PeerJ

Non-H3 CDR template selection in antibody modeling through machine learning

Xiyao Long[1], Jeliazko R. Jeliazkov[2] and Jeffrey J. Gray[1,2,3,4]

[1] Chemical and Biomolecular Engineering, Johns Hopkins University, Baltimore, MD, United States of America
[2] Program in Molecular Biophysics, Johns Hopkins University, Baltimore, MD, United States of America
[3] Sidney Kimmel Comprehensive Cancer Center, Johns Hopkins University, Baltimore, MD, United States of America
[4] Institute for Nanobiotechnology, Johns Hopkins University, Baltimore, MD, United States of America

ABSTRACT

Antibodies are proteins generated by the adaptive immune system to recognize and counteract a plethora of pathogens through specific binding. This adaptive binding is mediated by structural diversity in the six complementary determining region (CDR) loops (H1, H2, H3, L1, L2 and L3), which also makes accurate structural modeling of CDRs challenging. Both homology and *de novo* modeling approaches have been used; to date, the former has achieved greater accuracy for the non-H3 loops. The homology modeling of non-H3 CDRs is more accurate because non-H3 CDR loops of the same length and type can be grouped into a few structural clusters. Most antibody-modeling suites utilize homology modeling for the non-H3 CDRs, differing only in the alignment algorithm and how/if they utilize structural clusters. While RosettaAntibody and SAbPred do not explicitly assign query CDR sequences to clusters, two other approaches, PIGS and Kotai Antibody Builder, utilize sequence-based rules to assign CDR sequences to clusters. While the manually curated sequence rules can identify better structural templates, because their curation requires extensive literature search and human effort, they lag behind the deposition of new antibody structures and are infrequently updated. In this study, we propose a machine learning approach (Gradient Boosting Machine [GBM]) to learn the structural clusters of non-H3 CDRs from sequence alone. The GBM method simplifies feature selection and can easily integrate new data, compared to manual sequence rule curation. We compare the classification results using the GBM method to that of RosettaAntibody in a 3-repeat 10-fold cross-validation (CV) scheme on the cluster-annotated antibody database PyIgClassify and we observe an improvement in the classification accuracy of the concerned loops from 84.5% $\pm$ 0.24% to 88.16% $\pm$ 0.056%. The GBM models reduce the errors in specific cluster membership misclassifications when the involved clusters have relatively abundant data. Based on the factors identified, we suggest methods that can enrich structural classes with sparse data to further improve prediction accuracy in future studies.

Subjects Bioinformatics, Computational Biology
Keywords Protein structure, Structure prediction, Rosetta, Antibodies

Submitted 19 June 2018
Accepted 28 November 2018
Published 11 January 2019

Corresponding author
Jeffrey J. Gray, jgray@jhu.edu

Academic editor
Elena Papaleo

Additional Information and Declarations can be found on page 21

DOI **10.7717/peerj.6179**

How to cite this article Long X, Jeliazkov JR, Gray JJ. 2019. Non-H3 CDR template selection in antibody modeling through machine learning. *PeerJ* **7**:e6179 http://doi.org/10.7717/peerj.6179

INTRODUCTION

Antibodies are central to adaptive immunity. They are responsible for recognizing a variety of target molecules known as antigens. They acquire the ability to recognize any one of a diverse set of targets through two biological mechanisms: V(D)J recombination and affinity maturation. These gene-editing mechanisms can produce an enormous quantity of unique sequences, in theory on the order of 10^{13} (*Georgiou et al., 2014*; *DeKosky et al., 2016*; *Hou et al., 2016*), though the antibody repertoire of any single individual comprises only a fraction of the possible sequences. Recent advances in high-throughput sequencing techniques are permitting unparalleled access to the human antibody repertoire (*Boyd & Crowe, 2016*; *Luciani, 2016*), thus furthering our comprehension of immune response to vaccination, infection, and autoimmunity. Beyond sequence data, structural information can provide additional insights about the functions of antibodies. Yet only a very small fraction of antibodies have solved crystal structures in the Protein DataBank, reported as 3,087 structures (*Dunbar et al., 2014*) with a filtered set of 1,940 PDB antibody entries included in PyIgClassify as of August, 2017 (*Adolf-Bryfogle et al., 2015*). Most of these structures are murine (51.15%) and human (35.51%), while repertoire sequencing is rapidly expanding our knowledge of other species. It would be challenging and time-consuming to close the gap between structure and sequence knowledge through experimental structure determination methods. Computational modeling provides a feasible alternative. For example, in chronic lymphocytic leukemia, models of antibody structures added prognostic value over sequence data alone (*Marcatili et al., 2013*). Besides using modeling to develop biological understanding, docking studies of antibodies complexed with various antigens can reveal atomic details of antibody–antigen interactions (*Kuroda et al., 2012*; *Kilambi & Gray, 2017*; *Koivuniemi, Takkinen & Nevanen, 2017*; *Weitzner et al., 2017*). Finally, in antibody design studies, computational approaches can enhance affinity or design an antibody de novo—without prior sequence information (*Lippow, Wittrup & Tidor, 2007*; *Kuroda et al., 2012*; *Dunbar et al., 2016*; *Baran et al., 2017*; *Adolf-Bryfogle et al., 2018*). To be useful, however, computational methods must be able to accurately predict antibody structure.

Typical approaches to antibody structure prediction decompose the problem into three parts based on known antibody-structural features (*Almagro et al., 2014*). Antibodies are typically comprised of a light and heavy chain, both having variable (V) and constant (C) regions (Fig. 1A). While the constant region is important for signaling, it does not vary across antibodies and does not greatly affect the antigen-binding function. On the other hand, the variable region can differ between antibodies and is responsible for recognizing antigens. The variable region can be further divided into a framework region (FR), with greek-key β-barrel topology, and six complementarity-determining regions (CDRs), which are solvent-exposed loops connecting the β-strands comprising the aforementioned β-barrel (Figs. 1B– 1C). The FR is conserved and has a low rate of mutation across antibodies, whereas the CDRs, and in particular the CDR H3, are highly mutable in order to be able to bind a wide variety of antigens (*Schroeder, Cavacini & Cavacini, 2010*). Thus, the antibody modeling problem is often decomposed into (1) homology modeling of light and heavy

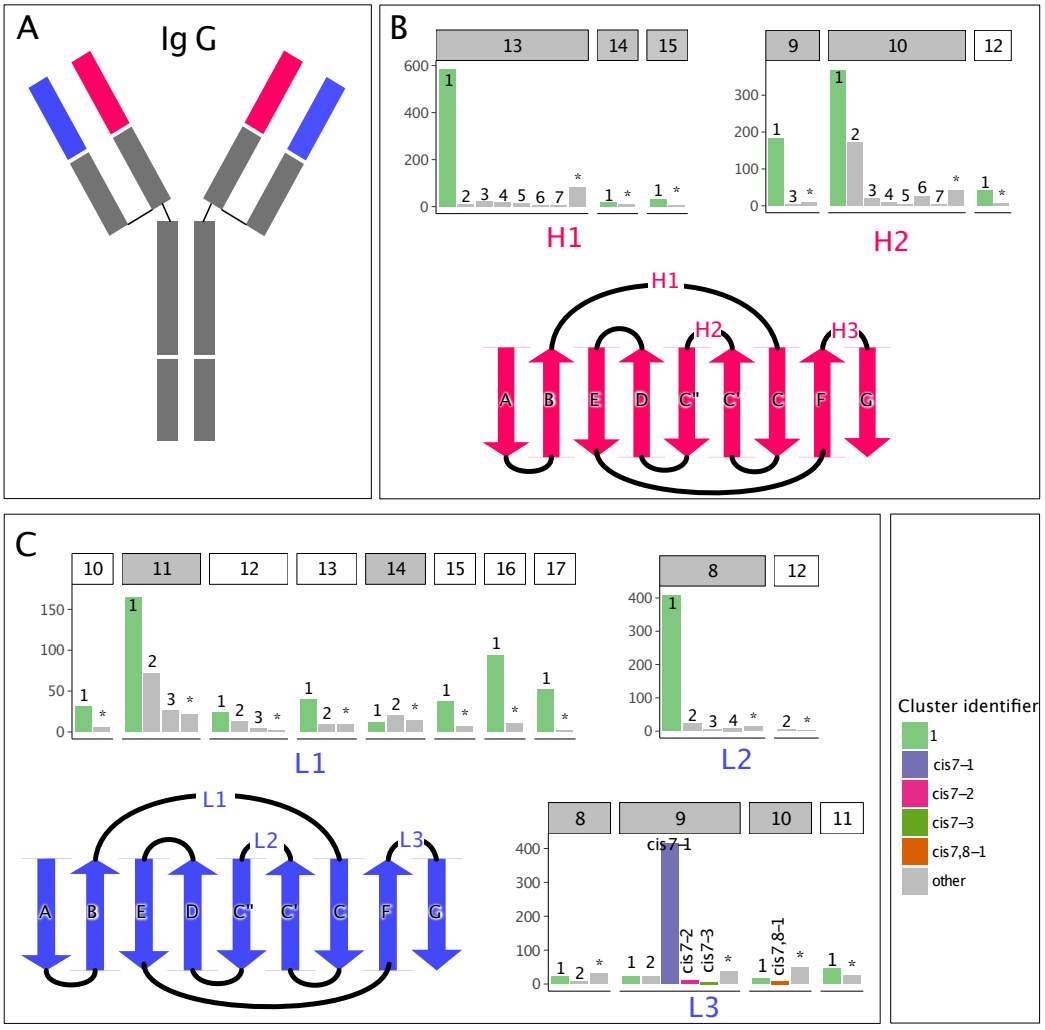

Figure 1 Clusters in canonical CDR loops are not balanced in their number of members. (A) IgG cartoon structure highlighting the variable heavy (VH, red) and light (VL, blue) domains, which bind antigen through their CDR loops. (B) Count of non-redundant CDR loops in the PyIgClassify database for each VH loop-length and -type cluster, with a gray header background indicating adequate numbers for GBM modeling and a white header background indicating inadequate numbers, and a cartoon highlighting the VH beta-strand connectivity and CDR loop location. The CDR H3 is excluded due to its highly variable nature. (C) Analogous to (B), but for the VL. The most populous cluster and clusters possessing cis-prolines are colored.

Full-size DOI: 10.7717/peerj.6179/fig-1

FRs, (2) homology modeling of the non-H3 CDR loops, and (3) *de novo* modeling of the CDR-H3 loop.

Of these three modeling problems, modeling the CDR-H3 loop is the most challenging. For example, an average backbone RMSD of 2.8 ± 0.4 Å was reported over eleven test antibodies and seven modeling approaches in a recent blind assessment (*Almagro et al., 2014*). By comparison, FR modeling was found to achieve sub-angstrom accuracy, on average, for both the light and heavy chains. The quality of the modeling of the

non-H3 CDRs was uneven, with average backbone RMSDs ranging from 0.5 ± 0.1 to 1.3 ± 1.1 Å for RosettaAntibody models of targets in the same assessment (*Weitzner et al., 2014*). This result was surprising since previous studies have found that, when divided by loop type and length (e.g., H1-10), non-H3 CDRs can be structurally clustered and a majority (85%) of the loops assume structures similar to just a few loops structures, called the cluster exemplars (*North, Lehmann & Dunbrack, 2011*). Whether antibody-modeling methods have been using this structural information effectively remains an open question.

