Supplementary material

Algorithm description

Algorithm 1 Overview of the IgGraph algorithm

Inputs:
$\mathcal{R}$: set of mAb reads
k: k-mer length
Output: L: $|\mathcal{R}| \times 3$ matrix of most likely V/D/J labelings of each read

1: **procedure** IgGraph($\mathcal{R}, k$)
2: $G \leftarrow$Antibody-graph($\mathcal{R}, k$) ▷ create de Bruijn graph over mAbs
3: $G, \mathcal{H}_\mathcal{C} \leftarrow$Add-references($G, \mathcal{V}, \mathcal{D}, \mathcal{J}$) ▷ add V, D, and J
4: **for all** $r \in \mathcal{R}$ **do**
5: **for all** $\mathcal{X} \in \{\mathcal{V}, \mathcal{D}, \mathcal{J}\}$ **do**
6: $C^\mathcal{X} \leftarrow$color-profile($G, r, \mathcal{X}, \mathcal{H}_\mathcal{C}$)
7: $L[r][\mathcal{X}] \leftarrow \max_{x \in \mathcal{X}} \sum_i C^\mathcal{X}[x][i]$
8: **end for**
9: **end for**
10: **return** L
11: **end procedure**

The functions Antibody-graph() and Add-references() create an antibody graph, and add colored reference sequences, as are described in the Methods section of the main text. The selection of labels is shown in line 7 as taking simple maximum of the sum of columns of a specific row j of a color profile matrix C. This is the simplest form, and is improved by selecting thresholds for V gene-segments, as described in the main text and Supplemental Figures S8 and S9.

Algorithm 2 Get color profile for a read r, given a set of reference gene-segments $\mathcal{C}$, and colored antibody graph G

Inputs:
r read
$\mathcal{C}$: set of reference gene-segments
G: colored antibody graph
Output: C: $|\mathcal{C}| \times n$ color profile matrix, where n is the maximum length of a read

1: **procedure** Color-profile($G, r, \mathcal{C}$)
2: $P \leftarrow a_1 a_2 \ldots a_n$ ▷ get path of read r
3: **for all** $c \in \mathcal{C}$ **do**
4: **for** $i \leftarrow 1 \ldots n$ **do**
5: **if** $c \in \mathcal{H}_\mathcal{C}[a_i]$ **then**
6: $C[c][a_i] \leftarrow score$[match]
7: **else if** $c \notin \mathcal{H}_\mathcal{C}[a_i]$ **then**
8: $C[c][a_i] \leftarrow score$[mismatch]
9: **end if**
10: **end for**
11: $C[c] \leftarrow$Propagate-color($P, c, r, C[c]$) ▷ Traverse bulges for color
12: **end for**
13: **return** C
14: **end procedure**

The Propagate-color() function traverses bulges, performs alignment, and propagates color; as described in Figure 5 of the main text.

Simulating Antibody Sequences

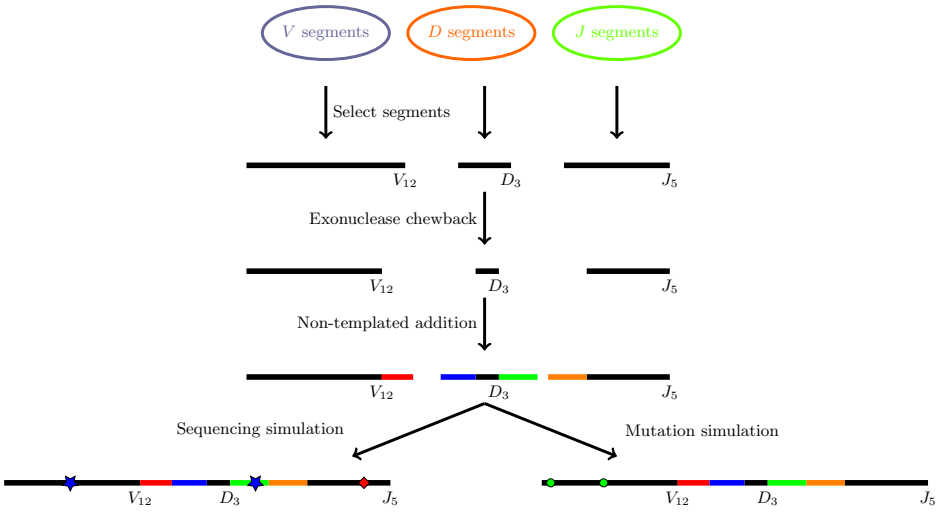


Fig. S1: Diagram of antibody simulation procedure. From pools of V, D, and J gene-segments, one of each is selected for a single smAb. Exonuclease chewback, and non-templated nucleotide additions, are performed based on empirical distributions. Finally, the read data can be generated from the smAb using a sequencing simulator for Roche 454, Illumina, or bypassing sequencing simulator altogether. The Roche 454 simulator can introduce homopolymer insertions and deletions, represented here by blue stars and red diamonds, respectively. Alternatively, the simulated antibodies can obtain mutations along the V gene-segment, shown as green circles.

Mutation distribution

Figure S2 shows the probability of mutating the first 275 positions of a V gene-segment. This distribution was computed from the 23,051 IMGT annotated sequences; as described in the main text. It was used for creating datasets with a given number of mutations, sampled from this distribution, without replacement.

