Vigneault F, Church GM, Kleinstein SH. Integrating B cell lineage information into statistical tests for detecting selection in Ig sequences. J Immunol. 2014 Feb;192(3):867–874.
41. Yaari G, Benichou JIC, Vander Heiden JA, Kleinstein SH, Louzoun Y. The mutation patterns in B-cell immunoglobulin receptors reflect the influence of selection acting at multiple time-scales. Philos Trans R Soc Lond B Biol Sci. 2015 Sep;370(1676).
42. Murrell B, Moola S, Mabona A, Weighill T, Sheward D, Kosakovsky Pond SL, et al. FUBAR: a fast, unconstrained bayesian approximation for inferring selection. Mol Biol Evol. 2013 May;30(5):1196–1205.
43. Dunn-Walters DK, Belelovsky A, Edelman H, Banerjee M, Mehr R. The dynamics of germinal centre selection as measured by graph-theoretical analysis of mutational lineage trees. Dev Immunol. 2002 Dec;9(4):233–243.
44. Shahaf G, Barak M, Zuckerman NS, Swerdlin N, Gorfine M, Mehr R. Antigen-driven selection in germinal centers as reflected by the shape characteristics of immunoglobulin gene lineage trees: a large-scale simulation study. J Theor Biol. 2008 Nov;255(2):210–222.
45. Fay JC, Wu CI. Hitchhiking under positive Darwinian selection. Genetics. 2000 Jul;155(3):1405–1413.
46. Neher RA, Bedford T, Daniels RS, Russell CA, Shraiman BI. Prediction, dynamics, and visualization of antigenic phenotypes of seasonal influenza viruses. Proc Natl Acad Sci U S A. 2016 Mar;113(12):E1701–9.
47. Hadfield J, Megill C, Bell SM, Huddleston J, Potter B, Callender C, et al. Nextstrain: real-time tracking of pathogen evolution. Bioinformatics. 2018 Dec;34(23):4121–4123.
48. Gérard A, Woolfe A, Mottet G, Reichen M, Castrillon C, Menrath V, et al. High-throughput single-cell activity-based screening and sequencing of antibodies using droplet microfluidics. Nat Biotechnol. 2020 Mar;.