In the four most popular methods SAbPred (*Dunbar et al., 2016*), PIGS (*Marcatili et al., 2014*), Kotai Antibody Builder (*Yamashita et al., 2014*) and RosettaAntibody (*Weitzner et al., 2017*), non-H3 CDR loops are generally modeled by homology: a CDR loop with a known structure is chosen as a template structure based on its sequence similarity to the query CDR loop. However, the use of additional structure-based rules, the scoring matrix used to determine sequence similarity, and the database of possible templates all vary among methods.

First, PIGS and Kotai Antibody Builder both use sequence-based rules to identify the structural cluster of the query CDR sequence. If a potential cluster or clusters can be identified, the methods constrain the template search to these clusters. While sequence rules are easy to interpret and can offer deterministic cluster assignments, they are limited in their adaptability and their power—as the number of known antibody structures and sequences grows, analysis by hand becomes more challenging. For example, the current PIGS method uses curated rules from a variety of previous studies (*Marcatili et al., 2014*). For the CDR H1 loop, it has four canonical clusters from four different loop lengths with sequence rules, but according to *North, Lehmann & Dunbrack (2011)* study there are now 17 structural clusters and six loop lengths for the CDR-H1 loop (*North, Lehmann & Dunbrack, 2011*). Another issue is that some clusters lack deterministic, human-identified rules. Kotai Antibody Builder (*Yamashita et al., 2014*) devised sequence rules for cluster identification in accordance with the clusters identified by *North, Lehmann & Dunbrack (2011)*, but in that publication there are not clear sequence rules for distinguishing among H1 clusters. In fact, only a fraction of the remaining non-H3 CDR clusters (26/56) have sequence rules (*Shirai et al., 2014*) and, worryingly, not all sequence rules are comprehensive. For example, under the Chothia numbering convention (*Chothia et al., 1989*), an arginine at position 71 in length 10 CDR-H2 loops can indicate membership to either the H2-10-1 or H2-10-2 cluster, but not all sequences belonging to the H2-10 cluster have that arginine: only 8 out of 155 CDRs in H2-10-1 and 38 out of 42 CDRs in H2-10-2 do. To address this problem and the problem of inadequate sequence-based rule coverage, Kotai Antibody Builder built position-specific-substitution-matrix (PSSM) profiles for each cluster, so that when sequence rules fail, PSSM-based scoring can be used to suggest a cluster (*Shirai et al., 2014*). When assessed using the PyIgClassify antibody dataset, Kotai Antibody Builder correctly identified the cluster in 90% (*Shirai et al., 2014*) of all CDR loops, including the CDR H3. However, it is not clear whether the tested data was excluded from the construction of the PSSM profiles, so the reported accuracy might have been overestimated.

Recent assessments of antibody structural modeling report varying accuracy of non-H3 CDR modeling. When RosettaAntibody was benchmarked in a recent study on

54 antibody targets (*Weitzner et al., 2014*), non-H3 CDR loop modeling achieved sub-angstrom backbone RMSD between the homology-modeled and crystal-structure CDRs in 42/54 (L1), 50/54 (L2), 37/54 (L3), 36/54 (H1), and 42/54 (H2) cases. Meanwhile, using a set of 689 antibody structures and leave-one-out-cross-validation (LOOCV), PIGS (*Marcatili et al., 2014*) was found to capture ∼50% of the modeled non-H3 CDRs with sub-angstrom backbone RMSDs to the native CDRs structures. Finally, SAbPred (*Dunbar et al., 2016*) was tested on the same set of 54 antibodies as RosettaAntibody and resulted in average backbone RMSDs as 1.09, 0.59, 1.00, 0.88 and 0.90 Å for 5 non-H3 CDRs (*Choi & Deane, 2011*). Despite mostly sub-angstrom average RMSDs for all methods and benchmarks, individual models with RMSDs much greater than an angstrom were not rare (*Choi & Deane, 2011*; *Weitzner et al., 2014*; *Almagro et al., 2014*), suggesting a need for special handling of these fail-prone cases. We propose introducing an extra step to non-H3 CDR modeling, where a machine learning approach is used to predict cluster membership and template structures are only selected from the predicted cluster. We hope to improve accuracy by preventing templates coming from a structurally distinct cluster with a large structural distance to the query CDR loop.

Machine learning has been used extensively in protein classification problems. For example, machine learning based methods have accurately predicted protein function (*Radivojac et al., 2013*), folding rate (*Corrales et al., 2015*), super-family levels for fold recognition (*Jain, Garibaldi & Hirst, 2009*), enzyme classes (*Kumar & Choudhary, 2012*), and functional binding sites (*Si, Zhao & Wu, 2015*). For antibody structure, LYRA uses a similarity-score-based template selection method for modeling the antigen-binding site (*Klausen et al., 2015*). Decision-tree-based models have been used on antibodies to predict the structural classes of antigen binding regions. *Chailyan et al. (2011)* used a random forest model to predict non-H3 loop clusters with about 90% accuracy on the data set available then, which had ∼200 antibodies in 10 clusters, before the more complete *North, Lehmann & Dunbrack (2011)* clustering was developed. *Messih et al. (2014)* focused on the CDR-H3 loop and compared the template selection quality of a random forest model to the BLAST similarity score method on a dataset comprising of 401 structures. They found that, on average, the random forest model produced smaller between-query-template RMSD values. Today, more structures are available, and *North, Lehmann & Dunbrack (2011)* have provided a more comprehensive non-H3 CDR clustering scheme. Therefore, a new study using state-of-the-art machine learning prediction performance on canonical CDR loops is needed.

Of many machine learning methods, Gradient Boosting Machine (GBM) was recently shown to yield the best accuracy for structural classification of proteins in the Structural Classification Of Protein database (SCOP) (*Jain, Garibaldi & Hirst, 2009*). The GBM method builds a succession of a tunable number of weak learners, with each learner being a decision tree with tunable tree depth and branch splitting rules. During training, incorrectly classified samples are upweighted in later iterations to converge on final decision trees that fix errors. Since non-H3 CDR loop cluster prediction is a protein structural classification problem, we will adapt this approach.

In this work, we attempted to increase the quality of CDR structural template selection by using the machine learning method GBM. For a relevant and fair assessment, we evaluated the quality of template selection rather than that of the final model. The assessment was performed on the comprehensive dataset PyIgClassify (*Adolf-Bryfogle et al., 2015*), comparing the original RosettaAntibody structural template identification method and the GBM method developed herein. As the disparities of cluster member sizes can affect the performance of GBM (*Sun et al., 2007*), we surveyed various techniques for overcoming the data imbalance problem. Approaches vary, from down-sampling the majority class to up-sampling the minority classes, or even adding synthetic members to balance the size of the clusters (*Chawla et al., 2002*; *Blagus & Lusa, 2013*). Previous results suggest that the best approach depends on the specific data set and size (*Kuhn & Johnson, 2013a*). In our study, we used the up-sampling strategy.

We show that (1) the new GBM can better identify the query CDR's structural cluster than RosettaAntibody and (2) selecting structural templates from within the query cluster results in lower RMSD templates than selecting outside the cluster. The GBM models also recapitulate previously known sequence motifs and identify new ones. The GBM models find that the presence or absence of a single residue on its own is not sufficient to assign a sequence to a specific structural cluster. Instead, the combination of residues in the query sequence is important for assigning a probable cluster. These findings suggest that incorporating machine learning methods may achieve closer-to-native templates selection during non-H3 CDR homology modeling and realize an automated feature selection, surpassing the manual curation of sequence rules.

MATERIALS AND METHODS

Dataset

We compared the CDR structural class prediction performance of GBM and blindBLAST on the non-redundant CDR loops in the PyIgClassify database (http://dunbrack2.fccc.edu/PyIgClassify/Download/Download.aspx). The structures and clusters were downloaded in February 2017 by selecting the "CDRs and clusters of non-redundant sequences for a given CDR" database. The database contains antibody structures from the PDB with 2.8 Å or better resolution and an 0.3 R-factor cutoff, while excluding non-proline *cis* loops or loops with highly improbable conformations (*North, Lehmann & Dunbrack, 2011*). The set of non-redundant canonical CDR loops from the database is partitioned by CDR loop type and length. It contains 3,558 total loops from 1,153 distinct antibody structures.

In PyIgClassify, CDR loops are partitioned by type (e.g., L1 or L2) and length (e.g., 10 or 11) and clustered such that the members of each cluster are more structurally similar to their cluster exemplar than to the exemplar of any other cluster, with the exemplar as defined in *North, Lehmann & Dunbrack (2011)*. The distribution of CDR cluster membership is unbalanced, with each CDR loop and length pair having one well-populated, dominant cluster and many sparsely populated, non-dominant clusters. In our study, CDR loops which were unable to find a nearest neighbor cluster within certain dihedral angle distance and clusters smaller than three members were merged into a single cluster labelled "none". The cluster member size distribution by CDR loop and length type is shown in Fig. 1.

Structural class prediction methods

We employed two methods, blindBLAST and GBM, for CDR structural class prediction. The blindBLAST approach comes from the current version of RosettaAntibody (*Weitzner et al., 2017*), which identifies template non-H3 CDR loops through a BLAST search against CDRs of the same length and type using the PAM30 matrix to rank sequence similarity. The BLAST parameters used are:

```
-substitution_matrix PAM30 -word_size 2 -max_target_seqs 3000 -evalue
2000
```

The template loop with the most sequence similarity to the query is then selected for grafting and further modeling. We refer to this approach as "blindBLAST", as it does not utilize CDR structural cluster information but rather identifies the structural class of a CDR loop implicitly by choosing a template with the highest bitscore. On the other hand, we trained supervised GBM models for each CDR loop and length type. Each model learns to predict the structural class (synonymous to the structural cluster) from the labelled CDR sequences, including the 10 flanking residues on either side. Sequences were vectorized by one-hot-encoding (*Beck & Woolf, 2000*): the observed amino acid is represented by a one and the other possible 19 amino acids are zeros. Thus, a CDR loop of length 10 is represented by a 30*20 matrix.

We trained our GBM model by searching a hyper-parameter grid in a nested 3-repeat 10-fold CV scheme. As typical for nested CV, the grid search was performed in the inner loop (consisting of 3-repeat 10-fold CV on the training folds for each iteration of the outer loop), and model accuracy was assessed over the outer loop. We used CV instead of a single training/test data split to counter data sparsity. Fold splitting was stratified, ensuring that the composition of each fold was representative of the whole dataset (*Kohavi, 1995*). To counter the unbalanced sample problem, classes with low population were up-sampled to either 50 or to the number of samples in the most popular cluster, whichever was lower (*Dittman, Khoshgoftaar & Napolitano, 2015*; *Sun, Kamel & Wang, 2006*). The hyper-parameters yielding the highest estimated model accuracy were used for the final model. All machine learning was performed using the Caret package (*Kuhn, 2008*).

Comparisons of different methods

For both blindBLAST and GBM, an error case was identified when the query cluster did not match to the predicted (template) cluster. The number of error cases and the corresponding accuracy were calculated for each loop and length type for each repeat and then averaged over the three repeats. To further analyze failures, we counted and compared the specific misclassifications (i.e., the number times a cluster A to cluster B misclassification occurred) for both GBM and blindBLAST.