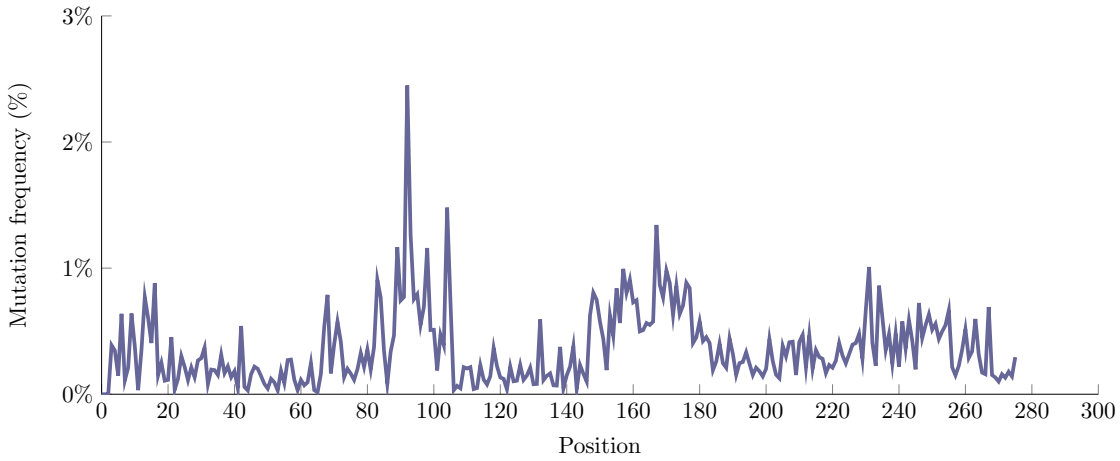


Fig. S2: Distribution of mutations along V gene-segments (CDR3 region is not included). Positions from 0 to 275 are shown on the x-axis, and the y-axis shows the mutation frequency at a given position on the human dataset described in the main text. This mutation distribution shows peaks representing CDR1 and CDR2 regions, and lower probability of mutations in framework regions. The positions are purposefully truncated since deviations from the reference are difficult to attribute to somatic hypermutation events, or to recombination events near the 3' end of the V gene-segment.

Scoring V segments

Somatic hypermutation occurs primarily in V gene-segments, and can have an effect on their labeling. A simple scoring approach can be taken for D and J gene-segments, with match/mismatch penalties, but when considering somatic hypermutation events, sequence properties and positional information must be taken into account. Considering only sequence properties, e.g., 4-mer motifs, ignores a strong signal that mutations occur primarily in CDR1 and CDR2. While only considering position can obviously over-call mutations as well.

We consider both 4-mer motifs and position in our scoring method. For the sake of simplicity, below we assume that the references R and read r are aligned to the same position. In reality, these can differ, an we can tolerate a shift in the read or reference. We wish to compute the probability of observing read r, given reference R, written as: $P(r|R) = \Pi_i P(a_i|b_i, i)$, for a nucleotide in the read a_i and an l-mer in the reference b_i, at position i.

If we instead consider mutations and not nucleotides a_i, then we can write the probability of a read r given a reference R as: $P(r|R) = \Pi_i P(m|b_i, i)$, for $m = \text{mut}$ representing a mutation (i.e., mismatch) and $m = \neg\text{mut}$ representing a match to reference. This turns into:

$$P(m|b_i, i) = \frac{P(b_i, i|m)P(m)}{\sum_{n \in \{\text{mut}, \neg\text{mut}\}} P(b_i, i|n)P(n)} \tag{1}$$

where the probability of a mutation can be the aggregate of non-matching identities at the jth position of 4-mer b_i, denoted b_i^j.

Unfortunately, the data used to compute the conditional joint probability $P(b_i, i|m = \text{mut})$ is sparse when using 67,108 mutation events. To remedy this, we simplify the joint probability to $P(b_i, i|m) = P(b_i|m)P(i|m)$, resulting in the form:

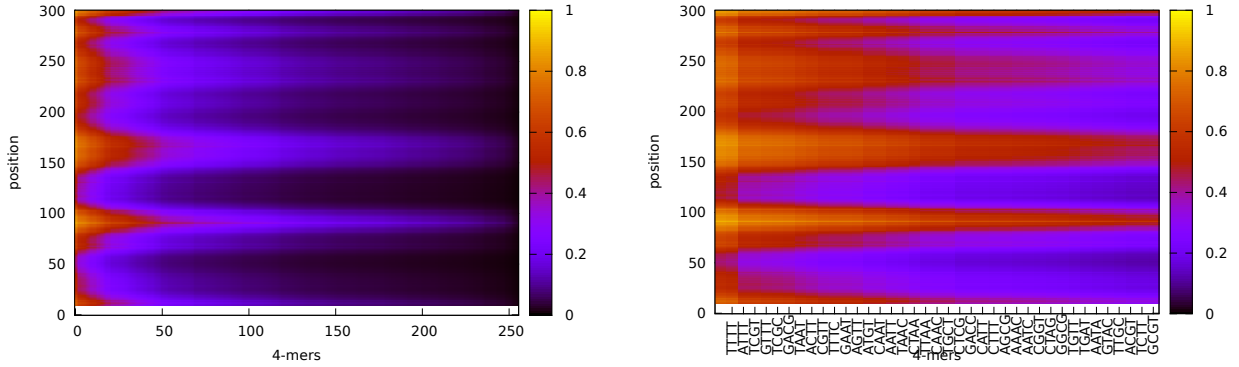
$$P(m|b_i, i) = \frac{P(b_i|m)P(i|m)P(m)}{\sum_{n \in \{\text{mut}, \neg\text{mut}\}} P(b_i|n)P(i|n)P(n)} \tag{2}$$

whose probabilities can be easily computed from the data. This is the same relaxation that is performed for Naive Bayes classifiers to avoid the curse of dimensionality. Further, counts from data used to compute the conditional probability, $P(i|m = \text{mut})$, are modified to smooth the counts. A window of 5 downstream, and 5 upstream positions is used to compute the local average count. This smoothing is to overcome any effects caused by indels in the alignment of the IMGT database.

