Improving Antibody CDR Template Selection by Structural Cluster Prediction

by
Xiyao Long

A thesis submitted to the Johns Hopkins University in conformity with the requirements of the degree of Master of Science in Engineering

Johns Hopkins University
Baltimore, Maryland
Nov, 2017

Abstract

With the advent of high-throughput sequencing, antibody sequences can be acquired at much greater speed than corresponding structures, creating a need for rapid structure determination. Computational modeling is the only feasible method for high-throughput structure determination, however it does not always produce models with high accuracy. In antibody modeling, the framework regions are well conserved and readily modeled to sub-Angstrom accuracy, but accurate modeling of the complementarity determining region (CDR) loops remains elusive. This is a challenge we must overcome if we are to study antibody function or design an antibody, using models. Of the six CDR loops, the non-H3 CDR loops (H1, H2, and L1–L3) are easier to model than the H3 loop, because they are shorter and have less structural and length variability. Moreover, most of the non-H3 CDR loop structures can be grouped by CDR and length and can be clustered into a few canonical structure clusters. The ability to accurately predict the correct cluster of a CDR from sequence alone could improve structural modeling. In this thesis, I assessed how well current modeling techniques can identify the CDR canonical structures from sequence alone and I improved the retrieval accuracy. First, I benchmarked the current CDR loop modeling method in Rosetta and found it failed to predict the correct canonical structure clusters for 19% of CDRs. Next, I assessed the significance of the failures by comparing to a random cluster selection model. Then, to improve the accuracy of template selection, I trained a machine learning classifier, for each CDR and length group, with sequences as features, and found that the classifier successfully improved the retrieval of canonical structures. This improvement is not achievable by the residue position rules alone. Finally, I propose incorporating canonical class prediction via machine learning to improve canonical structure retrieval accuracy and I expected this improvement to increase as the less populated CDR clusters become more enriched.

Acknowledgement

I would like to first thank Prof. Jeffrey Gray for giving me the opportunity to join this excellent group for the special opportunity to work with protein structure predictions. His patience, encouragements has given me space for exploring solutions to my scientific question while also guide me to progress my work with focus and to present my thoughts and results with greater clarity.

I want to express thanks to Prof. Rebecca Schulman for being my thesis reader. I enjoyed working on my first course project about antibodies with her guidance. I'm thankful to Artun Hoscan from Schulman lab to read my thesis and give good advices.

I would also like to thank Jeliazko Jeliazkov for being my TA for the computational protein modeling course and graduate student mentor for my thesis. He is always a good problem solver and can guide me through solving any specific problem I encounter in the lab, be it debugging a script, finding methods to test, effectively organizing my data, writing with more clarity or fixing a printer.

I own a lot of thanks to other group members. Shourya Sonkar Roy Burman is very knowledgeable in protein modeling and guide me on finishing a Capri modeling and docking Challenge. Same, Dr. Nick Marze and Dr. Jason Labonte can always suggest me with certain scientific databases or specific Rosetta coding solutions from their knowledge and expertise developed over the years. Rebecca Alford is always responsive, she helped me revise my PhD application materials and give me good suggestions on my thesis work. Sergey Lyskov has handed me a Rosetta coding project and let me both practice my coding skills and contribute to Rosetta.

Special thanks also direct to Joseph Lubin and Naireeta Biswas, for being my fellow master students in the lab and encourage each other on pursuing the degree. Joseph and I also

worked on a protein modeling course project together and he along with Zuo Xiaotong, Narieeta and Jeliazko inspired me to join the lab.

Other thanks go to Dr. Michael Pacella, Morgan Nance, Summer student Sophia, Paige Stanley, PMB rotation student Jacob, Kathy Wang, Kayvon Tabrizi for being my lab mate.

Outside the lab, I would like to especially thank Jared Adolf-Bryfogle originally from the lab of Dr.Benjamin North. He has answered a lot of my questions related to his PyIgClassify database and his work is the foundation to my method presented in this thesis. Dr. Brian Weitzner, Dr. Daisuke Kuroda and Dr. Nick Marze as previous lab members they have worked extensively on antibody modeling and established the current status of antibody modeling in the Gray lab, which inspires Jeliazko and me to work on the problem addressed by this thesis. I also want to thank Rohit Bhattacharya from Dr. Rachel Karchin's lab for helping me revise the proposal summarizing this thesis work for submission to the Women in Machine Learning conference.

Table of Contents

List of Tables

List of Figures

Chapter I: Introduction

I. Antibody CDRs

The adaptive immune system present in all jawed vertebrates[1] can respond to a myriad of pathogens. Adaptability stems from the maturing process of immunoglobulin producing lymphocytes (B cells) in which the immunoglobulin encoding genes undergo V(D)J recombination and somatic hyper-mutation (SHM) to produce an enormous variety of unique antibody sequences (~10^13)[2]. This process generates immunoglobulins, or antibodies, comprised of two paired light and heavy chains, where the light chain has one variable and one constant domain and the heavy chain has one variable and three constant domains (in the IgG isotype) shown in Figure 1-1. Each immunoglobulin domain consists of a beta sandwich, with the variable domains each having three loops, known as complementarity determining regions (CDRs), important for antigen binding and the framework (FW) region that embeds the three loops. The CDRs are often annotated as L1, L2, and L3 for the light chain and H1, H2, and H3 for the heavy chain shown in Figure 1-2. CDRs have high sequence and structure variability, just as expected by the functional requirement to bind to many possible antigens[2,3].

Antibodies have emerged as important therapeutic molecules[4] and research tools[5,6] because their ability to bind any one of a diverse set of molecules. Their biomedical importance and utility has led to the arduous study of their structure and function[7] [8], and to antibody design projects[4] to develop antibodies capable of binding various new pathogens or cell markers, or with improved affinity. However, both antibody structure determination and design are not always easily carried out experimentally. Protein X-ray crystallography is generally time and labor consuming and does not guarantee a solved structure. Experimental protein design approaches such as phage display can have problems in finding the most fit variant because of the nonlinear relationship between the number of mutations and protein fitness and the limited exploration of sequence space due to library size[9].

Computational modeling and design of antibodies can overcome the time and cost barrier present in experiments and provide value information[10,11]. For example, high-throughput modeling of antibody structures has been shown to add prognostic value of sequence data alone in chronic lymphocytic leukemia[12]. Beyond modeling, docking studies of antibodies complexed with various antigens can reveal atomic details of antibody-antigen interactions[13]. Finally, in antibody design, computational approaches can utilize various sequence space searching protocols to enhance affinity or design an antibody *de novo* (with no prior sequence information) [9–11,14].

One of the most crucial parts of antibody modeling is the modeling of CDRs. The CDR modeling problem is essentially loop modeling within a constrained environment. The constrained environment of each CDR loop consists of its loop stems adjoining to the conserved beta sandwiched framework, and the other CDRs and framework segments aligning in close proximity with the loop. Fortunately, the CDR modeling problem is alleviated in part by the ever expanding library of solved CDRs structures from their antibodies.

Current antibody modeling suites have taken several approaches in modeling the CDR regions, but most utilize homology modeling to choose a template structure followed by grafting onto the modeled antibody framework and "*de novo*" modeling for refining the grafted CDRs. The homology modeling step seeks a structure template, based on the query CDR sequence, most likely to resemble the query's native structure. The antibody modeling tool SAbPred[10] utilizes FREAD[15] to find CDR template structures, with an environmentally constrained substitution matrix[16]. Kotai Antibody Builder[17] and PIGS[12] manually set sequence-based rules for selecting CDR templates. RosettaAntibody[11] takes a minimalistic approach and finds the most similar sequence using BLAST within the CDRs of the same CDR loop and length combination, without sequence rules or specially built substitution matrix.