The χ^2 goodness-of-fit test was run on each loop type and length combination to test whether the blindBLAST errors differed significantly from random assignment. The χ^2 was calculated as

$$\chi^2 = \sum_{\mathrm{Y}} \sum_{\mathrm{X}} \frac{(\epsilon^{\mathrm{X}\to\mathrm{Y}}_{\mathrm{blindBLAST}} - E[\epsilon^{\mathrm{X}\to\mathrm{Y}}])^2}{E[\epsilon^{\mathrm{X}\to\mathrm{Y}}]}, \quad (1)$$

where $\epsilon^{\mathrm{X}\to\mathrm{Y}}_{\mathrm{blindBLAST}}$ is the average error count of misclassifying cluster X to cluster Y in 3-repeats-10-fold cross-validation, $E\left[\epsilon^{\mathrm{X}\to\mathrm{Y}}\right] = n_{\mathrm{X}} \cdot P_{\mathrm{Y}}$ is the expected error count of misclassifying cluster X to cluster Y, with n_{X} as the number of samples in cluster X in the dataset and P_{Y} as the fraction of cluster Y samples. The significance value $p^{\mathrm{X}\to\mathrm{Y}}$ corresponding to the $X \to Y$ misclassification is then found comparing to the χ^2 distribution with degrees of freedom equal to the number of clusters. When χ^2 exceeds the critical value, it means that blindBLAST is not different from random (H_0 is rejected).

Additionally, raw error cases counts are confounded by the member size differences between structural classes, so we can't compare the values directly. Instead, for a particular misclassification (e.g., H2-10-1 incorrectly classified as H2-10-4), we compare its blindBLAST error count to a simulated random error count distribution using a two-tailed hypothesis test at an 0.05 significance level (Eq. (1)). The random error counts are generated from 10,000 iterations of randomly assigning cluster identies in proportion to the naturally occurring rate. The comparison between the average blindBLAST error count over three repeats of 10-fold cross validation ($\epsilon^{X\to Y}_{\mathrm{blindBLAST}}$) and its simulated distribution ($\epsilon^{X\to Y}_{n,\ \mathrm{random}}$) hinges on the significance value, p, which is determined as the proportion of the random simulated error counts with smaller values than $\epsilon^{X\to Y}_{\mathrm{blindBLAST}}$:

$$
\begin{aligned}
H_0 &: \epsilon^{X\to Y}_{\mathrm{blindBLAST}} \text{ is equivalent to } \epsilon^{X\to Y}_{\mathrm{random}}, \\
p\left(\epsilon^{X\to Y}_{\mathrm{blindBLAST}}\right) &= \frac{\sum_{n=1}^{10000} I\left(\epsilon^{X\to Y}_{\mathrm{blindBLAST}}, \epsilon^{X\to Y}_{n,\ \mathrm{random}}\right)}{10000}, \\
\text{where } I\left(\epsilon^{X\to Y}_{\mathrm{blindBLAST}}, \epsilon^{X\to Y}_{n,\ \mathrm{random}}\right) &= \begin{cases} 1, \text{if } \epsilon^{X\to Y}_{\mathrm{blindBLAST}} \geq \epsilon^{X\to Y}_{n,\ \mathrm{random}} \\ 0, \text{if } \epsilon^{X\to Y}_{\mathrm{blindBLAST}} < \epsilon^{X\to Y}_{n,\ \mathrm{random}} \end{cases}
\end{aligned}
\tag{2}
$$

We reject the null hypothesis, if $p \leq 0.025$ (meaning blindBLAST misclassifies with significantly lower error than random) or if $p \geq 0.975$ (meaning blindBLAST misclassifies with significantly higher error than random). Three categories of misclassifications are generated using the p values (Table S1).

Structure and sequence metrics

To better understand why misclassification may have occurred, we computed two structural metrics and compared each of the metrics between the correct cases and the incorrectly predicted cases using the blindBLAST. Defining the dihedral angle distance by following (*North, Lehmann & Dunbrack, 2011*),

$$
D(i,j) = \sum_{r=1}^{N} D\left(\phi^i_r, \phi^j_r\right) + D\left(\psi^i_r, \psi^j_r\right), \tag{3}
$$

where $D(\theta_1, \theta_2) = 2(1 - \cos(\theta_1 - \theta_2))$, N is the length of the loop, r is the residue number, and i and j represent each CDR identity in the CDR pair for the dihedral angle distance calculation. First, we calculated the dihedral angle distance of every query case to the exemplar of its corresponding cluster and compared the distance distributions for correctly and incorrectly classified cases. Second, we counted the number of structural neighbors in the cognate cluster for all CDR loops. A structural neighbor is defined to be any CDR loop with dihedral angle distance to the query less than 1/15th of the radius of that cluster, where

the radius is the largest dihedral distance between the cluster exemplar and any CDR loop in the cluster. We compared the distributions of the number of structural neighbors for the correctly and incorrectly classified cases. We also computed the dihedral angle distance between clusters that are significantly better distinguished by the blindBLAST than random by the χ^2 test, and we compared the values to those of cluster pairs that are found to be nonsignificant in the χ^2 test.

In addition to investigating structural features, we extracted sequence features from the tuned GBM models based on the scaled feature importance. Absolute importance was calculated by determining how much a decision tree split reduces Gini impurity (*Louppe et al., 2013*) and then summing over all node-size-weighed reductions on splits corresponding to that feature over all boosting trees (*Kuhn & Johnson, 2013b*). The importance was then scaled to values from 0 to 100.

The code for the model generation and analysis can be found in https://github.com/xlong2/machine-learning-cdr.

RESULTS

BlindBLAST is more accurate than random assignment for all but one CDR loop type and length

In blindBLAST, cluster assignment accuracies varied among the different CDR loop and length types from below 50% to almost 100% according to 3-repeat 10-fold cross-validation, as shown in Fig. 2A. In most of the cases where the clusters of the query and the template CDR did not match, a more near-native structural template could be found if the BLAST search was restricted to within the query cluster (Fig. 2B). This suggests that identification of the query cluster could lead to selection of lower-RMSD templates.

To improve the accuracy with which we identify query sequences' CDR clusters, we first sought to understand why the accuracy of CDR cluster identification varies across loop lengths and types. We found that accuracy is affected by (1) the number of clusters in each loop length and type, (2) the number of loops populating each cluster, and (3) the total number of loops of a given length and type (Fig. S1). First, we found that loops with a larger number of clusters tended to have lower accuracy. For example, H1-13 has eight clusters and a blindBLAST assignment accuracy of 78.6 $\pm$ 0.4% whereas H2-9 has three clusters and an accuracy of 91.2 $\pm$ 0.5%. Second, we found that loops with uniform populations among clusters had lower accuracy. For example, H2-10 and H1-13 both have eight clusters, so based on our first observation we expected their accuracy to be similar. It is not: H2-10 has an accuracy of 73.3 $\pm$ 0.1% whereas H1-13 has an accuracy of 78.6 $\pm$ 0.4%. Analyzing the populations of the clusters for each loop, we observed that clusters H2-10-1 and H2-10-2 have a similar number of CDRs whereas clusters H1-13-1 have many more CDRs than any other H1-13 cluster (Fig. 1). Third, accuracy can be limited by sparse data. We have observed lower accuracies for loops with a small number of structures for a length and type. This is exemplified by H1-14, L1-12, L3-8, L3-10 having the worst accuracies among all loops: 45.1 $\pm$ 0.1%, 72.9 $\pm$ 4.8%, 65.4 $\pm$ 3.7%, 62.0 $\pm$ 2.8% (with a total number of 30, 43, 62, 72 loops), respectively.

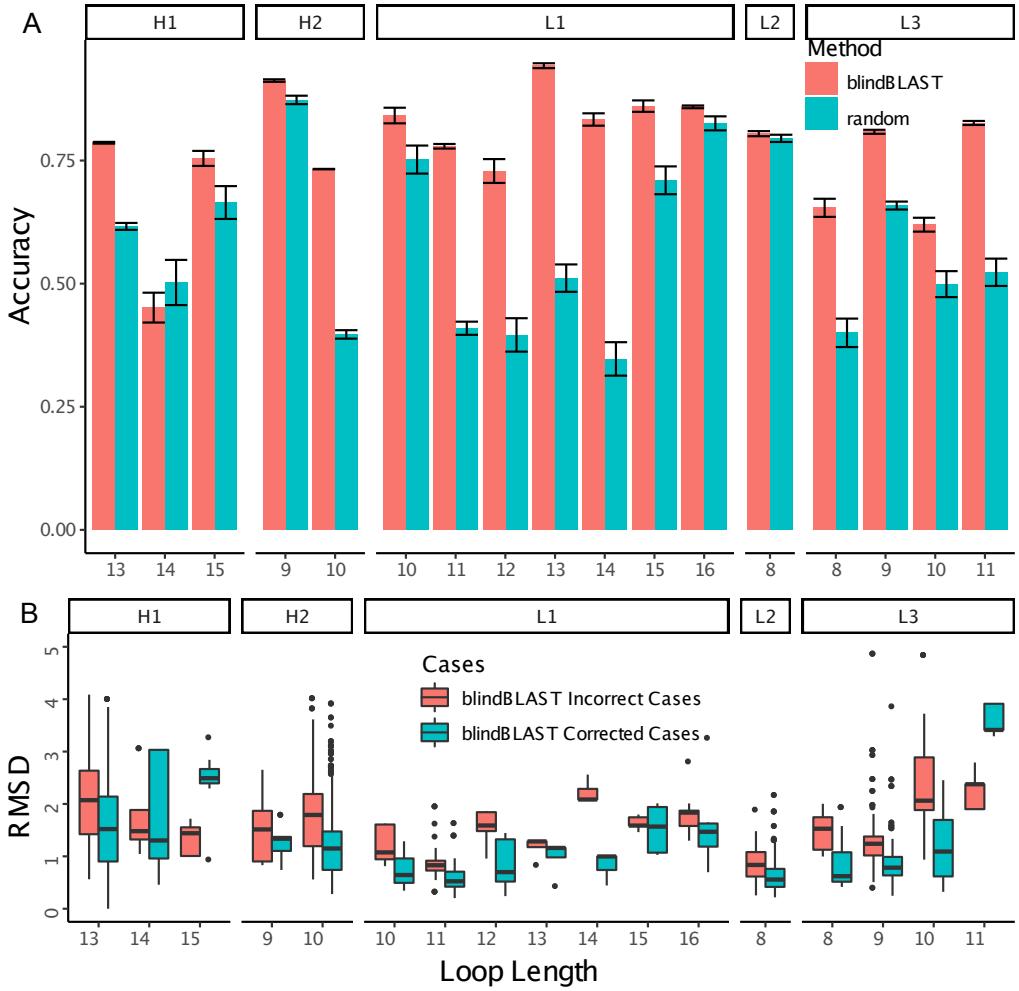

Figure 2 BlindBLAST can more accurately assign CDR loops than random, and, when assignment is to the cognate cluster, RMSD is lower. (A) Cluster assignment accuracy comparison between blindBLAST and random assignment. Error bars show the standard deviation of the accuracy. For blindBLAST, the standard deviation is calculated across the 3-repeat 10-fold cross validation, whereas for the random model it is the standard deviation of accuracies over 10,000 iterations. In all but one case (H1-14), blindBLAST determines clusters more accurately than random. (B) Comparison of query–template RMSD when the loop is selected by BLAST from the incorrect versus the "corrected" cluster. Incorrect cases are loops with templates from clusters other than the query and they are "corrected" by sequence alignment to only templates within their cluster. In most cases, BLAST finds lower-RMSD templates within the cognate loop cluster than outside of it, indicating that correct cluster determination from sequence can lead to a better structural template. However, two loop types (H1-15 and L3-11) do not have lower RMSD templates in their cognate cluster.