Plotting $P(m = \text{mut}|b_i, i)$, sorting each 4-mer according to the sum across their positions, provides Figure S3(a). The top 40 4-mers are shown in Figure S3(b). These plots clearly show strong favoring of some 4-mers in CDR1 and CDR2.

Alternate model

The probabilistic model used in IgGraph is detailed in the Supplemental section: Scoring V segments. This is ostensibly a Naive Bayes model, trained on the dataset described in the main text for predicting matching or mutated base pairs in the V gene-segment. For completeness, this model's performance is shown using 5-fold cross-validation on the described dataset in Figure S4. Performance of 4-mers, 5-mers, and 6-mers are shown. Due to a large class imbalance, $\approx 4\%$ samples are mutations, SMOTE [1] is employed to over-sample the minority class, while down-sampling the majority class. This class redistribution is performed only on each fold's training set, while the fold's test set remain unchanged.



(a) All 4-mers (b) Top 40 4-mers

Fig. S3: Probabilities of mutation for a given 4-mer and position. Shown are all 4-mers, across positions from 10 to 300, sorted according to the sum of mutation values across all positions for each 4-mer. The mutation value in each square of the heatmap represents the probability of a mutation at that position, for that 4-mer. (a) All sorted 4-mers reveal CDR1, CDR2, and CDR3 regions. (b) The top 40 4-mers show which motifs are favored in each region.

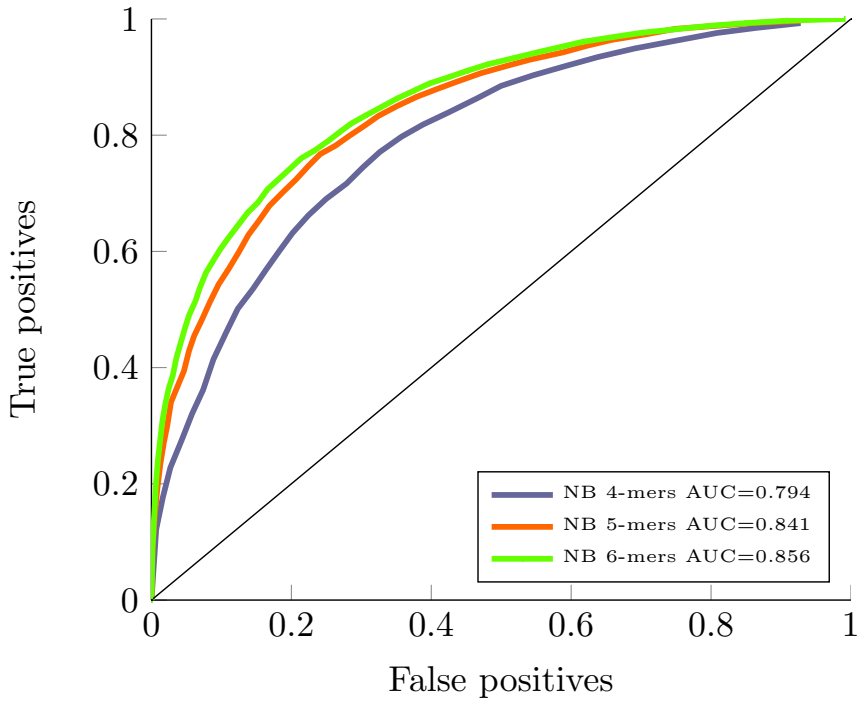


Fig. S4: Receiver operating characteristic (ROC) curve for different models. The ROC curve on 5-fold cross-validated dataset for Naive Bayes using different size k-mers as input. The training datasets are re-sampled using SMOTE [1], while the testing sets are unmodified.

Validating and evaluating predicted classifications

In order to assess the similarity of predicted VDJ classifications of antibody reads, we separate this task into two components: supervised and unsupervised comparisons. Supervised comparisons consist of comparing the predictions to ground truth labels. This supervised validation uses simulated reads since there is no guaranteed way of determining the true antibody references from immunoglobulin sequencing data.

There is also value in assessing the similarity of tools prediction on real data, despite the uncertainty of true labels. In this unsupervised comparison, the similarity of predictions between different tools is assessed.

Comparing the classifications of V, D, J, and total segments at the gene and allele level are performed using the Jaccard index. The junction sequence is compared for absolute equality, since this is often used to characterize the distribution of sizes across repertoires.

Additionally, comparing the clone partitioning is important to determine if different tools would cluster reads into the same, or different, clones which could drastically impact downstream analysis of clone evolution. Clone partitioning can be compared using indices for comparing clustering algorithms, specifically the Rand index, Jaccard index [2], and Fowlkes-Mallows index [3]. These three indices can be computed by comparing pairs of points (i.e., reads), and determining if partition A has clustered them together (1) or separate (0). Similarly, a 1 or 0 can be assigned to the same pair of points for partitioning B. Thus, each pair of points can be in one of the four categories: n_{11} for A and B both placing the pair in the same cluster; n_{00} for both A and B placing the pair in separate clusters; n_{10} for A placing them together while B separates them; and n_{01} for A separating them while B places them together. When A is viewed as the predicted clustering and B as the ground truth, n_{11}, n_{00}, n_{10}, n_{01} can be viewed as true positives, true negatives, false positives, and false negatives, respectively.