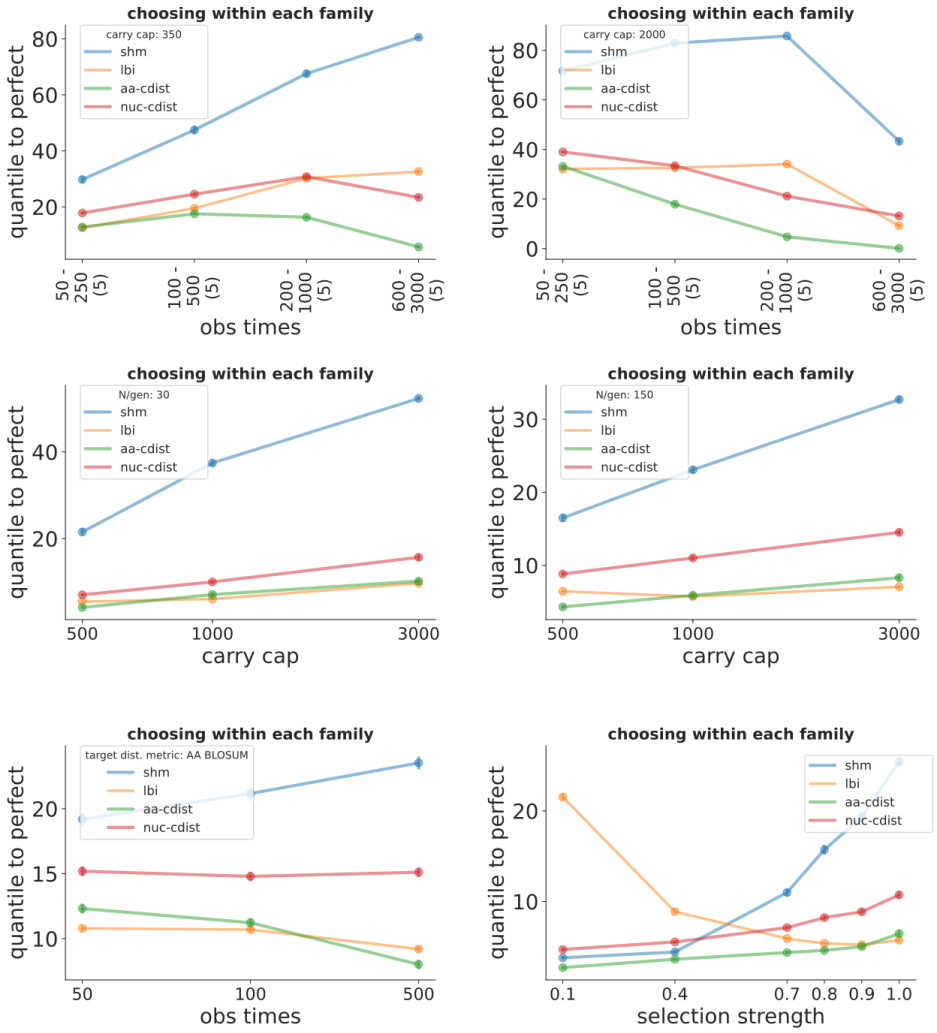
49. Yaari G, Vander Heiden JA, Uduman M, Gadala-Maria D, Gupta N, Stern JNH, et al. Models of somatic hypermutation targeting and substitution based on synonymous mutations from high-throughput immunoglobulin sequencing data. Front Immunol. 2013 Nov;4:358.
50. Matsuda K, Huang J, Zhou T, Sheng Z, Kang BH, Ishida E, et al. Prolonged evolution of the memory B cell response induced by a replicating adenovirus-influenza H5 vaccine. Sci Immunol. 2019 Apr;4(34).
51. Lau AW, Brink R. Selection in the germinal center. Curr Opin Immunol. 2019 Dec;63:29–34.
52. Meyer-Hermann M, Binder SC, Mesin L, Victora GD. Computer Simulation of Multi-Color Brainbow Staining and Clonal Evolution of B Cells in Germinal Centers. Front Immunol. 2018 Sep;9:2020.
53. Murugan R, Buchauer L, Triller G, Kreschel C, Costa G, Pidelaserra Martí G, et al. Clonal selection drives protective memory B cell responses in controlled human malaria infection. Sci Immunol. 2018 Feb;3(20).
54. Buchauer L, Wardemann H. Calculating germinal centre reactions. Current Opinion in Systems Biology. 2019 Oct;Available from: `http://www.sciencedirect.com/science/article/pii/S2452310019300447`.
55. Robert PA, Meyer-Hermann M. A 3D structural affinity model for multi-epitope in silico germinal center simulations; 2019.
56. Dhar A, Ralph DK, Minin VN, Matsen FA IV. A Bayesian Phylogenetic Hidden Markov Model for B Cell Receptor Sequence Analysis. 2019 Jun;.
57. Price MN, Dehal PS, Arkin AP. FastTree 2–approximately maximum-likelihood trees for large alignments. PLoS One. 2010 Mar;5(3):e9490.