Full-size DOI: 10.7717/peerj.6179/fig-2

In addition, the χ^2 goodness-of-fit test suggests that blindBLAST may classify some loop and length types no better than a random model (Table S2). The sparse loop and length types H1-14, H1-15, H2-9, H2-12, L1-10, L1-15, L1-16, L1-17 are included. The more populated loop and length types including H1-13, H2-10, L2-8 and L3-9 have significantly "better than random" predictions results. Furthermore, errors in each loop and length

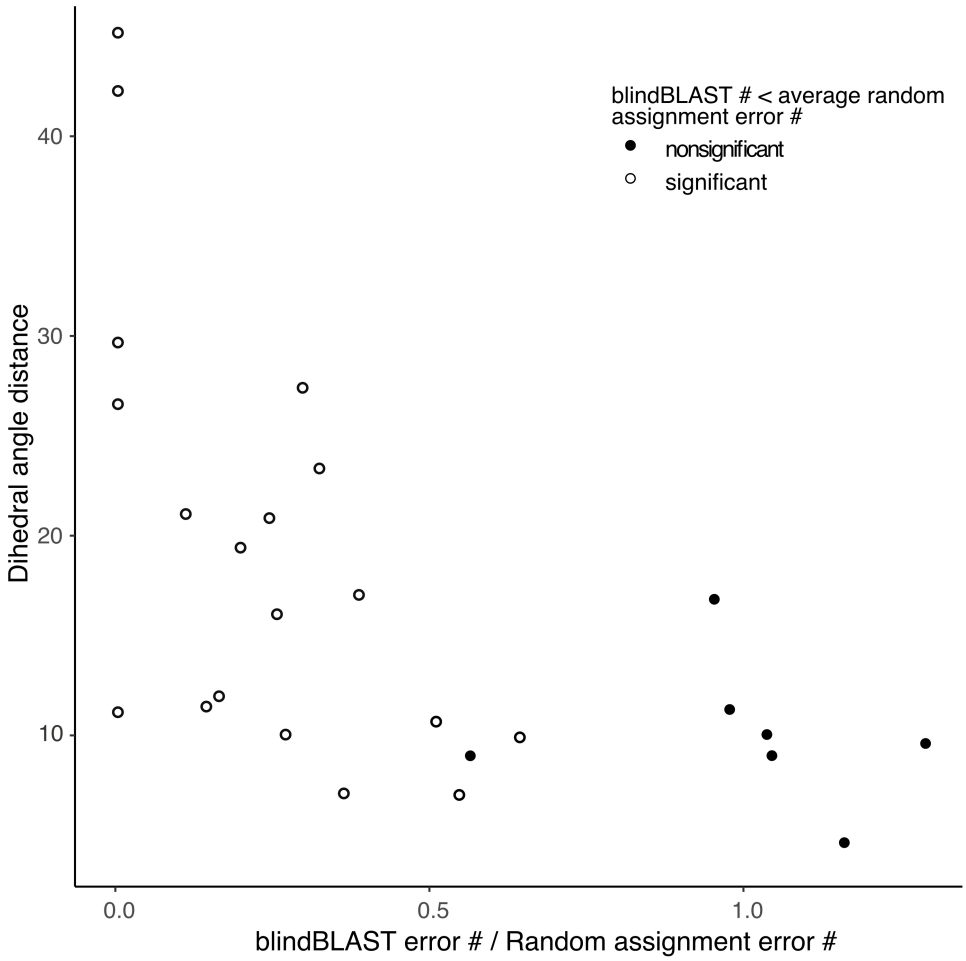

Figure 3 Small distance between a pair of clusters is associated with higher likelihood of misclassification using blindBLAST. Each point corresponds to a misclassification from cluster X → cluster Y, with a minimum of three misclassifications required. The *x* axis is the ratio of the average error count for blindBLAST to the average error count for random assignment. The *y* axis is the dihedral angle distance between the two clusters' exemplars. BlindBLAST misclassifications are significantly fewer than random when dihedral angle distance between exemplars is large.

Full-size DOI: 10.7717/peerj.6179/fig-3

type consist of several different misclassifications. From the random assignment test, we identified different misclassifications where blindBLAST performs better or worse than random (Table S3 , Fig. S2). The "random-like" misclassifications typically have smaller distances between the true and incorrectly predicted cluster than those of the "better than random" misclassifications (Fig. 3).

We also examined whether where the query CDR is situated inside its cluster affects its chance of being misclassified. We quantify a query CDR inside its cluster by two metrics: (1) the dihedral distance of the query CDR to its cluster exemplar and (2) the number of structural neighbors to the query CDR. The distributions of query–exemplar dihedral-angle distances (Fig. 4) and suggest that query CDRs that are more distant from their corresponding cluster exemplars are more likely to be misclassified by blindBLAST.

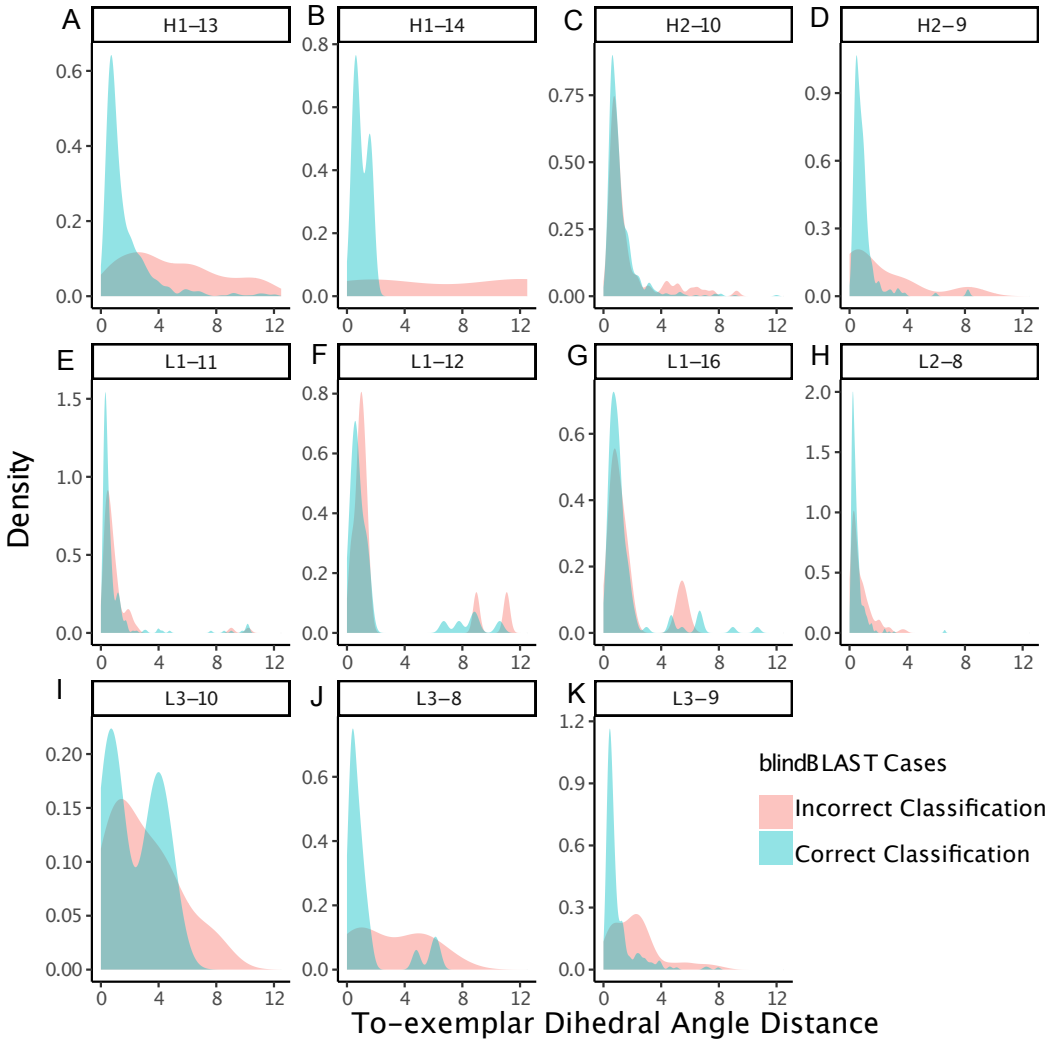

Figure 4 CDRs that are misclassified have a relatively large dihedral angle distance to their cluster exemplars. Density of dihedral angle distances between the query loops and their cluster exemplar loops. For most loop lengths and types (e.g., H1-13, H2-9, L1-15, L3-8, L3-9, and L3-10), the misclassified distribution has more density at larger dihedral angle distances with respect to the correctly classified distribution. The skewedness indicates that for many loops, if a query CDR is distant to its corresponding structural exemplar, then it is more likely to be incorrectly classified using the blindBLAST method.

Full-size DOI: 10.7717/peerj.6179/fig-4

The distributions of structural neighbor counts (Fig. 5) suggest that for some well populated clusters, such as H1-13-1, H2-9-1 and L3-9-cis7-1, CDRs with fewer neighbors in the same cluster are more likely to be misclassified. Taken together, these data indicate that query CDRs that are located centrally within their cluster—those having a small dihedral distance from the cluster exemplar and many neighbors—are more accurately classified.

GBM improves cluster identification accuracy over blindBLAST

Compared to blindBLAST, GBM models improve average query cluster identification accuracy from 79.0% ± 0.23% to 83.4% ± 0.11% (Fig. 6). The difference between GBM