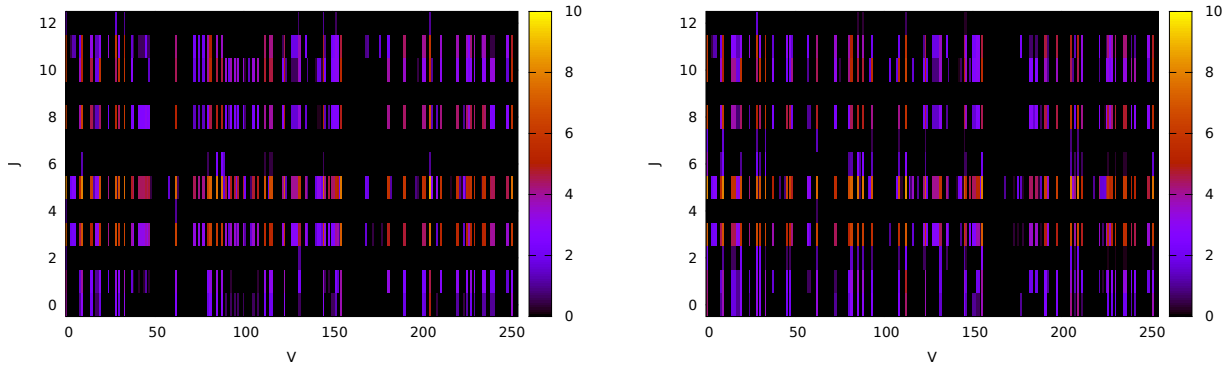
With this formulation, we can compute any of the Rand, Jaccard, or Fowlkes-Mallows indices. However, due to the nature of clone abundances in Ig-seq datasets containing many clones, this results in many clusters. The Rand index will be skewed due to the high number of different clusters. As such, the Fowlkes-Mallows (FM) index [3] provides us with an alternative. The FM index is computed as:

$$FM = \frac{n_{11}}{\sqrt{(n_{11} + n_{01}) * (n_{11} + n_{10})}} \tag{3}$$

and can be interpreted as the geometric mean of the precision in recall in a supervised setting. In our setting the geometric mean can normalize differences in scale between the two partitionings, while still providing a similarity comparison. A tool for simplified comparison of predicted labels, partition ranges, and clonal clusterings is provided as `IgVALVE` (Ig Vdj Antibody Labeling Validation and Evaluation). `IgVALVE` will compute accuracy in a supervised setting, i.e., when using simulated reads; or can compare predictions using the Jaccard index and Fowlkes-Mallows index, as described above, in an unsupervised setting when labels are not available. The validation and evaluation tool `IgVALVE` is available for use at: https://github.com/sbonisso/ig_valve

Stanford S22 VJ analysis

Analyzing the VJ pairings of labeled heavy/light chains can provide an idea of the distribution for the selection of which V and J are favored within a population of B-cells. When comparing different V/D/J labeling tools, this can help show the differences in labeling. The Stanford S22 dataset from [4] is used to highlight differences between tools. The predictions from IgBlast, IMGT, and SoDA are taken from the online resource [1], and IgGraph was run with $k = 21$ since only V and J gene-segments were sought. This returns slightly better results for V/J gene-segments as a longer k is more robust for the longer segments, but obviously, misses D segments that are smaller than this parameter. The values in Figure S5 below are computed by counting VJ pairs for which a prediction for both exists, and if more than one gene-segment is predicted, each predicted label is given an equal proportion of the read. Meaning, if three V segments are predicted for a single read, each obtains 0.33 points. This seeks to normalize for predicting many gene segments. The values for each VJ pair are then log2 transformed for easier comparison.



(a) IgBLAST (b) IMGT

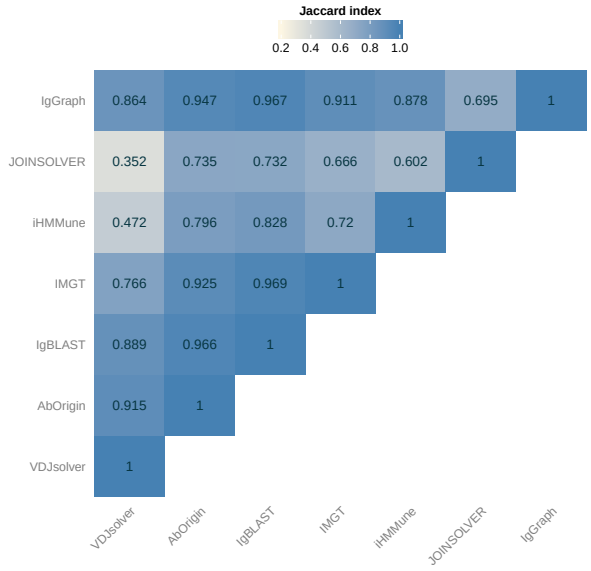
(c) SoDA (d) IgGraph

Fig. S5: VJ pairs for IgBlast, IMGT, SoDA, and IgGraph. The V and J gene-segments are index on the x-axis and y-axis, respectively. Each cell represents a V-J pair, and shows the log transformed count indicated by the colorbar.