II. Canonical CDR loops

As antibody variable regions' beta sandwich scaffolds that are structurally similar, the framework (FW) region of the antibody variable domain is conserved[18], with specific amino acid (a.a.)

identities in the same positions across different antibodies[19], whereas the CDRs tend to be variable in sequence and length as a result of V(D)J recombination and SHM. Thus the conserved residues serve as "landmarks" to enable a universal numbering scheme for the variable domain, with numbers repeated or deleted to accommodate variable length CDRs loops. Typically, a numbering scheme assigns identical numbers to structurally equivalent regions. The are many examples of antibody numbering schemes including Kabat[20], Chothia[21], enchanced Chothia (Martin)[22], IMGT[19], and Aho[23]. The Aho scheme is preferred because it can be used to number both antibodies and TCR, and considers the indel positions that may be missed in the Kabat and Chothia numbering schemes. Most importantly, the Aho scheme is used by PyIgClassify to define the CDR L1/H1 as residues 24–42, CDR L2/H2 as residues 57–72/69, and CDRs L3/H3 as residues 107-138 and these CDR definitions are used throughout this thesis.

When non-H3 CDRs of the same length and type (e.g. all L1 length 10 loops) were compared, studies[22,24] found that the many loops occupied just a few structural clusters, referred to as canonical conformations. In particular, recent work done by North et al. used similarity scores derived from a pairwise comparison of backbone torsion angles to generate structural clusters. By defining the canonical structure, or cluster exemplar, as the structural median of all cluster members, this work expanded on earlier work by Chothia and coworkers' manually identified canonical structures. The structural median is a good representative of the cluster because the mean torsion angle difference, per residue, between any cluster member and the exemplar is less than 40 degrees. Members in each cluster therefore deviate from the cluster exemplar by only a small average backbone dihedral angle distance. Therefore, CDRs in the same cluster are among the best structural templates to any CDR in the cluster.

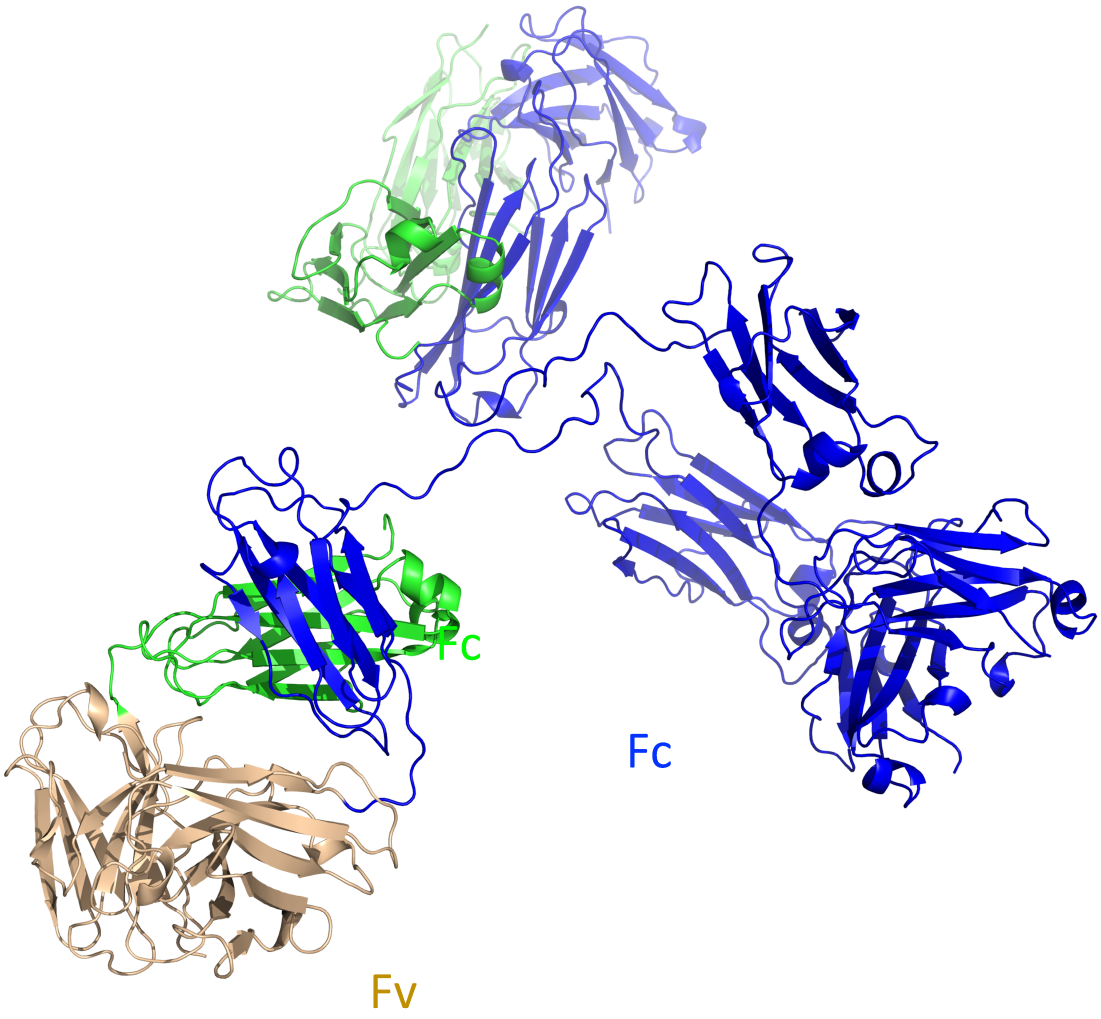


Figure 1-1. Fc and Fv region in heavy and light chain of a typical antibody.

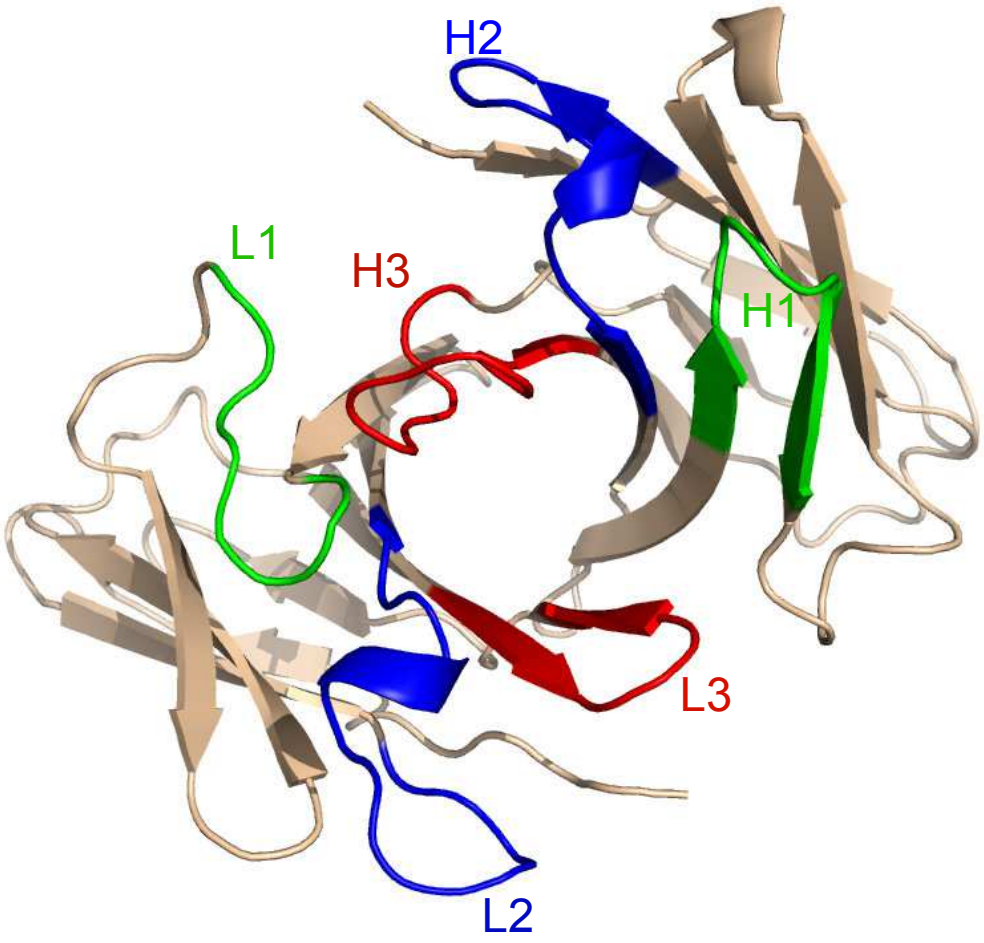


Figure 1-2. CDR loops of a typical antibody variable fragment(Fv).