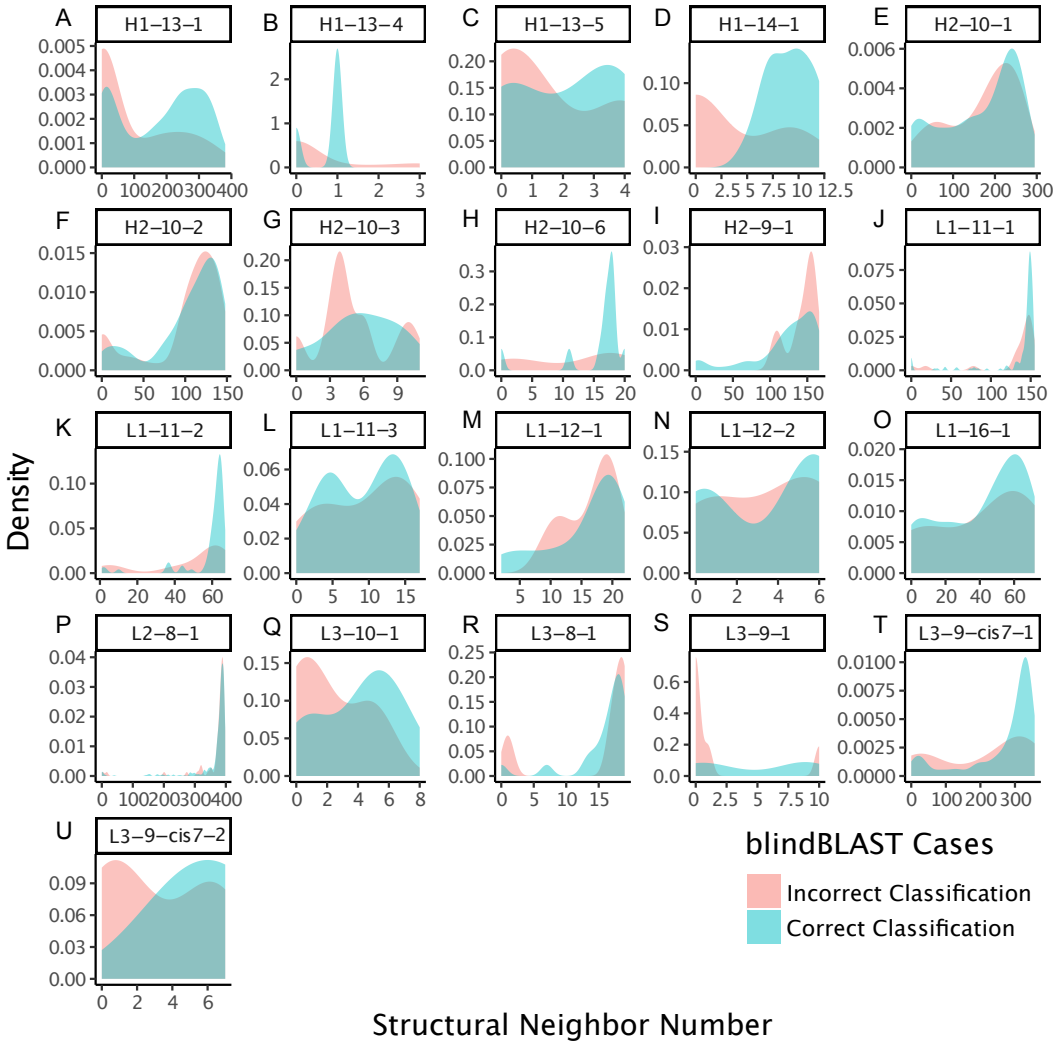

Figure 5 CDRs that are misclassified have fewer structural neighbors, in some clusters. The density of the number of structural neighbors for loop lengths and types with more than five members, both correctly and incorrectly classified. Structural neighbors, as defined in the Methods, are all CDRs with dihedral angle distances equal to or less than 1/15th of the cluster radius to a given CDR. The 1/15th is chosen because it has the best inference. In many clusters, including H1-13-1, H1-13-4, H1-13-5, H1-14-1, H2-10-6, L1-12-1, L3-10-1, L3-9-1, L3-9-cis7-1, L3-9-cis7-2, the misclassified CDRs have greater density at lower numbers of structural neighbors, with respect to the correctly classified CDRs. These data suggest that the number of structural neighbors may affect the chance of correct template selection for a query structure.

Full-size DOI: 10.7717/peerj.6179/fig-5

and blindBLAST accuracy is greater than the standard deviation calculated across each repeat of the three repeats in the 10-fold cross-validation scheme. Also, the GBM model accuracy variance arises from the sparsity of data as evidenced by the larger variance in the loops with sparser members (Fig. S3).

Next, to determine where the improvement in GBM accuracy is achieved, we decomposed the overall error count into changes in individual misclassification counts

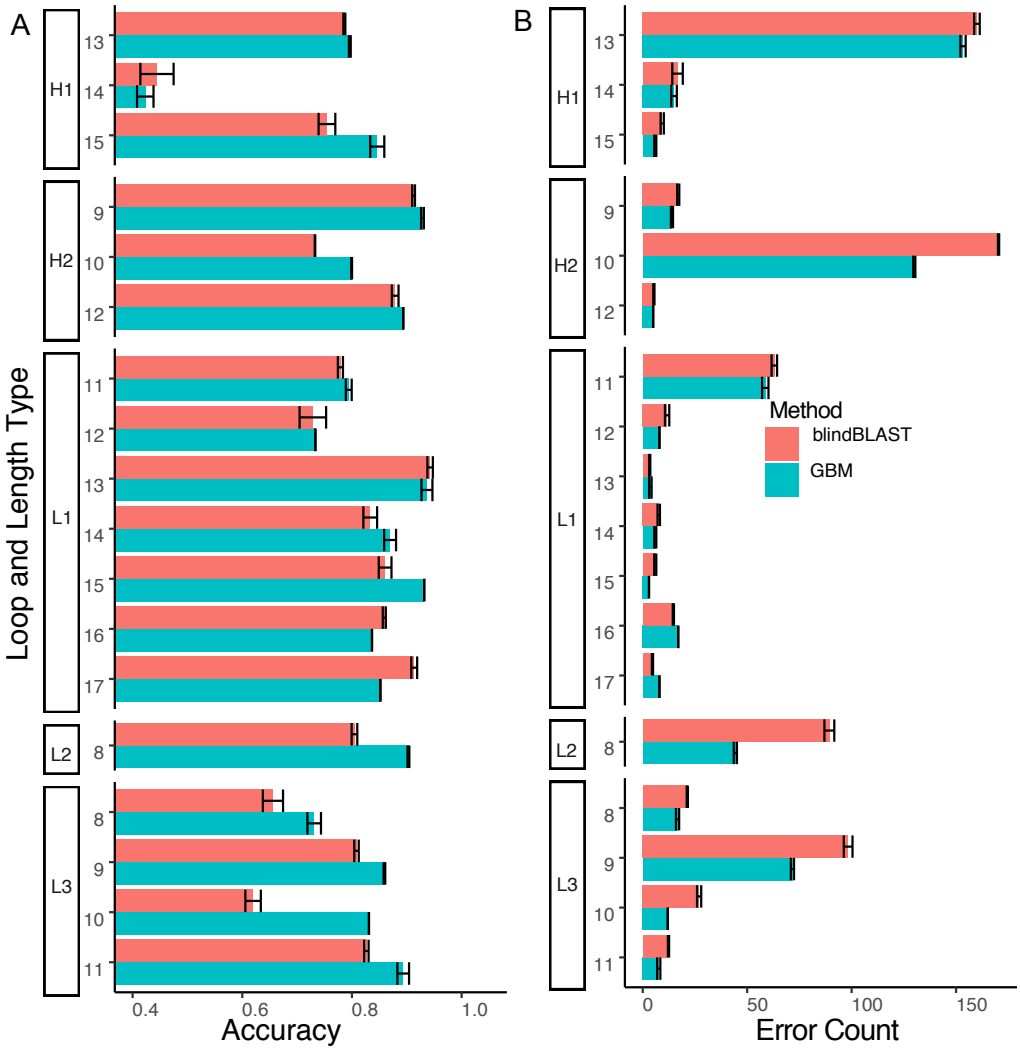

Figure 6 The GBM model has higher accuracy and lower error count than blindBLAST. (A) Comparison of blindBLAST (red) and GBM (blue) accuracy in assigning CDR sequences to clusters. Error counts and accuracies are averaged over each of 3 repeats (in the 3-repeat 10-fold CV scheme) for both GBM and blindBLAST (see Methods). (B) Comparison of the number of erroneously assigned clusters in blindBLAST and GBM error count. The GBM model universally lowers error count.

Full-size DOI: 10.7717/peerj.6179/fig-6

(Fig. 7) and compared potential sequence rules to key features extracted from the GBM models (Figs. 8 and 9).

Decomposing the error counts into their constituent misclassifications provided a few insights into how GBM models outperform the blindBLAST method. For H2-10 loops, the GBM model improved H2-10-2 → H2-10-X misclassifications over blindBLAST. For example the H2-10-2 → H2-10-1 error count was reduced from 14 to 8 (Fig. 7A). Correspondingly, H2-10-X → H2-10-2 misclassifications increased. For example the H2-10-6 → H2-10-2 error count increased from 4 to 7. Taken together, these observations indicate that blindBLAST was failing to properly classify CDR loops as H2-10-2. Similarly,

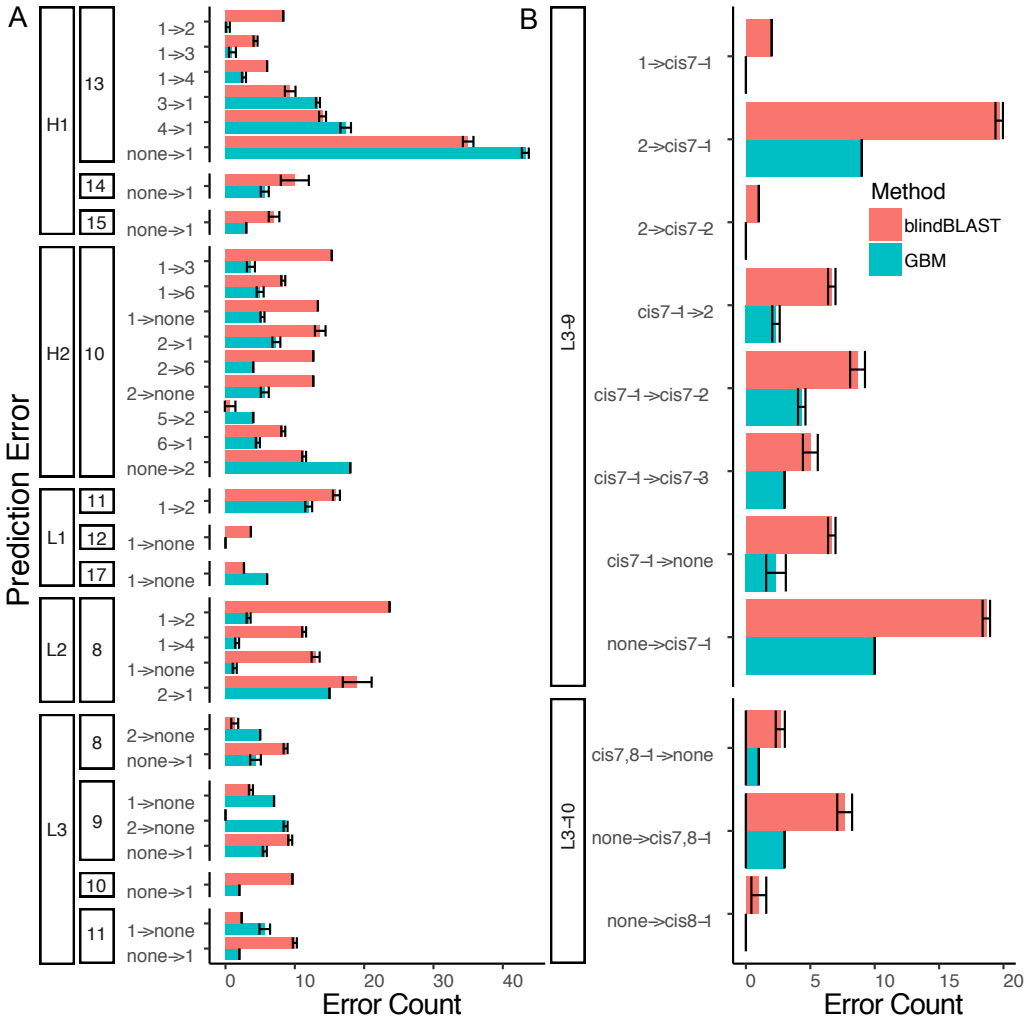

Figure 7 A detailed dissection of error count reduction by the GBM model. (A) All misclassifications with a difference of three or more in error count between GBM and blindBLAST are plotted. A misclassification labeled as 1 →2 denotes queries belonging to cluster 1 that are incorrectly classified as cluster 2. (B) Misclassifications involving at least one cis cluster with corresponding blindBLAST and GBM error counts.