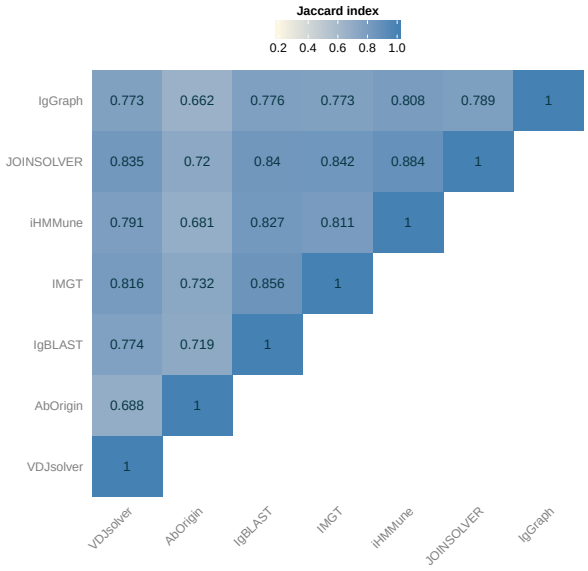
Figure S5 shows the distributions of VJ pairings for various tools. One element to note is how IgBLAST, seen in Figure S5(a), dilutes its predictions across multiple allelic variants.

Table S1 compares performance of various tools for VDJ classification and illustrates that IgGraph performs well for all gene-segments. While the error percentage is higher for V gene-segments, this could

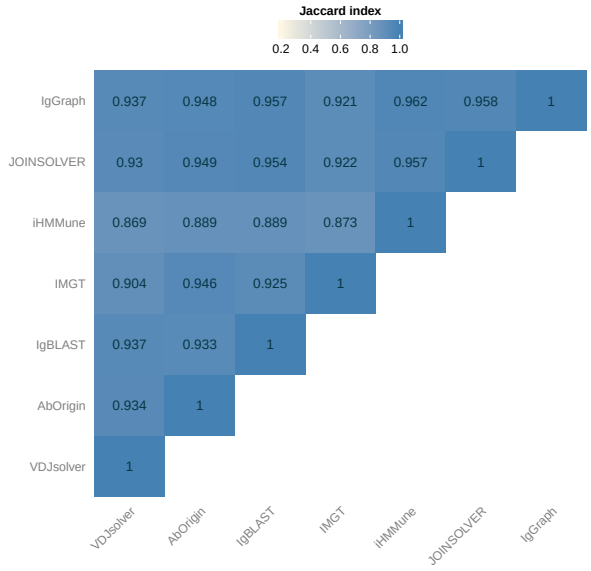
[1] http://www.emi.unsw.edu.au/~ihmmune/IGHUtilityEval/eval_help_datasets.php



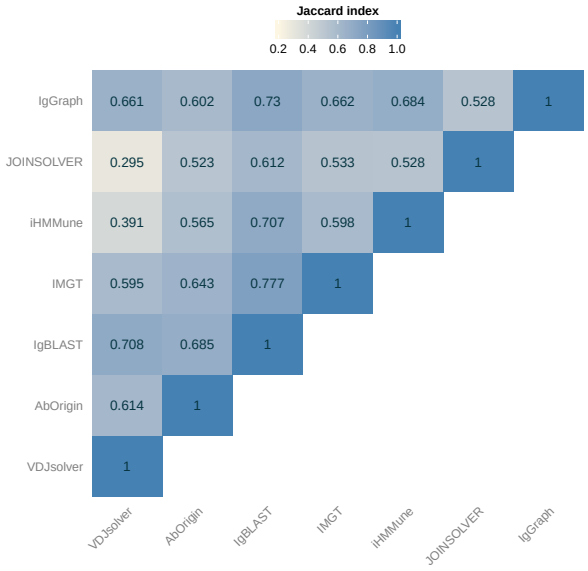
(a) V



(b) D



(c) J



(d) total

Fig. S6: Comparison of predictions from different tools on the Stanford S22 dataset. Similarity is measured by the Jaccard index in predictions of (a) V, (b) D, (c) J, and (d) total.

potentially be further improved with a more sophisticated scoring model than the one we employed. One detail to note, is that, like for most other VDJ classification tools, the majority of errors are mispredictions of allelic variants. These types of errors particularly difficult to distinguish, but our approach (along with most others) is able to correctly identify the correct genotype. An example of a typical error is shown in Supplementary Figure S10.

Table S1: Table of error percentages on Stanford S22 dataset reproduced from [4] with the colored antibody graph (IgGraph) appended. The errors shown are the percentage of incorrect allelic variant reported, and the percentage of incorrect gene reported; a rarer event than incorrect allelic variant. The total column represents the percentage of sequences that include an incorrect gene or allele for either the V, D, or J gene-segments. The results for IgGraph shown are with $k = 11$ and $m = 2$.

		Alleles				Genes		
Citation	Utility	IGHV	IGHD	IGHJ	Total	IGHV	IGHD	IGHJ
[5]	iHMMune-align	3.21	2.21	1.95	7.11	0.21	1.27	0.0
[6]	IMGT	4.90	5.09	1.55	10.87	0.22	2.81	0.0
[7]	IgBLAST	3.84	3.96	0.85	8.39	0.75	2.16	0.0
[8]	Ab-origin	4.06	7.94	2.53	13.74	0.22	5.53	0.0
[9]	JOINSOLVER	6.17	6.93	1.24	7.89	0.86	4.92	0.0
[10]	SoDA	2.68	6.82	1.50	10.37	0.29	6.63	0.0
[11]	VDJSolver	6.87	1.96	0.71	9.09	0.48	0.79	0.0
	IgGraph	5.47	0.93	0.65	6.07	0.15	0.82	0.0

Table S2 benchmarks performance of IgGraph for different values of k-mer sizes (k) along with different sizes of l-mers used for scoring. While increasing k improves performance labeling V gene-segments, the l-mer used influences the performance. The left part of the table referes to 213 alleles, while the right part refers to 55 genes.