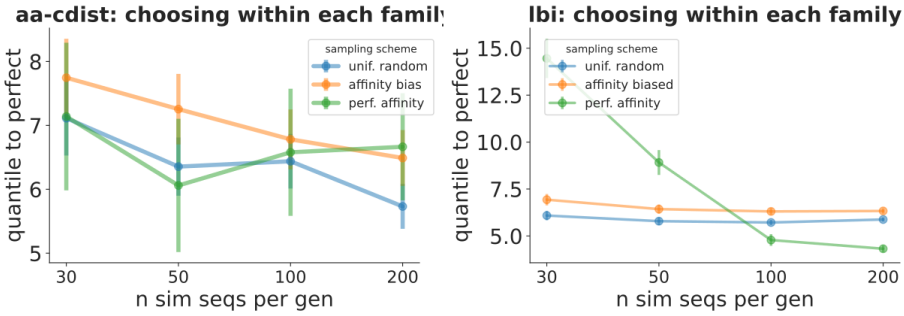
S1 Figure. **Cartoon showing calculation of lbi (left) and lbr (right).** The darkness of each branch represents the exponentially decaying weight factor, which decreases with distance from the node (in red) for which we're calculating the metric. On the left, we show a node with low lbi (top) and high lbi (bottom). At right, in calculating lbr for the red node, we split the tree into two pieces: offspring of the node in the numerator; and parents, siblings, cousins, and their offspring in the denominator.



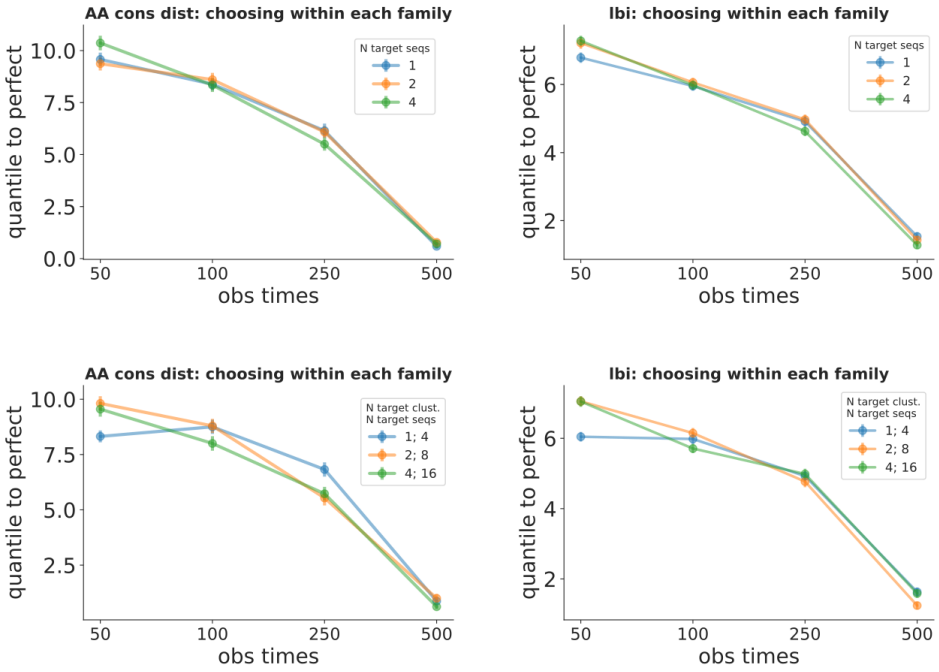
S2 Figure. **Simulation performance for affinity prediction among all families**, similar to Fig 1 but for scans across a variety of different parameters. Performance is shown vs observation times (units of N generations), where sampling occurred at five different time points spanning the indicated values for carrying capacity of both 350 (top left) and 2000 (top right). This is in contrast to Fig 1, where sequences were sampled only at the same, final time point. Performance is also shown vs carrying capacity with 30 (middle left) and 150 (middle right) sampled sequences per family; vs observation time for a non-default affinity calculation utilizing BLOSUM matrices (bottom left); and vs a parameter describing the strength of selection (bottom right). The corresponding within-family plots are in S3 Figure, and many other parameter values (such as plots similar to the middle two, but for N sampled of 75 and 500) are in `https://doi.org/10.5281/zenodo.3728068`.