Full-size DOI: 10.7717/peerj.6179/fig-7

we examined L1-11 loops, which are similar to H2-10 in that the second cluster is well populated (Fig. 1). Yet, L1-11 loop classification improvements came from fewer L1-11-1 loops being misclassified as L1-11-2, rather than fewer L1-11-2 loops being misclassified as L1-11-X. This case improves less than H2-10, likely because the L1-11-1 and L1-11-2 clusters are already similar, involving just a flip of two residues, while the other dihedrals have conserved structure. Finally, we investigated improvements for L3-9 and L3-10 loops, both loops that occasionally contain *cis* peptide bonds (Fig. 7B). For both loops, we observed that the most drastic improvements came from cases where blindBLAST was incorrectly assigning loops, some without prolines, to *cis* clusters; To BLAST, the penalty of misaligning a proline at the *cis* position is more or less equal to the penalty of misaligning

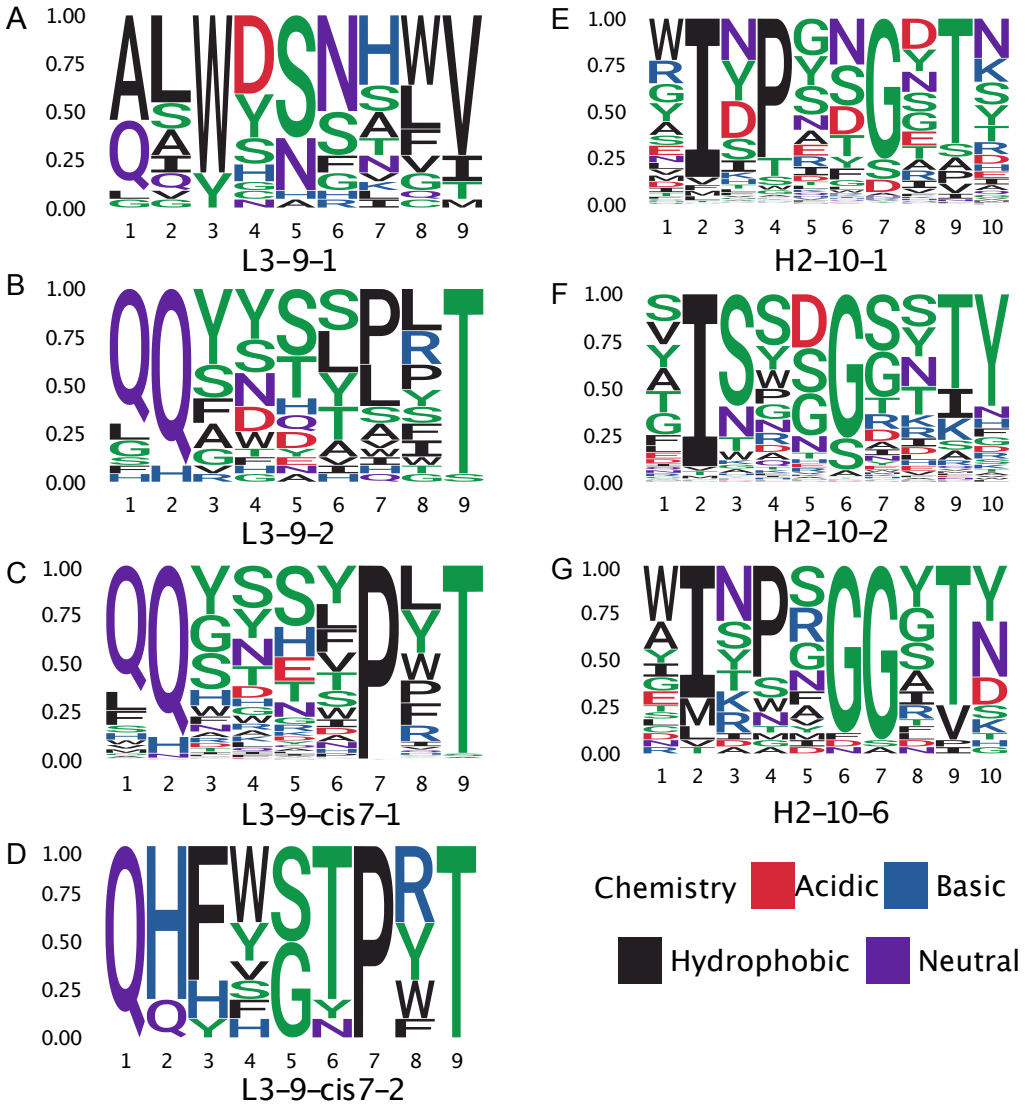

Figure 8 **Sequence logos of selected CDR clusters show there are no readily available sequence rules.** (A–D) The amino acid compositions of the L3-9 clusters. There are no universal distinguishing residues, except for L3-9-1 which does not contain a proline at position 7. (E–G) Similarly, for H2-10-1, -2, and -6 there is not a universal difference in sequences.

Full-size DOI: 10.7717/peerj.6179/fig-8

any other position, but such parity is not required by a GBM model, which can assign greater importance to having a proline in the *cis* position. Other than promoting the importance of the proline residue at a *cis* position using machine learning, another approach is to filter out any template from a cis-cluster. The GBM method beats the filtering method in identifying cis-related clusters in several situations. It reduced prediction errors involving query sequences with cis-proline residue and a non-proline bearing template candidate. It also reduced errors in which query sequences from one *cis*-proline cluster incorrectly identified as some other *cis*-proline cluster (cis7-1 →cis7-2). In some other cases, reduced

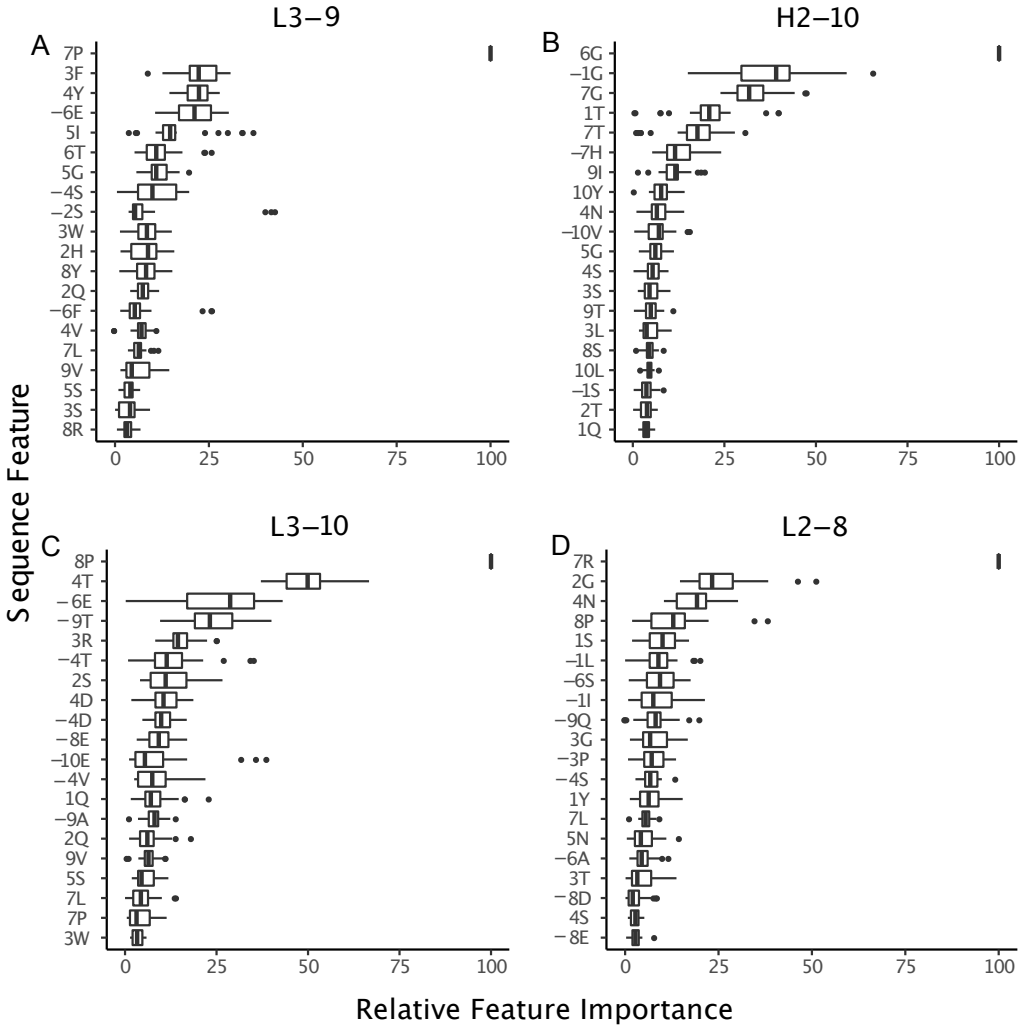

Figure 9 Relative importance can be extracted from GBM models, permitting the identification of key sequence features. For the best GBM model of each loop and length type (though only L3-9 (A), H2-10 (B), L3-10 (C) and L2-8 (D) are shown), the training features were ranked by how much they can help to reduce the training error on a scale of least to most important (1–100). The features are named by the position and the residue type. For example, position "1" is the start of the CDR loop and "−1" is the preceding residue of the loop. Data are from 3-repeat-10-fold splitting between the training/validation and the testing set. As expected for L3-9 and L3-10, two proline containing loops, the presence of a proline at key positions is the most important feature. For H2-10 and L2-8, 6G and 7R are identified as the most important features, respectively. The 6 , 7 and −1 position Gly are the most important features for H2-10 models. In H2-10-6, GBM captured the conserved glycines at positions 6 and 7, which enable the segment to adopt the E region conformation in Ramachandran map. Capturing this feature reduced the error count related to H2-10-6 classification (Fig. 8).

Full-size DOI: 10.7717/peerj.6179/fig-9

errors are those which cis-proline loops are predicted into clusters with non-cis-proline (cis7-1 →2 with nine cases in L3-9-2 having 7th-non-cis-proline, cis7-1 →. None with ten 7th-position-non-cis-proline cases in L3-9-none) which the incorrectly predicted cases can't be excluded by proline filtering in the 7th position.

The proline observation raised the question: were there any other sequence features missed by BLAST, but identified by GBM models? To this end, we constructed sequence logos (*Crooks et al., 2004*) of select CDR loop clusters (Fig. 8) and compared them to residue features deemed important in our GBM models (Fig. 9). For L3-9, we observed proline at position 7 in the loop to be key in both *cis* clusters and the most important feature for our GBM model. The GBM model additionally identified a key threonine residue at position 6, which is never present in L3-9-1. Results are similar for L3-10, with proline residues at *cis* positions being by far the most important. For H2-10, the most important feature of glycine at the 6 position, and the second and third important features are glycine at the -1 and 7 positions. The accuracy improvements arise not from a single residue presence or absence but from the combination of many features with varying weights in the training process.

Comparison to other methods

Several other groups have attempted to predict CDR canonical cluster membership. Two other comparable methods are PIGS and SCALOP. SCALOP (*Wong et al., 2018*) uses PSSMs derived for the length-independent clusters identified first by *Nowak et al. (2016)*. They then assign membership to the highest scoring cluster based on the PSSMs, except for the L2 CDR which is always assigned to the most populous cluster. On the other hand, PIGS (*Chailyan et al., 2011*; *Marcatili et al., 2014*) classifies based on residue identities at key positions of γ light chain CDR clusters and heavy chain clusters curated from previous literature, as well as using clusters found by agglomerative clustering on λ light chain CDR loops using TM-score distance as distance function. In addition to using different cluster definitions, both methods self-report measures of accuracy that differ from our own, confounding direct comparisons with our work.