III. Utilization of canonical clusters in current CDR loop modeling

Currently antibody modeling methods vary in how, or if at all, to incorporate CDR canonical structure information. Of the four major antibody modeling software I examined (RosettaAntibody, PIGS, Kotai Antibody Builder, and SabPred), only PIGS and Kotai Antibody Builder have explicitly set sequence rules to predict the canonical structure of the query CDRs. The sequence rules utilized are manually curated and the identity of residues in specific positions is used to assign a structural cluster. These rules can offer deterministic cluster assignment and also are easy for human interpretation, but they are limited in their adaptability and power because some clusters are devoid of such rules and the

simplicity prevents them from reaching the real margin of clusters boundaries in the sequence space. For instance, sequence rules utilized by PIGS and Kotai Antibody Builder can identify fewer canonical clusters in loop H1-6 and H1-9(labeled for a different CDR definition scheme) than the number identified by North study. Kotai Antibody Builder took one step further in this issue[17] by devising a two-voter method to incorporate all canonical structures candidates in cluster identification other than just which sequence rules can identify. The method select structural clusters passing position specific substitution matrix (PSSM) thresholds and also identified from specific structural cluster by curated key residues sequence. If the sequence is not covered by position specific rules and sequence PSSM scores passing thresholds corresponding to multiple clusters, then the cluster with CDR of the same origin as the selected framework structure template is favored. The Kotai cluster identification method correctly identified cluster in 90% of all queries, yet it is not clear whether the tested data is excluded from constructing the PSSM profiles, therefore the accuracy might be overestimate. RosettaAntibody and SabPred do not utilize canonical structure identification for CDR template searching, but use similarity scores to find CDR templates based on sequence similarity. RosettaAntibody uses BLAST with the PAM30 substitution matrix for similarity scoring, whereas the homology modeling method named Fread in SabPred utilize environment specific substitution matrix for scoring. Additionally, Rosetta has a proline filter that discards CDR templates belonging to any one of the cis clusters if the query CDR does not have proline at the correct position, because cis-trans isomerization is only common for proline.

IV. Comparison of CDR loop modeling accuracy of different methods.

It stands to reason that if CDRs in the same structure cluster assume very similar structure, then the ability of a modeling method to predict cluster membership is a good indicator of its accuracy. However, except for Kotai Antibody Builder, the above methods have not reported accuracies for cluster prediction. Most of the published work has focused on reporting the backbone RMSD between the query and model CDR. For example, a study benchmarked the RosettaAntibody homology modeling method

on 54 antibody targets, with 42/54 (L1), 50/54 (L2), 37/54 (L3), 36/54 (H1), and 42/54(H2) having less than 1 Å RMSD between the homology modeled and actual CDR. As another example, to benchmark PIGS, a set of 689 antibody structures was used with leave one cross validation (LOOCV) to evaluate performance. The results showed only 50% of the modeled non-H3 CDRs have less than 1 Å RMSDs when compared to the actual structures. The smaller percentage of sub-angstrom accuracy models compared to Rosetta is possibly due to the smaller template library of PIGS. Finally, AbodyBuilder(SabPred), which utilizes FREAD for loop modeling, was tested on a set of 54 antibodies. The average query to model RMSD is reported with RosettaAntibody having lower RMSDs in 3 out of 5 non-H3 CDRs compared to the Fread (in Angstrom 1.09,1.00,0.88 versus 0.83, 0.91, 0.83 for loops L1, L3, H1).

As stated above, the benchmarking studies for different modeling methods are done using different sets of antibodies, so evidence suggesting the superiority of one method over another is confounded by the disparity of test set size and template library size between the studies. Most of the performed studies are also limited in the dataset size for evaluating the qualities of selected CDRs structural templates. The evaluation on just a small or incomplete data sets can also overlook some CDR template selection problems that are only significant under the examination of a greater dataset. Moreover, the above comparisons were done using final models having its grafted template backbone structures further sampled to minimize the energy, which are indicative of the CDRs template structure selection quality, but not equivalent, as the energy refined structure can assume a different canonical structure cluster even the initially threaded template structure is in another cluster[25].

V. The significance of proline residues and importance of distinguishing them

The Proline in proteins can assume either cis or trans conformation at its omega dihedral angle identified by the bonds connecting the atoms $C_{\alpha,n-1}$-C_{n-1}-N_{pro}-$C_{\alpha,pro}$. The cis conformational proline is found to be especially conserved evolutionarily and frequently sit near active sites of proteins[26]. The

isomerization between cis and trans can serve a lot of functional roles in biological system, including being the rate limiting step of protein folding, modulating signaling proteins and ion channels between their active and inactive forms[27,28]. The failure of the isomerization regulation is found to induce protein aggregation in neurodegenerative disease or affect phosphorylation signaling which leads to cancers. It is also suggested that the trans conformation is more flexible than cis conformation in the loop context.

Antibody CDRs have certain residue positions with conserved proline, a portion of these Prolines being cis and the other portion being trans. For example, the 7th position Prolines in L3-9 are predominantly cis, and this portion can be further divided into 3 canonical structural clusters as cis7-1,2 and 3 in the North clustering study. The other much smaller portion (12/445) with trans proline at 7^{th} position from the solved structures exist only in cluster L3-9-2.

Although the North clustering study has assigned a unique trans or cis conformation to each of the unique CDR sequence, evidence are found to both support or oppose the necessity of identifying a Proline residue on CDRs to be exclusively cis or trans conformation. On the opposing side, there are studies identified the Proline cis-trans isomerization occurred in folded proteins not only in non-antibody protein mentioned above but also in the antibody loops. An early study suggested that the presence of two consecutive Proline on L3 CDR enable cis-trans isomerization because of the highly strained conformation. A more recent study reported CDR H3 Proline isomerized from cis to trans on an epiregulin (EPR) antibody upon binding to EPR[29]. Take L3-9 as the example again, one of the CDRs in L3-9-2 have consecutive Proline at 7^{th} and 8^{th} positions and all antibody structures within this cluster are from the antibody-antigen complexes, thus the Prolines with trans conformations in L3-9-2 may be results from isomerization of cis conformational CDRs due to the consecutive Proline loop strain or higher loop flexibility required by antibody antigen bindings. The small dihedral distances of cluster exemplars from the two clusters can also such conclusion.

On the supporting side, cis or trans conformation of Proline is not a trivial structure variant. There's an energy barrier of 14-24 kcal/mol[30] of transforming between the two protein in disordered

proteins suggesting its relative stable structure. And trans-Proline can exists on H1-13 CDRs of an antibody in its antigen free from demonstrated by the antibody structure(4LRI) with good resolution in PyIgClassify, which means not all trans conformational proline on CDR is a result of antigen binding. The cis conformation of certain Proline on an antibody Fv is found to be essential for its correct VH/VL interface formation during protein folding and its isomerization step serves as the rate limiting step, hence it is an important structural feature to capture during CDR modeling. In the modeling process, once proline on the query CDR is assumed to be either cis or trans conformation by selecting a template, Rosetta can't sample the possibility of it being the other conformation, therefore finding the most likely conformation of Proline sites is important for the ensuing modeling success.

VI. Rosetta Antibody non-H3 template searching method "BlindBLAST"

In Rosetta Antibody, structural template of a nonH3 CDR loop is found by BLAST search against a database of CDR loop sequences of the same length and CDR loop. Because the method does not utilize canonical CDR structural cluster, it is referred to as "blindBLAST". BLAST parameters used is "-substitution_matrix PAM30 -word_size 2 -max_target_seqs 3000 -evalue 2000". The template hits aligns with the query CDR are ranked by bitscore calculated by equation (1), in which only the similarity score determines the bitscores once the gap-penalty and substitution matrix is chosen. PAM30 is used because it models a relatively shallow evolutionary model in which homologous sequences with over 75% sequence identity at branches of phylogenetic trees which better describes the "shallow evolution" in antibody CDR generation. Gap penalty is the default value 11 and only 0.11% alignments are found with any gap out of the hits of all queries in this parameter setting, therefore the gap is not relevant in most cases. The method implemented in RosettaAntibody selects 10 CDRs with the highest bitscores for each query out of all candidate CDRs. The template structure is then grafted onto the modeled framework and subject to further energy minimization and structural refinement.