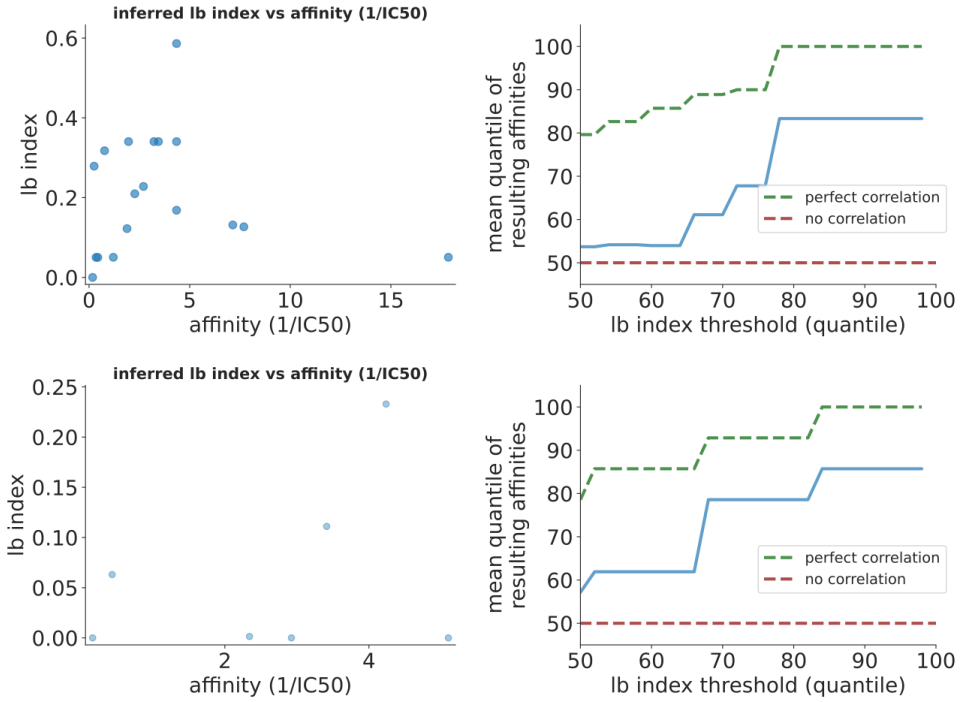
S3 Figure. **Simulation performance for affinity prediction within families**. See the corresponding among-families plots for details (S2 Figure).



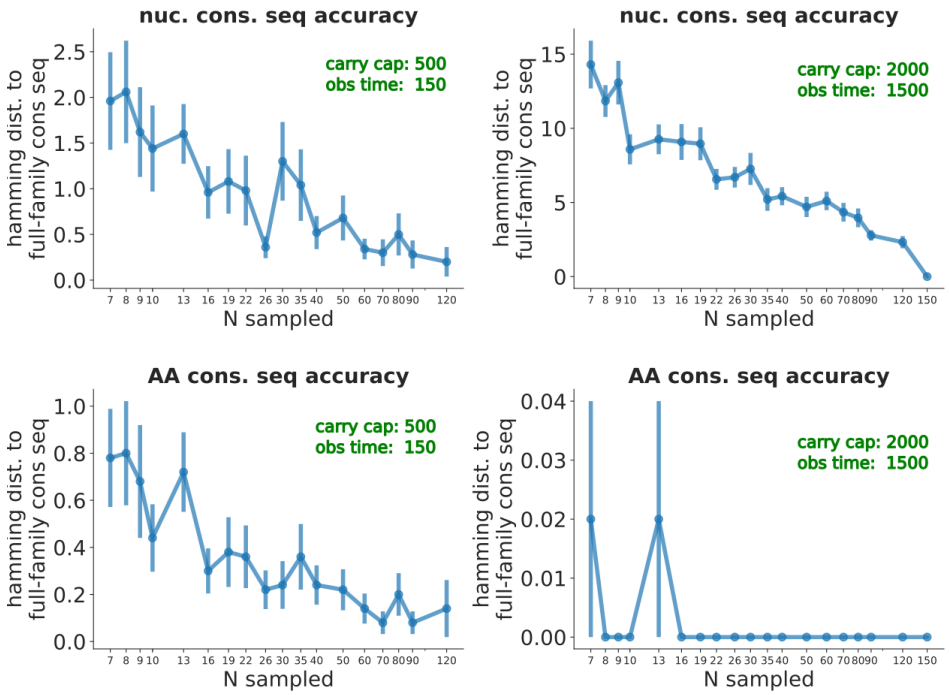
S4 Figure. **Simulation performance for within-family affinity prediction with different sampling schemes vs N sampled sequences per family for aa-cdist (left) and lbi (right).** Schemes shown are "uniform random" (the default, which is shown in all other plots), "affinity biased" (the probability of sampling each cell is proportional to its affinity), and "perfect affinity" (sample the N cells with highest affinity). Corresponding among-families plots, and plots for all other metrics, are at `https://doi.org/10.5281/zenodo.3728068`.