Table S2: Table of error percentages for V gene-segments on Stanford S22 dataset for different parameterizations of IgGraph, and different sized l-mers for scoring. Errors involving an incorrect gene, and errors for an incorrect allelic variant, are shown.

	Alleles			Genes		
Parameters	4-mers	5-mers	6-mers	4-mers	5-mers	6-mers
$k = 11$	7.65	5.47	8.90	0.12	0.15	0.74
$k = 15$	6.32	4.29	7.67	0.09	0.10	0.68
$k = 21$	5.94	3.95	7.51	0.09	0.08	0.68

Simulated dataset VJ distribution

The simulated dataset with no mutations, described in the main text, is visualized in Figure S7 by showing the distribution of counts for all VJ pairs.

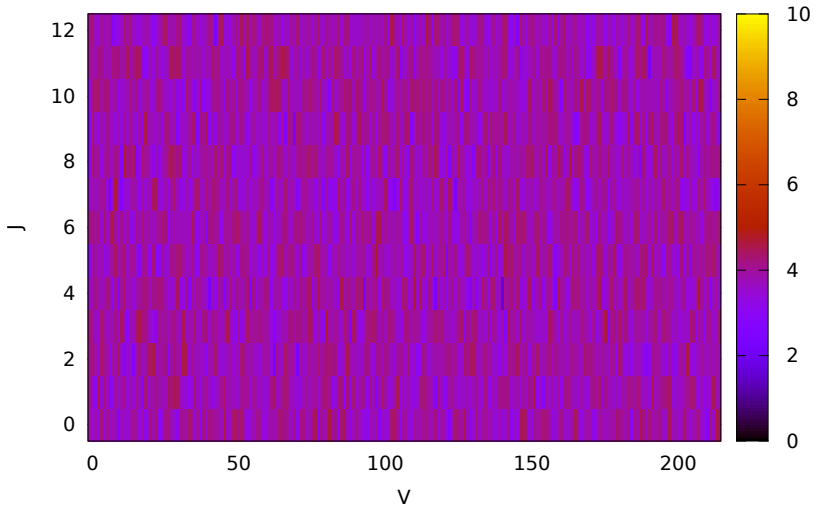


Fig. S7: VJ pairs represented in the simulated antibody dataset containing no mutations. The V and J gene-segments are index on the x-axis and y-axis, respectively. Each cell represents a V-J pair, and shows the log transformed count indicated by the colorbar.

Threshold determination

To select the maximal number of labels to return for each gene segment, the mean accuracy of labelings, varying the threshold for maximal labels, m, is plotted. Providing an additional option allows for improved accuracy, without providing a long candidate list. Manual inspection showed that these improvements are typically in cases where the gene segment is indistinguishable from a different allelic variant. Based on this plot, we select a threshold of 3 for human.

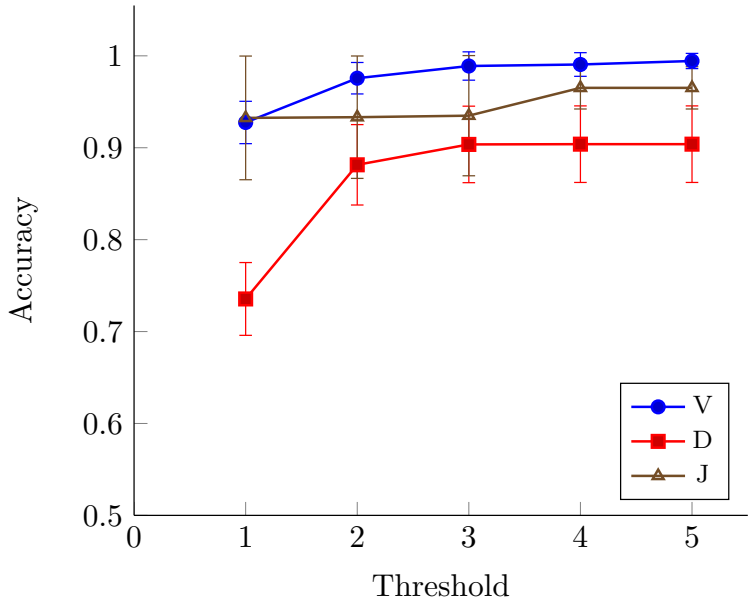
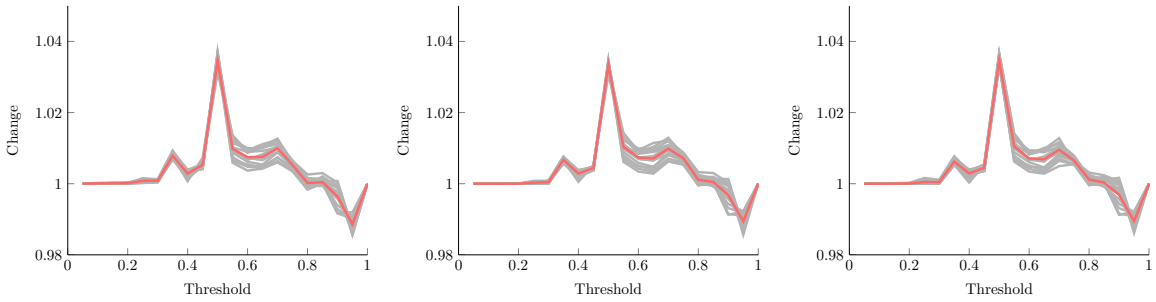


Fig. S8: Mean accuracy of labeling each class of gene segments on human dataset using colored antibody graph with $k = 11$. Providing even a single alternative explanation, i.e. a threshold of 2, improves the accuracy greatly, particularly for D gene segments, while there are diminishing returns for additional options.