Cases of modeled antibodies with CDRs assuming non-query-cluster canonical structures have been reported from previous uses of Rosetta, most likely because of the template coming from non-

query clusters. The majority of these errors are concentrated in a few error types identified as misclassifying cluster X to cluster Y, or clusterX-clusterY misclassification type. The prevalence of certain misclassification types can come from two sources, one is the large member size in at least one cluster involved in the cluster pair, the other is the bias toward certain misclassification than others in the method. In this thesis, the top similarity template for each query is used for evaluating the effectiveness of blindBLAST in recovering canonical CDR structure of the query.

$$S = \sum_{i=1}^{l} u_i \, , i = 1 \ldots l \; position \; along \; the$$

$$S' = \frac{\lambda S - \ln (\mathrm{K})}{\ln (2)} \qquad (1)$$

VII. Machine learning on protein classification and data sampling scheme

Machine learning has been used extensively in protein classification problem such as predicting protein function, folding rate, superfamily, for fold recognition, enzyme class, functional binding sites. A machine learning classifier can be optimized by its training method, cost function, and sampling method. Various machine learning methods have been used in these studies. A recent study demonstrated that the ensemble method gradient boosted trees gives the best accuracy in tests of classifying dataset of Structural Classification Of Protein database (SCOP)[31]. The conclusion of this study is applicable for the method choice in CDR canonical cluster prediction because both the features and the nature of predicted class bear good similarity. Various sampling methods for combating the sample class imbalance have also been evaluated including down-sampling the majority class and up-sampling the minority class by resampling or adding synthetic cases in previous studies[32,33]. Results suggest none of the sampling methods are always better than others, but dataset and data size dependent, but down-sampling majority classes inside each weak learner in an ensemble of weak learners is reported to give

the most robust receiver operating characteristic(ROC). The advantage is reasoned to come from the edge given by independent realizations of weak learners trained by more distinct sets of samples in majority classes.

VIII. The goal of the thesis

In this thesis, I aim to improve non H3 CDR homology modeling by finding template which are more structurally similar than current Rosetta template searching strategy achieves. This is achieved by introducing machine learning classifier trained for each CDR loop and length of the CDR queries prior to specific template selection by similarity score. I evaluate its effectiveness by collecting the prediction accuracy improvement compared to simply BLAST at the resolution of specific misclassification types responding to the canonical cluster pairs. **Chapter II** gives the methods I'm using for these analysis with the corresponding results listed in the **Chapter III Results** section. The reasons for success and failure of using the machine learning classifier and the features most important for class predication are inferred from the result, and summarized in **Chapter IV Discussion**.

Chapter II: Methods

I. Dataset

A set of non-redundant canonical CDR loops was obtained from the PyIgClassify. In the PyIgClassify set, the CDR loops are partitioned by CDR loop (L1, L2, L3, H1, H2, and H3) and length, each of which are further clustered by their structures, so that the cluster members are always more structurally similar to their own cluster exemplar than any other cluster exemplars. The distribution of CDR cluster membership is very unbalanced, with each CDR loop and length pair having generally one well-populated cluster and many other sparsely-populated clusters. The overall cluster membership distribution by CDR loop is plotted as Figure 2-1.

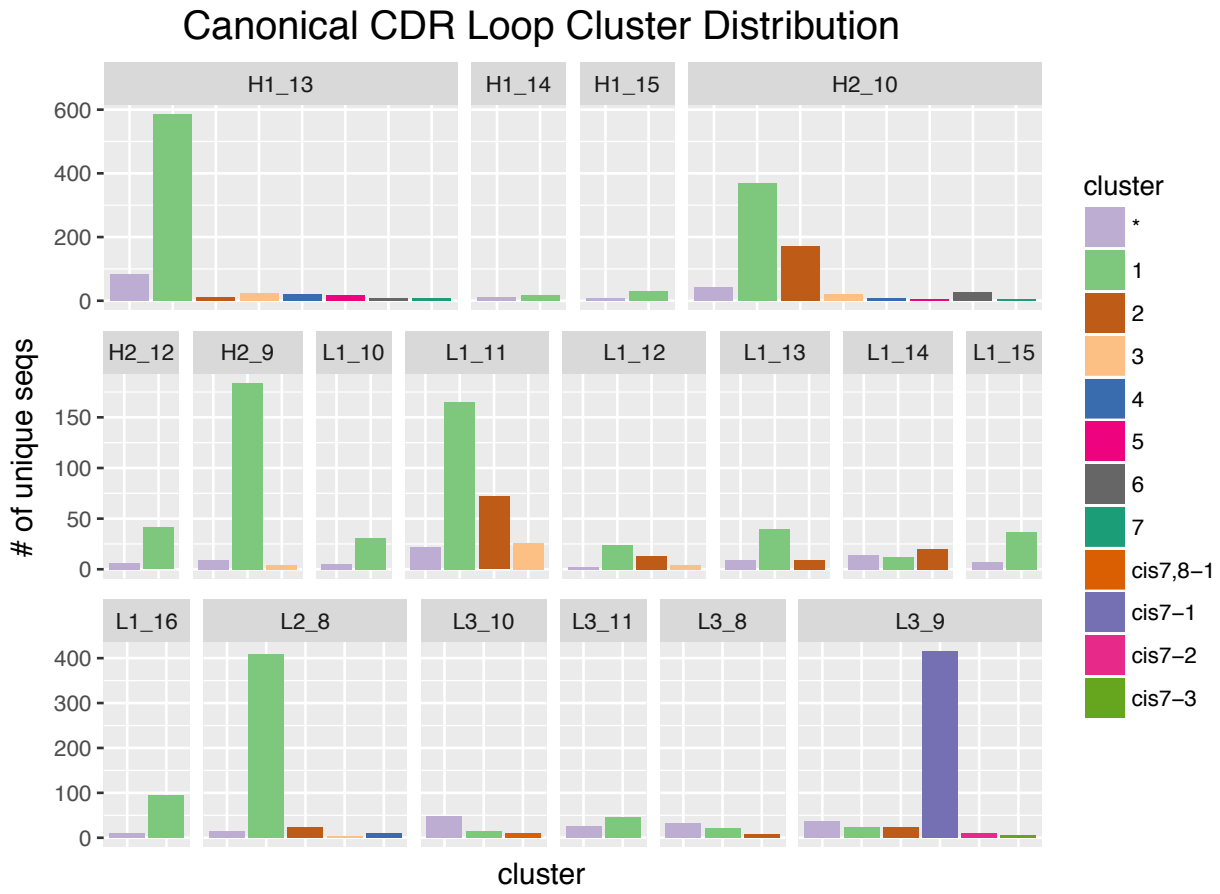


Figure 2-1. Canonical CDR loop cluster distriburion:
The barplot represents the non-redundant case number distribution of structure clusters in different cdr loop lengths in current PyIgClassify database. Some loop length have very sparse data while other loop length are very populated. The same pattern is observed for the structure clusters distribution within each loop length.