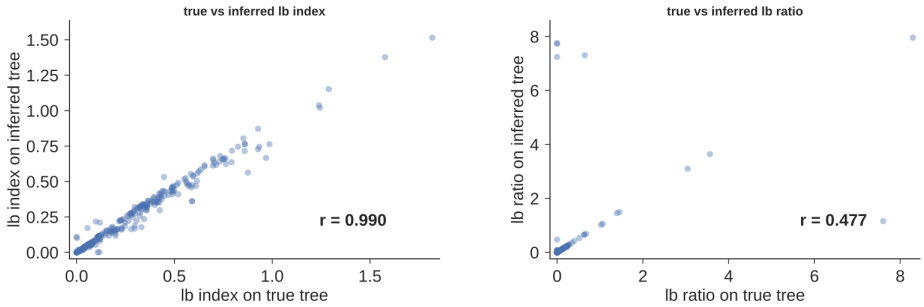
S5 Figure. **Simulation performance for within-family affinity prediction with different numbers of "target sequences" vs observation time for aa-cdist (left) and lbi (right).** The target sequence represents a hypothetical optimal antibody toward which selection is directing the cells (see text). Shown for 1, 2, and 4 independently-chosen target sequences (top); as well as for 4, 8, and 16 target sequences divided among the indicated number of "clusters" (bottom). Corresponding among-families plots, and plots for all other metrics, are at `https://doi.org/10.5281/zenodo.3728068`.



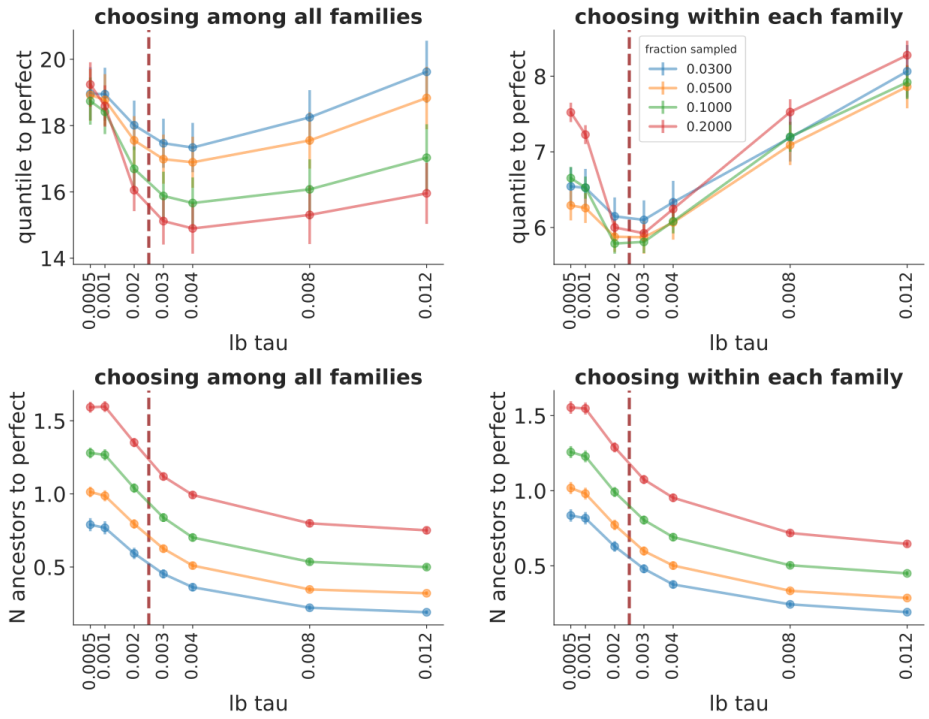
S6 Figure. **Performance on real data for lbi** from [16] (top) and [17] (bottom). See caption to Fig 3