The threshold m determines the maximal number of gene-segment labels to report, but we allow for 0 to m labels to be reported for each read. This is based on the probabilistic score of each label. Rather than simply selecting the top m scoring labels, we select the top m scoring labels that cumulatively exceed a threshold t. This means that if there are $m+1$ labels that cumulatively do not exceed t, no label is reported. Conversely, if a single label exceeds t, despite $m = 2$, only that single label is reported. This threshold t is not a user tunable parameter, and was selected by identifying the threshold at which there were diminishing returns correctly identified labels. Figure S9 shows, for each threshold, the ratio between the number of correctly identified gene-segments at that threshold compared to the previous threshold. The threshold when the delta plateaus and/or drops below 1.0 is how a $t = 0.8$ is selected for use in all datasets; all mutation datasets with an even number of mutations and the Stanford S22 dataset.. The curves are generated over the mutation datasets with an odd number of mutations, each curve in gray, and their average shown in red.



(a) $k = 11$ (b) $k = 15$ (c) $k = 21$

Fig. S9: Change in correctly labeled samples when varying the threshold. Each gray curve represents a single dataset with a fixed number of odd mutations, and the red curve is the mean across all used datasets. Thresholds are varied and the change in number of correctly labeled samples is shown on the y-axis. The threshold at which the change plateaus or drops below 1.0 is selected.

Analysis of common errors

One common error IgGraph commits on the Stanford S22 dataset is calling IGHV3-33*01 as IGHV3-33*06, an example of this error is shown in Figure S10 below. The beginning of the read matches perfectly with IGHV3-33*01, at position 166 of the read, where the SNP that distinguishes IGHV3-33*01 from IGHV3-33*06 occurs, the nucleotide matches IGHV3-33*06. This is labeled as IGHV3-33*01, however, the 4-mer surrounding the mismatch is not among the most prevalent 4-mers from the training dataset, and as such, receives a low probability. Disambiguating such a labeling is difficult to perform, with significant confidence, in favor of either gene-segment.

```
IGHV3-33*06  ..........................................   172
IGHV3-33*01  ..........................................   172
read         AGGGGCTGGAGTGGGTGGCAGTTATATGGTATGATGGAAGTAATA    45

IGHV3-33*06  ..........................................   217
IGHV3-33*01  ..........................................   217
read         AATACTATGCAGACTCCGTGAAGGGCCGATTCACCATCTCCAGAG    90

IGHV3-33*06  ..........................................   262
IGHV3-33*01  ..........................................   262
read         ACAATTCCAAGAACACGCTGTATCTGCAAATGAACAGCCTGAGAG   135

                                              SNP
                                               ↓
IGHV3-33*06  ..............................A----   296
IGHV3-33*01  ..............................G..A----   296
read         CCGAGGACACGGCTGTGTATTACTGTGCGAAAGTATCG   173
                                                ↑
                                          junction start
```

Fig. S10: Alignment of read and reference sequences. This alignment shows a read from the Stanford S22 dataset and reference gene-segments IGHV3-33*01 and IGHV3-33*06. This read has ground truth labeling as IGHV3-33*01, and IgGraph assigns it as IGHV3-33*06. Distinguishing between these two is difficult due to the SNP at position 293 of IGHV3-33*01.

Selectng the *k*-mer size

The selection of k can have consequences on performance, and must be chosen carefully. Here, we show how k can negatively impact the antibody graph by creating cycles, or loops, bridging two segments of the antibody that are far away from each other. For example, if a (k-1)-mer is shared between a V segment and a J segment, this can bridge the two gene-segments and cause difficulties in assigning a label. Figure S11 shows the mean number of shared k-mers across all 99,450 human, 4,290 rabbit, and 52,272 mouse combinations of VDJ gene-segments.

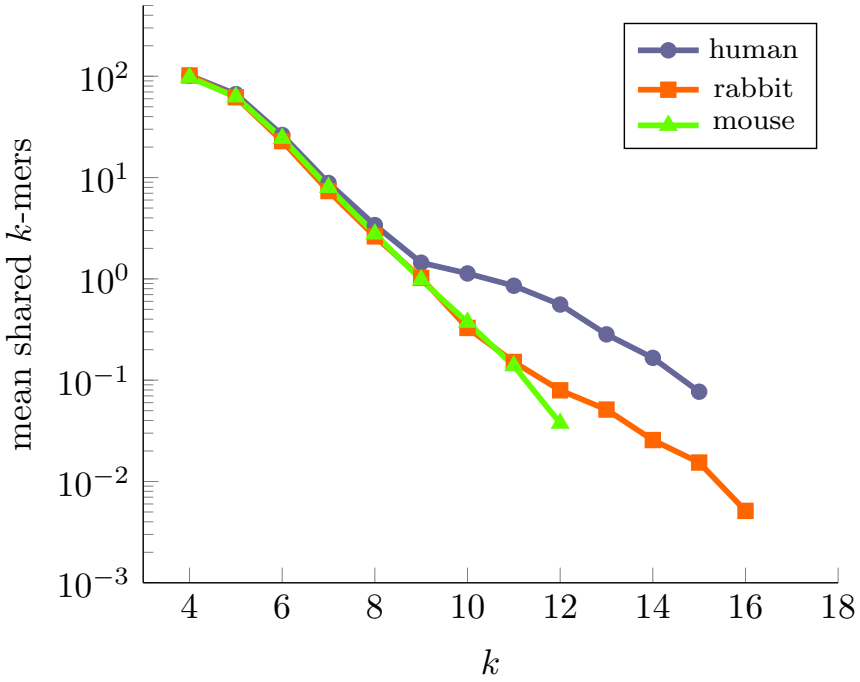


Fig. S11: The effect of varying k over the average number of shared k-mers per antibody transcript. All values of k between 4 and 25 where computed, and each curve is truncated at the value of k where all k-mers are unique within each transcript.