II. Methods for evaluating the CDR loop structures difference.

a). Structural difference between each pair of CDR loops

i). Pairwise dihedral distance

The PyIgClassify database is clustered based on pairwise dihedral angle distances, within CDRS of the same loop and length group. The ϕ and ψ dihedral angles are defined by backbone atoms of a given residue (k) and it's neighbors (k$\pm$1): $C_{k-1}, N_k, C_{\alpha,k}, C_k$ and $N_k, C_{\alpha,k}, C_{\alpha,k}, N_{k+1}$, respectively. The pairwise loop dihedral angle distance is denoted as $D(i,j)$ for loop pair i and j and calculated by the following equation:

$$D(i,j) = \sum_{k=1}^{n_{res}} (2\left(1 - \cos\left(\phi_{i,k} - \phi_{j,k}\right)\right) + 2\left(1 - \cos\left(\psi_{i,k} - \psi_{j,k}\right)\right)) \quad (2)$$ [34]

$\phi_{i,k}$ denotes ϕ dihedral angle at k^{th} residue, $\phi_{i,k}$ denotes ϕ dihedral angle at k^{th} residue.

ii). Pairwise rmsd

In addition to dihedral distances, a common measure of loop similarity is RMSD. To calculate RMSD, a pair of CDRs in the same loop and length group is aligned by the 5 residues at the loop anchor sites (5 residues before/after the loop in question), then the RMSD between the two CDRs is calculated by:

$$rmsd = \sqrt{\frac{1}{N}\sum_{i=1}^{N}((v_{ix} - w_{ix})^2 + \left(v_{iy} - w_{iy}\right)^2 + (v_{iz} - w_{iz})^2)} \quad (3)$$

i from 1 to N are the backbone atoms N, C_a, C along the CDR loop

b). Structural characterization of each cluster and cluster-wide structural comparison.

i). Measuring dihedral distance between cluster exemplars

The cluster exemplar, which is the median structure to all of the cluster members, has been defined by the PyIgClassify database. The structural difference between two clusters can then be measured as the dihedral distance between cluster exemplars. Therefore, dihedral distances of such pairs are extracted and used for comparison.

ii). Mean and variance of dihedral angles at each position per cluster

Within each cluster, the dihedral angle mean and standard deviation are calculated for each residue. The mean dihedral angle is found as an angle value that can minimize the dihedral angle variance from all members at the angle position. The standard deviation is the square root of this variance. The means and variances are used for demonstrating the pattern of structural divergence

among clusters and within a single cluster. Corresponding equations are listed in supplementary material.

III. Categorizing misclassification types observed in blindBLAST

a). Construction of null model

A null model is constructed to simulate the prevalence of misclassifications between clusters assuming the prediction method has no true discriminative power. In the null model, cluster membership is randomly assigned to every CDR in PyIgclassify dataset according to the real cluster member size distribution in the dataset. Such sampling is performed 1000 times. In each sampling, an error case is identified when the cluster of the query CDR and the randomly assigned cluster are different as illustrated in Figure 2-2. The empirical error count distribution for each misclassification can be obtained under the null model. An example of the random assignment error count distribution is shown in Figure 2-3.

b). blindBLAST Leave-One-Out-Crossvalidation

The CDR misclassification profile of the blindBLAST method is constructed to evaluate the misclassification bias present in blindBLAST. BlindBLAST searching is conducted in LOOCV setting, in which candidate templates for a CDR query are all the CDRs of the same length and type, excluding the query itself. The template CDR that gives the highest similarity score is chosen as the template for each query as described in the introduction section. An error case is identified when the cluster of a query and the cluster of its selected template are different.

c). Significance test on cluster A-cluster B misclassification

The bias of blindBLAST toward certain cluster-to-cluster misclassifications can be evaluated by comparing the observed blindBLAST error count and the empirical error count under the null model. A two-tailed hypothesis test at 0.05 level is formulated as the following equation: x_{n}

H_0: observed x_{error} consistent with random assignmnet

H_a: observed x_{error} not consistent with from H_0

$$\alpha = 0.05$$

$$F_{x_{\text{error}}} = \frac{\sum_{n=1}^{N_{\text{all}}=1000} I(x_{\text{n}}^* \geq x_{\text{error}})}{N_{\text{all}}} \quad (4)$$

$$p = \min\left[F_{x_{\text{error}}}, 1 - F_{x_{error}}\right]$$

$$p \leq 0.025, H_0 \text{ is rejected}$$

$$p > 0.025, H_0 \text{ is not rejected,}$$

in which x_{error} is the error count from LOOCV blindBLAST and x_n^* with n from 1 to 1000 are the error counts for 1000 iteration random assignment simulation.

A quantity termed "effect size" quantifies how many is the misclassification count from results of blindBLAST different from the mean value of the count in the null model by the standard deviation of the count in the null model, offering quantification of the difference beyond significance.

$$z = \frac{x - \bar{x}_{\text{simu}}}{s_{\text{simu}}} \quad (5)$$

, in which x and $\bar{x}_{\text{simu}}, s_{\text{simu}}$ are the blindBLAST derived count, the average and standard deviation of the corresponding count from the random cluster assignment null model. The values are plotted in the Figure 5-7.

d). Misclassification grouping

Cluster misclassifications can be grouped by whether or not a specific misclassification occurs more frequently in blindBLAST than in the null model, based on the value of $F_{x_{\text{error}}}$. For example, values of $F_{x_{\text{error}}} \geq 0.975$ indicate that blindBLAST is misclassifying loops significantly more than null model from one cluster to another. The remainder are grouped based on both the specific x_{error} and the value of $F_{x_{\text{error}}}$, with detailed discussion in the Results.

Because for a CDR query belong to a cluster, it can be correctly classified only if blindBLAST avoid all possible wrong classification of this query. For a broader view, the same test hypothesis test is also performed using the recovery count of a cluster instead of all possible misclassifcations, shown in Figure 5-5 in Supplementary material.

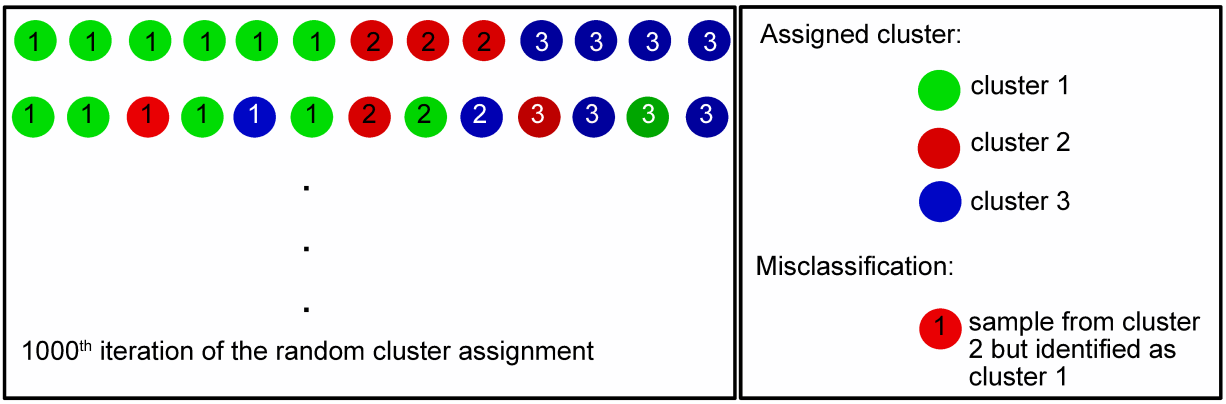


Figure 2-2. Schematics of random assignment simulation for each loop and length type: For each loop type and length combination, its CDR members are assigned with cluster number randomly as indicated by the number inside each circle. When the number does not match to the real cluster indicated by the color code, one case is added to the error count of the corresponding misclassification. This random assignment simulation is repeated 1000 times.