SCALOP reports precision or the number of predictions identifying a cluster with a loop within 1.5 Å backbone RMSD to the target, dived by the total number of predictions. They report precisions of 89.26% for the CDR H1 loop, 93.60% (H2), 95.67% (L1), 99.13% (L2), and 93.31% (L3). We can compute a similar measure, albeit for different clusters: the number of correct cluster predictions divided by the number of total cluster predictions. The GBM achieves precisions of 85.65% (H1), 88.13% (H2), 87.67% (L1), 93.15% (L2), 87.94% (L3) without accounting for the "none" clusters, same as how SCALOP reports the precisions. These values are unsurprisingly lower as we attempt to classify into greater number of possible clusters.

In their recent update to the webserver PIGSPro (*Lepore et al., 2017*), Lepore et al. report only the average Cα RMSD of all loops (including CDR H3) following alignment of framework regions. They report an average value of 1.79 ± 1.03 Å. Our GBM achieves 1.03 ± 1.06 Å in a comparison of the CDRs, excluding the H3, identified by our GBM-improved BLAST. As expected, our reported average RMSD is lower because we are not considering the difficult-to-model CDR-H3 loop.

DISCUSSION

In the current implementation of RosettaAntibody, blindBLAST is used to select templates for the non-H3 CDR loops, so we are interested in investigating and improving cases where blindBLAST identifies templates with high RMSD. We found that one way to improve template RMSD is to take advantage of the fact that non-H3 CDR loops cluster and to search for templates within the query cluster. BlindBLAST does not consider cluster information explicitly, instead it selects templates of the same length and loop type based on sequence similarity alone. When we tested the ability of blindBLAST to identify clusters compared to a random model, we found multiple and diverse sources of error in cluster classification, so we turned to machine learning, and in particular a GBM model, to improve classification accuracy.

When comparing blindBLAST to a random model, we observed various potential sources, but found no single cause, for inaccurate cluster assignment. First, some loops with few structures, such as H1-14, H1-15, L3-8 were more difficult to classify than loops with many structures because many of the associated misclassifications from the blindBLAST result have random-like error counts (Table S3). Second, other loops with many unbalanced clusters (i.e., where most loops belong to one cluster and few loops belong to the remaining clusters), such as H1-13, resulted in low accuracy. For these loops, identifying a low-RMSD template is confounded by the sparsity of potential templates in the cognate cluster and the large number of potential templates in the other clusters (Fig. S1). Assuming a high sequence similarity across all loops of the same length and type, it is likely that the highest bitscore will arise from an alignment to the most popular cluster. Indeed, we observe misclassifications from sparsely populated clusters to the most populated cluster frequently (6/15 times, Table S4) when blindBLAST performs worse than random. Third, low accuracy was observed when there were two clusters with approximately equal membership. H2-10 loops, which have two highly populated clusters (1 and 2), are such an example and account for 2 of the 15 misclassifications when blindBLAST performs worse than random. Additionally, blindBLAST misclassifies loops where clusters have small dihedral-angle distance between exemplars, such as between L2-8-1 and L2-8-2 (4.5) and between L2-8-1 and L2-8-4 (8.8). Furthermore, in many loop and length types, queries lying at a greater dihedral-angle distance to the cluster exemplar and with a smaller number of structural neighbors were found to have a greater chance to be misclassified.

With no single clear source for blindBLAST misclassifications, we turned to GBM models to improve classification accuracy. As shown in Fig. 7, GBM can better distinguish some cluster pairs with even relatively small amounts of structural data. For example, misclassifications from L3-9-2 to L3-9-cis7-1, from H2-10-1 to H2-10-6, and from L2-8-1 to L2-8-4 have reduced error counts despite small dihedral-angle distances between their cluster exemplars (6.9, 6.8 and 8.8, respectively). However, better performance was not observable for misclassifications involving clusters L2-8-1 and L2-8-2 with only 4.45 dihedral-angle distance between exemplars.

For clusters with relatively large between-clusters-dihedral-angle-distance, GBM models may still not offer any improvement, such as the misclassification between cluster pairs

H1-13-4 & H1-13-1 or H1-13-6 & H1-13-1 with dihedral-angle distances of 17 and 23 between their exemplars, respectively. Having the lack of improvements for such cases in mind, along with the fact that most misclassifications with reduced error counts with GBM models involve clusters that have relatively abundant sample number, we propose that the abundance of data in the non-dominant clusters of the cluster pairs affects how effectively GBM models can improve the blindBLAST performance.

Overall, our results suggest that relative to blindBLAST, GBM is able to better capture features and assign more sensible feature importance with only limited data. GBM models test results have reduced error count (>3) in nine out of 15 listed blindBLAST "worse than random" misclassifications, in 14 out of 33 listed blindBLAST "random-like" misclassifications, and 12 out of 21 listed blindBLAST "better than random" misclassifications.

CONCLUSIONS

In summary, our study has demonstrated that a CDR template from the corresponding structural cluster generally has lower RMSD than a template from the wrong cluster. We have examined the ability of blindBLAST, which is the method used by RosettaAntibody, to identify non-H3 CDR loop clusters implicitly. We trained a GBM model for each CDR loop and length type, and cumulatively improved the canonical structural cluster identification accuracy from 79.0% (±0.23%) test accuracy using the blindBLAST approach in RosettaAntibody to 83.4% ± 0.11% test accuracy using GBM models. If we remove the query cases from the "none" clusters because predicting a loop correctly as a "none" cluster may not narrow the template candidates, then the test accuracy improves from 84.5% ± 0.24% for blindBLAST to 88.16% ± 0.056% for the GBM. The GBM model reduces error counts in all categories of misclassification we benchmarked for blindBLAST. However, most of the misclassifications with GBM reduced error counts involve clusters with relatively abundant sample sizes, especially the non-dominant clusters. Thus, the bottlenecks to further improvement are primarily the member size imbalance between clusters and data sparsity in clusters. Methods that can generate valid data to enrich clusters with sparse data may improve the estimation accuracy of the GBM model. A set of structures that lie within the cluster radius constraint could be generated using Rosetta, emulating the SMOTE method (*Chawla et al., 2002*) for enriching samples in underpopulated classes. Another approach that serves to increase the member sizes of these clusters is to use semi-unsupervised learning to incorporate the sequenced antibodies without solved structures.

Furthermore, the GBM models are found incapable of further reducing errors in misclassifications involving clusters with small dihedral angle distance such as between L2-8-1 and L2-8-2. To address this limitation, we may wish to reflect the differences of distances between cluster pairs in the loss function in the machine learning training process using generated synthetic data, so that mismatches between clusters of greater structural differences can be penalized more heavily. On the other hand, the sampling and learning process can also be adjusted by training each weak learner with an under-sampled dominant

cluster rather than oversampling the non-dominant clusters used in this study. Finally, instead of ten residues upstream and downstream of the loop proper used in our method, antibody framework residues which are neighboring the CDR loop residues are known to affect loop conformation and could also be included as features (*Ting et al., 2010*).

ACKNOWLEDGEMENTS

We would like to thank Drs. Roland Dunbrack (ORCID 0000-0001-7674-6667) and Jared Adolf-Bryfogle for help with the PyIgClassify database.

ADDITIONAL INFORMATION AND DECLARATIONS

Funding

Xiyao Long, Jeliazko R. Jeliazkov, and Jeffrey J. Gray were funded by NIH R01-GM078221. Jeliazko R. Jeliazkov was additionally funded by NIH F31-GM123616. The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Grant Disclosures

The following grant information was disclosed by the authors:
NIH: R01-GM078221.
NIH: F31-GM123616.

Competing Interests

Jeffrey J. Gray is an unpaid board member of the Rosetta Commons. Under institutional participation agreements between the University of Washington, acting on behalf of the Rosetta Commons, Johns Hopkins University may be entitled to a portion of revenue received on licensing Rosetta software including programs described here. As a member of the Scientific Advisory Board of Cyrus Biotechnology, Jeffrey J. Gray is granted stock options. Cyrus Biotechnology distributes the Rosetta software, which may include methods described in this paper.

Author Contributions

- Xiyao Long conceived and designed the experiments, performed the experiments, analyzed the data, contributed reagents/materials/analysis tools, prepared figures and/or tables, authored or reviewed drafts of the paper, approved the final draft.
- Jeliazko R. Jeliazkov conceived and designed the experiments, analyzed the data, contributed reagents/materials/analysis tools, prepared figures and/or tables, authored or reviewed drafts of the paper, approved the final draft.
- Jeffrey J. Gray conceived and designed the experiments, contributed reagents/materials/analysis tools, authored or reviewed drafts of the paper, approved the final draft.

Data Availability

The following information was supplied regarding data availability:

The code is available on Github in the xlong2/machine-learning-cdr repository: https://github.com/xlong2/machine-learning-cdr.

Supplemental Information

Supplemental information for this article can be found online at http://dx.doi.org/10.7717/peerj.6179#supplemental-information.

REFERENCES

Adolf-Bryfogle J, Kalyuzhniy O, Kubitz M, Weitzner BD, Hu X, Adachi Y, Schief WR, Dunbrack Jr RL. 2018. RosettaAntibodyDesign (RAbD): a general framework for computational antibody design. *PLOS Computational Biology* **14**:e1006112 DOI 10.1371/journal.pcbi.1006112.

Adolf-Bryfogle J, Xu Q, North B, Lehmann A, Dunbrack RL. 2015. PyIgClassify: a database of antibody CDR structural classifications. *Nucleic Acids Research* **43**:D432–D438 DOI 10.1093/nar/gku1106.

Almagro JC, Teplyakov A, Luo J, Sweet RW, Kodangattil S, Hernandez-Guzman F, Gilliland GL. 2014. Second antibody modeling assessment (AMA-II). *Proteins: Structure, Function, and Bioinformatics* **82**:1553–1562 DOI 10.1002/prot.24567.

Baran D, Pszolla MG, Lapidoth GD, Norn C, Dym O, Unger T, Albeck S, Tyka MD, Sarel J. 2017. Principles for computational design of binding antibodies. *Proceedings of the National Academy of Sciences of the United States of America* **114**:10900–10905 DOI 10.1073/pnas.1707171114.

Beck JE, Woolf BP. 2000. *High-level student modeling with machine learning.* Heidelberg: Springer, Berlin, 584–593 DOI 10.1007/3-540-45108-0_62.

Blagus R, Lusa L. 2013. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics* **14**:106 DOI 10.1186/1471-2105-14-106.

Boyd SD, Crowe JE. 2016. Deep sequencing and human antibody repertoire analysis Sequence analysis techniques for antibody variable genes *Current Opinion in Immunology* **40**:103–109 DOI 10.1016/j.coi.2016.03.008.

Chailyan A, Marcatili P, Cirillo D, Tramontano A. 2011. Structural repertoire of immunoglobulin λ light chains. *Proteins* **79(5)**:1513–1524 DOI 10.1002/prot.22979.

Chailyan A, Marcatili P, Tramontano A. 2011. The association of heavy and light chain variable domains in antibodies: implications for antigen specificity. *FEBS Journal* **278**:2858–2866 DOI 10.1111/j.1742-4658.2011.08207.x.

Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* **16**:321–357 DOI 10.1613/jair.953.

Choi Y, Deane CM. 2011. Predicting antibody complementarity determining region structures without classification. *Molecular BioSystems* **7**:3327–3334 DOI 10.1039/c1mb05223c.

Chothia C, Lesk AM, Tramontano A, Levitt M, Smith-Gill SJ, Air G, Sheriff S, Padlan EA, Davies D, Tulip WR, Colman PM, Spinelli S, Alzari PM, Poljak RJ. 1989. Conformations of immunoglobulin hypervariable regions. *Nature* **342**:877–883 DOI 10.1038/342877a0.