Each curve ends where there no longer exist any shared k-mers among any possible re-arrangement of VDJ gene-segments. This shows that a selection of k=18 would ensure no sharing for any dataset, this selection for k will miss some D gene-segments in human and mouse. For this reason, we are willing to tolerate some redundancy and can see that smaller selections for k can be tolerated. This plot is a simplification of the scenario since it does not take into account non-templated nucleotides and how they may affect k-mer sharing. This is particularly possible since these nucleotides are GC rich.

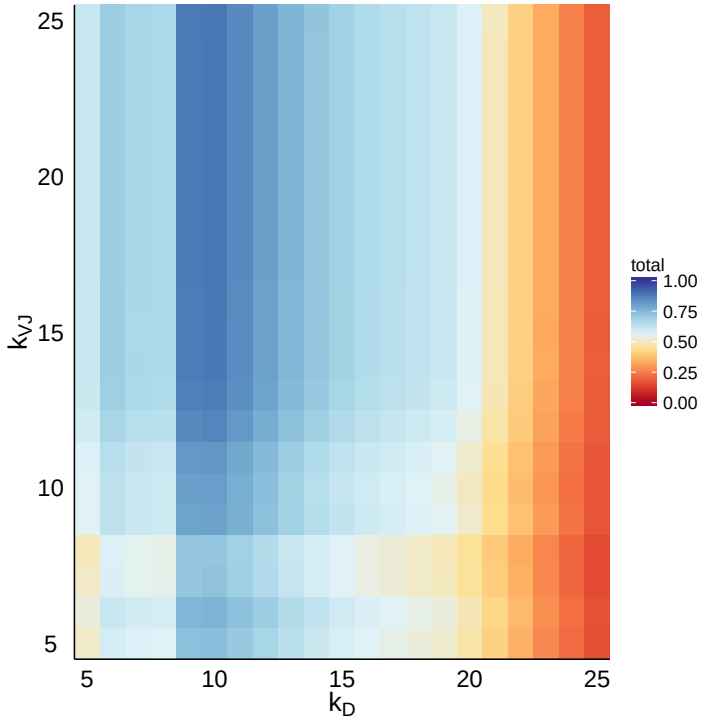


Fig. S12: Accuracy over alleles for different values of k for V/J (k_{VJ}) and D (k_D) gene-segments. Accuracy for total V, D, and J segments over a dataset of 2000 simulated antibody reads is shown. Standard scoring was used.

Supplementary References

1. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.
2. P. Jaccard, *Nouvelles recherches sur la distribution florale*. 1908.
3. E. B. Fowlkes and C. L. Mallows, "A method for comparing two hierarchical clusterings," *Journal of the American statistical association*, vol. 78, no. 383, pp. 553–569, 1983.
4. K. J. Jackson, S. Boyd, B. A. Gaëta, and A. M. Collins, "Benchmarking the performance of human antibody gene alignment utilities using a 454 sequence dataset," *Bioinformatics*, vol. 26, no. 24, pp. 3129–3130, 2010.
5. B. A. Gaëta, H. R. Malming, K. J. Jackson, M. E. Bain, P. Wilson, and A. M. Collins, "iHMMune-align: hidden Markov model-based alignment and identification of germline genes in rearranged immunoglobulin gene sequences," *Bioinformatics*, vol. 23, no. 13, pp. 1580–1587, 2007.
6. X. Brochet, M. Lefranc, and V. Giudicelli, "IMGT/V-QUEST: the highly customized and integrated system for IG and TR standardized VJ and VDJ sequence analysis," *Nucleic Acids Research*, vol. 36, no. suppl 2, pp. W503–W508, 2008.
7. J. Ye, N. Ma, T. L. Madden, and J. M. Ostell, "IgBLAST: an immunoglobulin variable domain sequence analysis tool," *Nucleic Acids Research*, vol. 41, no. W1, pp. W34–W40, 2013.
8. X. Wang, D. Wu, S. Zheng, J. Sun, L. Tao, Y. Li, and Z. Cao, "Ab-origin: an enhanced tool to identify the sourcing gene segments in germline for rearranged antibodies," *BMC Bioinformatics*, vol. 9, no. Suppl 12, p. S20, 2008.
9. M. M. Souto-Carneiro, N. S. Longo, D. E. Russ, H.-w. Sun, and P. E. Lipsky, "Characterization of the human Ig heavy chain antigen binding complementarity determining region 3 using a newly developed software algorithm, JOINSOLVER," *The Journal of Immunology*, vol. 172, no. 11, pp. 6790–6802, 2004.
10. J. M. Volpe, L. G. Cowell, and T. B. Kepler, "SoDA: implementation of a 3D alignment algorithm for inference of antigen receptor recombinations," *Bioinformatics*, vol. 22, no. 4, pp. 438–444, 2006.
11. L. Ohm-Laursen, M. Nielsen, S. R. Larsen, and T. Barington, "No evidence for the use of DIR, D–D fusions, chromosome 15 open reading frames or VHreplacement in the peripheral repertoire was found on application of an improved algorithm, JointML, to 6329 human immunoglobulin H rearrangements," *Immunology*, vol. 119, no. 2, pp. 265–277, 2006.