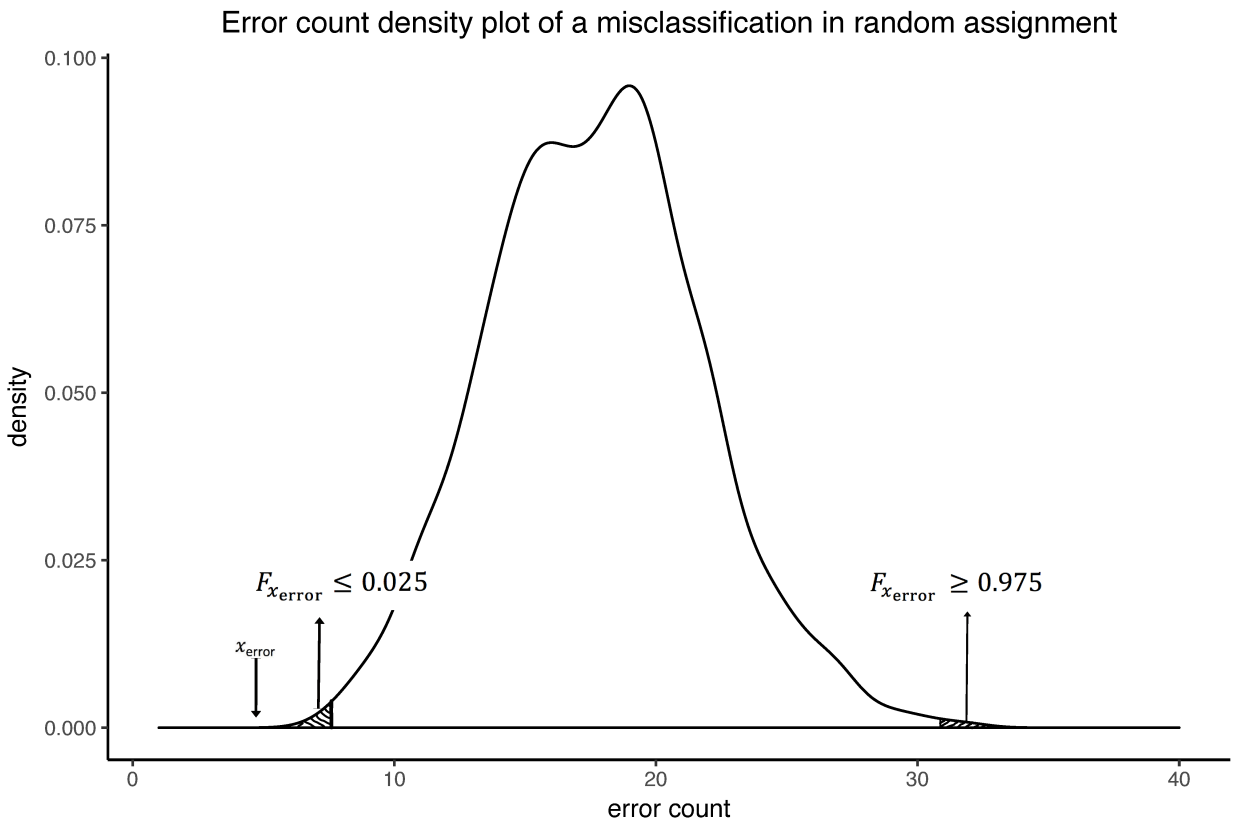


Figure 2-3. Error count density plot of H1-13-1_h1-13-2 in random assignment:
The region where the error count corresponds to the smallest 2.5% or greatest 2.5% of all the error counts are considered to be the critical region. If the error count of the same misclassification from LOOCV blindBLAST falls in the critical region its null hypothesis will be rejected.

IV. Evaluating Canonical CDR Modeling within Rosetta Antibody (blindBLAST)

To permit comparison with machine learning approaches, in addition to LOOCV, a 10-fold cross-validation (CV) scheme is used to evaluate canonical cluster identification performance by blindBLAST.

To that end, for each CDR length and loop type, the dataset is divided into ten folds and for each validation, one fold comprises the query (test) set and the remaining nine comprise the template (training) set. To evaluate accuracy, this stratified 10-fold CV is repeated three times as illustrated in Figure 2-4. The average CDR cluster identification accuracy for each non-H3 loop is calculated by averaging the corresponding acc_1 to acc_3. The stratified fold division ensures CDRs clusters composition in every query set and training set each resembles that in the entire dataset.

In addition to evaluating cluster classification accuracy, I assess the impact of selecting a template from the wrong cluster by structural alignment. Following triplicate 10-fold CV, for query sequences with incorrectly predicted clusters, query sequences, structures, and corresponding predicted template structures are extracted. Then, blindBLAST search is repeated for those query sequences except only against CDRs from the correct cluster to extract a "correct" template. Finally, the query structures are aligned to both templates by superimposing the five residues flanking either end of the CDR loop and the RMSDs are calculated.

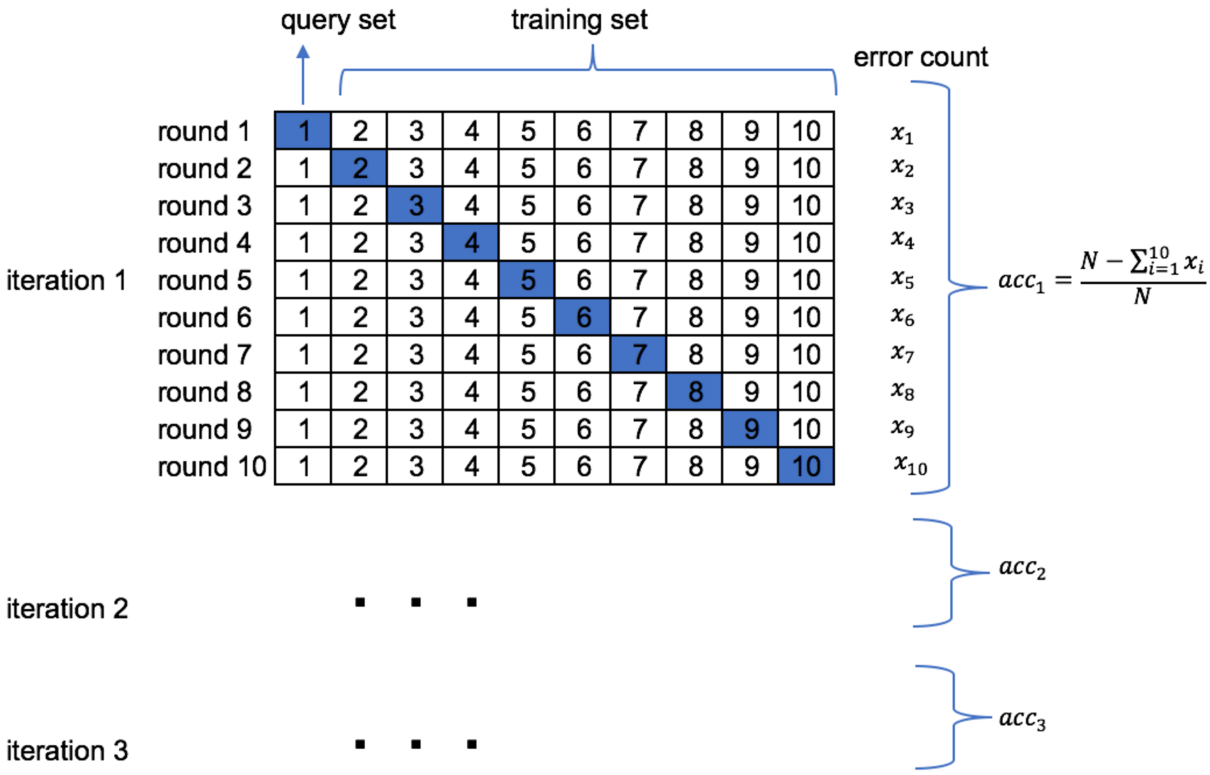


Figure 2-4. Three repeats 10 folds cross validation:

Error count is the number of errors in the query set for each round. N is the total of case number in the query set and training set put together. The term acc_1 denotes the cluster identification accuracy in the 1st repeat of 10-fold cross validation, acc_2 and acc_3 are prediction accuracies resulted from different fold divisions.

V. The guidedBLAST method

a). Machine learning algorithm selection.