Corrales M, Cuscó P, Usmanova DR, Chen H-C, Bogatyreva NS, Filion GJ, Ivankov DN. 2015. Machine learning: how much does it tell about protein folding rates? *PLOS ONE* **10**:e0143166 DOI 10.1371/journal.pone.0143166.

Crooks GE, Hon G, Chandonia J-M, Brenner SE. 2004. WebLogo: a sequence logo generator. *Genome Research* **14**:1188–1190 DOI 10.1101/gr.849004.

DeKosky BJ, Lungu OI, Park D, Johnson EL, Charab W, Chrysostomou C, Kuroda D, Ellington AD, Ippolito GC, Gray JJ, Georgiou G. 2016. Large-scale sequence and structural comparisons of human naive and antigen-experienced antibody repertoires. *Proceedings of the National Academy of Sciences of the United States of America* **113E**:2636–2645 DOI 10.1073/pnas.1525510113.

Dittman DJ, Khoshgoftaar TM, Napolitano A. 2015. The effect of data sampling when using random forest on imbalanced bioinformatics data. In: *2015 IEEE international conference on information reuse and integration*. Piscataway: IEEE, 457–463 DOI 10.1109/IRI.2015.76.

Dunbar J, Krawczyk K, Leem J, Baker T, Fuchs A, Georges G, Shi J, Deane CM. 2014. SAbDab: the structural antibody database. *Nucleic Acids Research* **42**:D1140–D1146 DOI 10.1093/nar/gkt1043.

Dunbar J, Krawczyk K, Leem J, Marks C, Nowak J, Regep C, Georges G, Kelm S, Popovic B, Deane CM. 2016. SAbPred: a structure-based antibody prediction server. *Nucleic Acids Research* **44**:W474–W478 DOI 10.1093/nar/gkw361.

Georgiou G, Ippolito GC, Beausang J, Busse CE, Wardemann H, Quake SR. 2014. The promise and challenge of high-throughput sequencing of the antibody repertoire. *Nature Biotechnology* **32**:158–168 DOI 10.1038/nbt.2782.

Hou D, Chen C, Seely EJ, Chen S, Song Y. 2016. High-throughput sequencing-based immune repertoire study during infectious disease. *Frontiers in Immunology* **7**:1–11 DOI 10.3389/fimmu.2016.00336.

Jain P, Garibaldi JM, Hirst JD. 2009. Supervised machine learning algorithms for protein structure classification. *Computational Biology and Chemistry* **33**:216–223 DOI 10.1016/j.compbiolchem.2009.04.004.

Kilambi KP, Gray JJ. 2017. Structure-based cross-docking analysis of antibody-antigen interactions. *Scientific Reports* **7**:1–15 DOI 10.1038/s41598-017-08414-y.

Klausen MS, Anderson MV, Jespersen MC, Nielsen M, Marcatili P. 2015. LYRA, a webserver for lymphocyte receptor structural modeling. *Nucleic Acids Research* **43(W1)**W349–W355 DOI 10.1093/nar/gkv535.

Kohavi R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Proceedings of the 14th international joint conference on artificial intelligence, Vol. 2*. New York: ACM, 1137–1145.

Koivuniemi A, Takkinen K, Nevanen T. 2017. A computational approach for studying antibody-antigen interactions without prior structural information: the anti-testosterone binding antibody as a case study. *Proteins: Structure, Function, and Bioinformatics* **85**:322–331 DOI 10.1002/prot.25226.

Kuhn M. 2008. Building predictive models in R using the caret package. *Journal of Statistical Software* **28**:1–26 DOI 10.18637/jss.v028.i05.

Kuhn M, Johnson K. 2013a. *Applied predictive modeling*. New York: Springer DOI 10.1007/978-1-4614-6849-3.

Kuhn M, Johnson K. 2013b. Classification trees and rule-based models. In: *Applied predictive modeling*. New York: Springer, 369–413 DOI 10.1007/978-1-4614-6849-3_14.

Kumar C, Choudhary A. 2012. A top-down approach to classify enzyme functional classes and sub-classes using random forest. *EURASIP Journal on Bioinformatics & Systems Biology* **2012**:1 DOI 10.1186/1687-4153-2012-1.

Kuroda D, Shirai H, Jacobson MP, Nakamura H. 2012. Computer-aided antibody design. *Protein Engineering, Design & Selection* **25**:507–521 DOI 10.1093/protein/gzs024.

Lepore R, Olimpieri PP, Messih MA, Tramontano A. 2017. PIGSPro: prediction of immunoGlobulin structures v2. *Nucleic Acids Research* **45(W1)**:W17–W23 DOI 10.1093/nar/gkx334.

Lippow SM, Wittrup KD, Tidor B. 2007. Computational design of antibody-affinity improvement beyond *in vivo* maturation. *Nature Biotechnology* **25**:1171–1176 DOI 10.1038/nbt1336.

Louppe G, Wehenkel L, Sutera A, Geurts P. 2013. Understanding variable importances in forests of randomized trees. 431–439.

Luciani F. 2016. High-throughput sequencing and vaccine design. *Revue scientifique et technique (International Office of Epizootics)* **35**:53–65 DOI 10.20506/rst.35.1.2417.

Marcatili P, Ghiotto F, Tenca C, Chailyan A, Mazzarello AN, Yan XJ, Colombo M, Albesiano E, Bagnara D, Cutrona G, Morabito F, Bruno S, Ferrarini M, Chiorazzi N, Tramontano A, Fais F. 2013. Igs expressed by chronic lymphocytic Leukemia B cells show limited binding-site structure variability. *The Journal of Immunology* **190**:5771–5778 DOI 10.4049/jimmunol.1300321.

Marcatili P, Olimpieri PP, Chailyan A, Tramontano A. 2014. Antibody structural modeling with prediction of immunoglobulin structure (PIGS) web server. *Nature Protocols* **9**:2771–2783 DOI 10.1038/nprot.2014.189.

Messih MA, Lepore R, Marcatili P, Tramontano A. 2014. Improving the accuracy of the structure prediction of the third hypervariable loop of the heavy chains of antibodies. *Bioinformatics* **30**:2733–2740 DOI 10.1093/bioinformatics/btu194.

North B, Lehmann A, Dunbrack RL. 2011. A new clustering of antibody CDR loop conformations. *Journal of Molecular Biology* **406**:228–256 DOI 10.1016/j.jmb.2010.10.030.

Nowak J, Baker T, Georges G, Kelm S, Klostermann S, Shi J, Sridharan S, Deane CM. 2016. Length-independent structural similarities enrich the antibody CDR canonical class model. *MAbs* **8(4)**:751–760 DOI 10.1080/19420862.2016.1158370.

Radivojac P, Clark WT, Oron TR, Schnoes AM, Wittkop T, Sokolov A, Graim K, Funk C, Verspoor K, Ben-Hur A, Pandey G, Yunes JM, Talwalkar AS, Repo S, Souza ML, Piovesan D, Casadio R, Wang Z, Cheng J, Fang H, Gough J, Koskinen P, Törönen P, Nokso-Koivisto J, Holm L, Cozzetto D, Buchan DW, Bryson K, Jones DT, Limaye B, Inamdar H, Datta A, Manjari SK, Joshi R, Chitale M, Kihara D, Lisewski AM, Erdin S, Venner E, Lichtarge O, Rentzsch R, Yang H, Romero AE, Bhat P, Paccanaro A, Hamp T, Kaßner R, Seemayer S, Vicedo E, Schaefer C, Achten D, Auer F, Boehm A, Braun T, Hecht M, Heron M, Hönigschmid P, Hopf TA,

Kaufmann S, Kiening M, Krompass D, Landerer C, Mahlich Y, Roos M, Björne J, Salakoski T, Wong A, Shatkay H, Gatzmann F, Sommer I, Wass MN, Sternberg MJ, Škunca N, Supek F, Bošnjak M, Panov P, Džeroski S, Šmuc T, Kourmpetis YA, van Dijk AD, ter Braak CJ, Zhou Y, Gong Q, Dong X, Tian W, Falda M, Fontana P, Lavezzo E, Di Camillo B, Toppo S, Lan L, Djuric N, Guo Y, Vucetic S, Bairoch A, Linial M, Babbitt PC, Brenner SE, Orengo C, Rost B, Mooney SD, Friedberg I. 2013. A large-scale evaluation of computational protein function prediction. *Nature Methods* **10**:221–227 DOI 10.1038/nmeth.2340.

Schroeder HW, Cavacini L, Cavacini L. 2010. Structure and function of immunoglobulins. *The Journal of Allergy and Clinical Immunology* **125**:S41–S52 DOI 10.1016/j.jaci.2009.09.046.

Shirai H, Ikeda K, Yamashita K, Tsuchiya Y, Sarmiento J, Liang S, Morokata T, Mizuguchi K, Higo J, Standley DM, Nakamura H. 2014. High-resolution modeling of antibody structures by a combination of bioinformatics, expert knowledge, and molecular simulations. *Proteins: Structure, Function and Bioinformatics* **82**:1624–1635 DOI 10.1002/prot.24591.

Si J, Zhao R, Wu R. 2015. An overview of the prediction of protein DNA-binding sites. *International Journal of Molecular Sciences* **16**:5194–5215 DOI 10.3390/ijms16035194.

Sun Y, Kamel MS, Wang Y. 2006. Boosting for learning multiple classes with imbalanced class distribution. In: *Proceedings—IEEE international conference on data mining, ICDM*. Piscataway: IEEE, 592–602 DOI 10.1109/ICDM.2006.29.

Sun Y, Kamel MS, Wong AKC, Wang Y. 2007. Cost-sensitive boosting for classification of imbalanced data. *Pattern Recognition* **40**:3358–3378 DOI 10.1016/J.PATCOG.2007.04.009.

Ting D, Wang G, Shapovalov M, Mitra R, Jordan MI, Dunbrack RL. 2010. Neighbor-dependent Ramachandran probability distributions of amino acids developed from a hierarchical Dirichlet process model. *PLOS Computational Biology* **6**(**4**):e1000763 DOI 10.1371/journal.pcbi.1000763.

Weitzner BD, Jeliazkov JR, Lyskov S, Marze N, Kuroda D, Frick R, Adolf-Bryfogle J, Biswas N, Dunbrack Jr RL, Gray JJ. 2017. Modeling and docking of antibody structures with Rosetta. *Nature Protocols* **12**:401–416 DOI 10.1038/nprot.2016.180.

Weitzner BD, Kuroda D, Marze N, Xu J, Gray JJ. 2014. Blind prediction performance of RosettaAntibody 3.0: grafting, relaxation, kinematic loop modeling, and full CDR optimization. *Proteins: Structure, Function and Bioinformatics* **82**:1611–1623 DOI 10.1002/prot.24534.

Wong WK, Georges G, Ros F, Kelm S, Lewis AP, Taddese B, Leem J, Deane CM. 2018. SCALOP: sequence-based antibody canonical loop structure annotation. *Bioinformatics* Epub ahead of print 15 October 2018 DOI 10.1093/bioinformatics/bty877.

Yamashita K, Ikeda K, Amada K, Liang S, Tsuchiya Y, Nakamura H, Shirai H, Standley DM. 2014. Kotai antibody builder: automated high-resolution structural modeling of antibodies. *Bioinformatics* **30**:3279–3280 DOI 10.1093/bioinformatics/btu510.