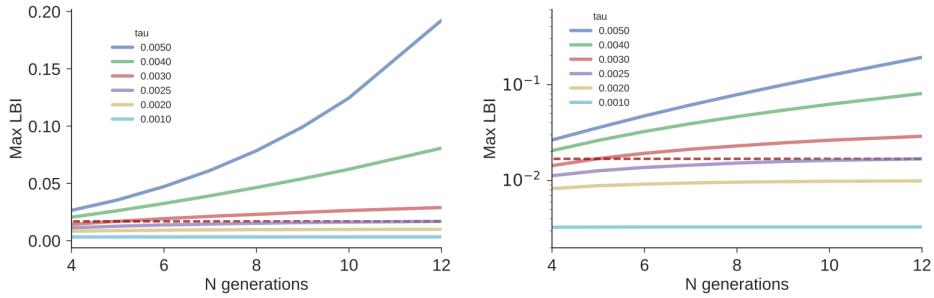
S7 Figure. **Accuracy of the consensus calculation as a function of number of sampled sequences for nucleotide (top) and amino acid (bottom) consensus sequences**. This shows the two extremes of parameters that seem to affect accuracy the most: very early times and small carrying capacities (left) vs very late times and large carrying capacities (right). Each point is the mean ($\pm$ standard error) of 50 families, each with $\simeq$150 sequences. The y value is the hamming distance (i.e. inaccuracy) between the consensus sequence calculated with the indicated number of sampled sequences (x axis) and the consensus sequence calculated on the entire family.



S8 Figure. **Comparison of lbi (left) and lbr (right) calculated on true trees vs inferred trees**, as scatter plots with Pearson's linear correlation coefficient. Inferred trees are made with the approximate, but very fast, method run by default partis (see Methods). In the righthand plot, the handful of points with highly discrepant true and inferred values indicate that for lbr it is worth using a more sophisticated phylogenetic inference program if at all possible.



S9 Figure. **Effect of exponential decay length τ variation on performance** when sampling different fractions of the carrying capacity (colors) for lbi (top) and lbr (bottom) when choosing among all families (left) and within each family (right). Dashed red line corresponds to the value expected from dimensional analysis $1/\ell_{\text{seq}} = 1/400$. The fraction observed corresponds to sampling between 30 and 200 sequences from a carrying capacity of 1000. Note that the vertical ordering of lines (i.e. whether performance is better for higher or lower sampling fractions) is not really informative in this plot – the order reverses depending on whether we sample ancestors or not, i.e. to a large extent it just measures the fraction of sampled sequences that are leaves.



S10 Figure. **Finding a maximum lbi value to use for normalization** at different τ values (colors) with both linear (left) and log (right) scales. Plots show the maximum lbi value among the nodes in a particular reference synthetic "super branchy" tree as a function of tree depth (N generations). The value of $1/\ell_{\text{seq}} = 1/400$ is shown in dashed red; the maximum lbi value converges to an asymptote for τ less than this, while for τ greater than this the maximum diverges.