I propose "guidedBLAST", a method that introduces a machine-learning model to predict the cluster membership before using BLAST to search for a template within the predicted cluster. A preliminary search for the machine learning approach with high accuracy and low model complexity approach to this classification task was performed. The surveyed methods include linear models, decision trees, support vector machines and gradient boosting machines (GBM). GBM was chosen as the best learning method for predicting CDR clusters.

b). Features

I use known antibody sequence as features to train machine learning classifiers for predicting CDR cluster membership of antibodies with unknown structures. The feature set included amino acids identities of the CDR loop and the 10 flanking upstream and downstream residues. At each position on the loop, the presence and absence of each of the 20 amino acids was represented by a binary feature (0 or 1).

c). Model tuning

The GBM model complexity is tuned for each CDR loop and length combination independently. The tuning process starts with a search over a hyper-parameter grid, where each grid corresponds to different model complexity. Each grid specifies the set of parameters used when the models are trained with the same data division scheme as blindBLAST as illustrated in Figure 2-4. To cope with the class imbalance problem (some cluster more well represented than others), 9 out of 10 folds used as the training data in each round undergo an additional sampling step, in which cases belonging to the less popular classes are resampled to match the number of cases belonging to the most popular class,

therefore attain a balanced clusters-member-size distribution in the training set. The model estimation accuracy for the parameter grid was then obtained by averaging the results from cross validation. The best model is chosen by choosing the grid that gives better model estimation accuracy than grids representing lower complexity, but not lower than the average of the next 5 more increasingly complex ones.

d). Variable importance

The features that are most important to the classifications are extracted from the best GBM model. These features are ranked by their importance. The feature residue i at position j with a relative importance scale of 100 is considered to be the most important feature in classifying clusters in this CDR loop and length type. The importance is calculated by first calculating how much a decision tree split reduces Gini impurity after the split(4), then summing over all node-size-weighed reductions on splits corresponding to that feature over all boosted trees as equation(5). The variable importance can be viewed as the distinguishing power of that feature accumulated over the entire training process.

Gini impurity:

$$I_G = 1 - \left({p_1}^2 + {p_2}^2 + \cdots + {p_j}^2\right) \; for\ class\ 1 \ldots j \qquad (4)$$

$$Vi = \sum_{1}^{n_{iter}} \sum_{i} (n_{base} * I_G(\text{base node}) - n_{left} * I_G(\text{left node}) - n_{right} * I_G(\text{right node})) \qquad (5)$$

VI. AMAII comparison between GBM guidedBLAST, blindBLAST, FREAD, Disgro.

For the antibodies in the second antibody modeling assessment (AMA II), I calculated the RMSD between the predicted template versus actual query CDR structures using different loop modeling methods. This is done to extend the performance comparison of guidedBLAST to other established template searching methods. The antibodies in the AMAII dataset are excluded from method template libraries except for DiSGro, which is a non-homology modeling method and does not rely on templates.

a). FREAD-3.0.1

FREAD version 3.0.1 was downloaded and installed locally. The FREAD template search was executed by the following command:

./FREAD –f query_pdb –l loop_start –s loop_end -b PyIgClassify_antibody_database –C -10 -m output_log –r

The template with the highest similarity score is extracted from the output log file and its RMSD to the query CDR extracted from the pre-calculated RMSD table.

b). DiSGro

DiSGro was downloaded and installed locally. For each query antibody CDR, the query chain structure was extracted from the antibody and the to-be-modeled loop had its coordinates set to 0.0 (as instructed by the DiSGro manual). For each loop, five thousand candidate structures are generated with only the 100 best output. The most confident model is selected as the best model, and have its RMSD to the query CDR calculated by aligning the 5 residues stem region flanking the CDR on each loop end. A sample DiSGro command line is:

./disgro -mode smc -f model_seq -n 5000 -nds 32 -start loop_start -end loop_end -eval -confkeep 1000 -ellip -nscc 20 -pdbout 100

c). GBM-guided-BLAST and blindBLAST (Rosetta Antibody)

The GBM-guided-BLAST and blindBLAST is executed for all the loops as IV and V in Methods. For every query loop, the RMSDs of the loop aligned with either template searched by blindBLAST or

template searched within the CDRs templates corresponding the GBM predicted cluster are extracted from precaculated RMSD table.

Chapter III: Results:

I. Misclassification grouping identifies the problematic cluster pairs prone to be misclassified.

To answer which cluster-A-to-cluster-B misclassification is more important to improve, the misclassifications are categorized. The categorization is based on the prevalence of error in blindBLAST and how is the value compared to random assignment model in the statistical test listed in Equation (4). The resulting groups are listed below with its members discussed.

a). Significantly worse than random assignment:

Misclassifications with $F_{x_{\text{error}}} \geq 0.975$ and blindBLAST LOOCV $x_{\text{error}} > 3$ are categorized as significantly worse than random assignment, and listed in Table 3-1. In this category, there are some template candidates belonging to one cluster, but gives very high similarity score when it pairs with multiple query sequences in another cluster. These high similarity scores are directly responsible for the high number of error count. For example, six out of the eight error cases in the L2-8-1 to L2-8-5 misclassification are caused by a single CDR template candidate in L2-8-5.

Because of the over presence of certain templates that generate a lot of misclassification cases, there may be some residues pair that gives a large favor to the similarity scores of these wrong alignments. I examined the identities of these amino acid pairs.

For these error cases, the corresponding right query-template alignment can be found by constraining the BLAST searching inside the right cluster of the query. Since similarity scores are the

sum of amino acids substitution scores along each alignment position. The amino acids pair that gives the largest favor to the wrong alignment compared to the correct alignment are extracted. Misclassifications from L2-8-1 to L2-8-5 are caused by EE alignment favored over ED at the 7th position of the loop(five out of eight cases), and the remaining three cases are caused by TT alignment being favored over TS, TN, or TF at variable positions. For all misclassifications cases in this category, the amino acid substitution pairs are shown in Table 3-2. In these error cases, PAM30 gives a large favor to wrong alignment substitution GG compared to GD/E/A/ for H1-13, similarly, to EE compared to ED or TT compared to TS/N/F in H2-8, and II compared to IS/T/R, or WW compared to WA/S, or YY compared to YS/N in H2-10.

b). Similar to random assignment, but with greater than 3 error count

The misclassifications with $0.025 < F_{x_{error}} < 0.975$ and LOOCV $x_{\text{error}} > 3$ have blindBLAST performance consistent with random assignment model performance at 0.5 significance level. As shown in Table 3-4, a lot of them appear in pairs, for example L2-8-2 to L2-8-1 misclassification and L2-8-1 to L2-8-2 misclassification. The pair corresponds to two clusters with usually one of them being the most popular cluster in the loop type. The only misclassifications between both non-popular clusters in this group are in the H2-10. Misclassification pairs of L2-8-2 and L2-8-1; H1-13-1 and H1-13-4; H2-10-1 and H2-10-6 are among the pairs with the top error counts with misclassification of L3-9-2 to L3-9-ci7-1 being a singlet in the top error counts.

c). Significant improvement over random assignment, but with more than 3 error count

This group consists of misclassifications with $F_{x_{error}} \leq 0.025$ and LOOCV $x_{\text{error}} > 3$ as shown in part of Table 3-4, sorted by blindBLAST LOOCV x_{error}. They are the ones with significant improvement but still having substantial error counts. Many of the top ones appear in pairs and have one member of the pair to be the most popular cluster as well, and belonged to many of the loop and length types proved

problematic in the previous misclassification types including H1-13 and H2-10, L3-9. The other top ones are L1-11 and L3-8.

d). Significant improvement over random assignment, with less than 3 error count

This group consists of misclassifications with $F_{x_{error}} \leq 0.025$ and error count LOOCV $x_{\text{error}} \leq 3$ listed in part of Table 3-4. Misclassifications in this group are rescued by BLAST from random assignment the most. They appear in pairs as well with generally one member being one of the most popular clusters, with the exception of misclassifications between L1-11-2 and L1-11-3; H2-10-3 and H2-10-2 not involving the most popular cluster in a CDR loop and length type.

error count	mean simu error count	sd	query cluster	template cluster	significance
significantly worse than random					
13	5	2.1	H2-10-2	H2-10-none	1
11	2.8	1.7	L2-8-1	L2-8-5	1
7	2.5	1.3	H2-10-4	H2-10-2	1
6	0.6	0.8	L1-11-3	L1-11-none	1
5	0.9	0.9	H1-13-5	H1-13-3	1
5	1.7	1.2	L3-10-none	L3-10-cis7,8-1	0.997
4	1	1	H1-13-3	H1-13-2	0.999
4	0.9	0.9	H1-13-2	H1-13-3	0.999
4	0.8	0.9	H2-10-none	H2-10-6	0.997
3	0.7	0.8	H1-13-2	H1-13-4	0.995
3	0.3	0.6	H1-13-5	H1-13-7	0.998
3	0.6	0.8	L1-11-none	L1-11-3	1
3	0.8	0.8	L3-10-cis7,8-1	L3-10-cis8-1	0.999

Table 3-1. Significantly worse misclassification using blindBLAST instead of random simulation:
"query cases" denotes the number of error count that fall into the misclassification type of query cluster to template cluster. The "template cases" denote the number of unique template sequences corresponding to the listed query sequences. Largest repeat denotes the number of being found as the most similar CDR in the misclassification.

q-wt	q-rt	pos	q-cluster	wt-cluster
			DA	
DA	DY	11	H1-13-3	H1-13-2
DD				
DD	DP	11	H1-13-3	H1-13-2
DD	DG	5	H1-13-2	H1-13-3
DD	DK	10	H2-10-none	H2-10-6
			DE	
DE	DA	7	L2-8-1	L2-8-5
DN				
DN	DY	10	H1-13-5	H1-13-3
			EE	
EE	ED	7	L2-8-1	L2-8-5
EE	ED	7	L2-8-1	L2-8-5
EE	ED	7	L2-8-1	L2-8-5
EE	ED	7	L2-8-1	L2-8-5
			FF	
FF	FQ	1	H2-10-4	H2-10-2
			FL	
FL	FV	1	H2-10-2	H2-10-none
			GG	
GG	GR	6	L1-11-3	L1-11-none
GG	GD	13	H1-13-5	H1-13-3
GG	GE	4	H1-13-3	H1-13-2
GG	GA	11	H1-13-2	H1-13-3
GG	GA	6	H2-10-none	H2-10-6
GG	GP	4	H2-10-none	H2-10-6
			II	
II	IS	2	H2-10-2	H2-10-none
II	IT	9	H2-10-4	H2-10-2
II	IR	9	H2-10-4	H2-10-2
II	IW	1	H2-10-none	H2-10-6
			KK	
KK	KF	10	H2-10-2	H2-10-none
KK	KR	8	L1-11-3	L1-11-none
KK	KR	8	L1-11-3	L1-11-none
KK	KR	8	L1-11-3	L1-11-none
			LL	
LL	LV	1	H2-10-2	H2-10-none
LL	LV	1	H2-10-2	H2-10-none
LL	LW	6	L3-10-none	L3-10-cis7,8-1
LL	LY	9	L3-10-none	L3-10-cis7,8-1

q-wt	q-rt	pos	q-cluster	wt-cluster
			LN	
LN	LK	8	H2-10-2	H2-10-none
			MM	
MM	MQ	1	L3-10-none	L3-10-cis7,8-1
			NN	
NN	NS	3	H2-10-2	H2-10-none
NN	NT	5	L2-8-1	L2-8-5
NN	NT	5	L3-10-none	L3-10-cis7,8-1
			RK	
RK	RY	10	H2-10-2	H2-10-none
			RR	
RR	RG	5	H1-13-5	H1-13-3
RR	RY	5	H1-13-2	H1-13-3
			SS	
SS	SK	7	H2-10-2	H2-10-none
SS	SL	2	L2-8-1	L2-8-5
SS	SR	7	L1-11-3	L1-11-none
SS	SN	5	H1-13-5	H1-13-3
SS	SV	8	H1-13-2	H1-13-3
			TT	
TT	TS	8	L2-8-1	L2-8-5
TT	TN	3	L2-8-1	L2-8-5
TT	TF	1	L2-8-1	L2-8-5
TT	TI	9	H2-10-4	H2-10-2
TT	TA	2	H1-13-3	H1-13-2
			VV	
VV	VS	11	H1-13-5	H1-13-3
			WW	
WW	WA	4	H2-10-2	H2-10-none
WW	WS	4	H2-10-2	H2-10-none
WW	WS	4	H2-10-2	H2-10-none
WW	WL	6	L3-10-none	L3-10-cis7,8-1
			YY	
YY	YS	8	H2-10-4	H2-10-2
YY	YS	10	H2-10-4	H2-10-2
YY	YN	10	H2-10-4	H2-10-2

Table 3-2. Amino Acids substitution pairs most responsible for significantly worse misclassification:

q-wt is residues pair aligned between query and template from a wrong cluster, q-rt is for the same position in the preceding q-wt, residue pair aligned between query and template from a right cluster, pos is the position along the loop, q-cluster is the query cluster, wt-cluster is the wrong template cluster which the specific q-wt alignment comes from.

error count	mean simu error count	sd	query cluster size	query cluster	template cluster	sig	error percentage
27	27.7	2.2	34	L3-9-2	L3-9-cis7-1	insignificant	0.79
25	44.9	4.3	75	L1-11-2	L1-11-1	smaller	0.33
21	22.5	1.5	25	L2-8-2	L2-8-1	insignificant	0.84
17	12.6	2.3	22	H2-10-3	H2-10-1	larger	0.77
15	25.5	2.3	32	H1-13-3	H1-13-1	smaller	0.47
14	18.2	1.9	23	H1-13-4	H1-13-1	insignificant	0.61
14	17.4	2.7	30	H2-10-6	H2-10-1	insignificant	0.47
10	9.0	0.9	10	L2-8-4	L2-8-1	larger	1.00
9	17.4	1.9	22	H1-13-2	H1-13-1	smaller	0.41
7	2.5	1.3	9	H2-10-4	H2-10-2	larger	0.78
6	6.5	1.1	8	L3-9-cis7-3	L3-9-cis7-1	insignificant	0.75
6	10.5	2.0	17	L3-8-2	L3-8-1	smaller	0.35
5	8.9	1.3	11	L3-9-cis7-2	L3-9-cis7-1	smaller	0.45

Table 3-3. Percentage of different misclassifications

The above analysis gives the performance of BLAST compared with random assignment. The performance of BLAST in terms of the error percentage of classifying one cluster into some other clusters with over 0.3 and error count greater than 4 are listed in the following figure.

error count	mean simu error count	sd	query cluster	template cluster	significance
not significantly different from random					
27	27.7	2.2	L3-9-2	L3-9-cis7-1	0.438
26	17.6	4.1	H2-10-1	H2-10-6	0.971
21	22.5	1.5	L2-8-2	L2-8-1	0.242
19	22.8	4.7	L2-8-1	L2-8-2	0.26
18	12.9	3.5	H2-10-1	H2-10-3	0.934
14	18.2	1.9	H1-13-4	H1-13-1	0.029
14	17.4	2.7	H2-10-6	H2-10-1	0.147
13	8.4	2.7	H2-10-2	H2-10-6	0.968
11	18.5	4.2	H1-13-1	H1-13-4	0.054
9	8.9	2.9	L3-9-cis7-1	L3-9-cis7-2	0.601
9	8.1	2.6	L3-10-1	L3-10-none	0.733
6	11.3	3.2	H1-13-1	H1-13-6	0.066
6	6.5	1.1	L3-9-cis7-3	L3-9-cis7-1	0.457
6	5.4	2.3	L1-16-1	L1-16-none	0.714
5	7.3	2.7	H1-13-1	H1-13-9	0.268
5	8.2	2.5	H2-10-6	H2-10-2	0.137
4	9.7	3.1	H1-13-1	H1-13-none	0.039
4	2.3	1.3	H2-10-7	H2-10-2	0.949
4	5.1	1.9	H2-10-none	H2-10-2	0.406
4	2.7	1.6	L3-9-1	L3-9-2	0.851
4	1.9	1.3	L1-10-1	L1-10-2	0.963
significantly better than random but still a lot of error counts					
25	44.9	4.3	L1-11-2	L1-11-1	0
15	25.5	2.3	H1-13-3	H1-13-1	0
14	25.7	5	H1-13-1	H1-13-3	0.011
13	44.4	5.7	L1-11-1	L1-11-2	0
12	102.2	6.6	H2-10-2	H2-10-1	0
12	27.6	5	L3-9-cis7-1	L3-9-2	0
9	17.4	1.9	H1-13-2	H1-13-1	0
8	103.4	8.4	H2-10-1	H2-10-2	0
6	10.5	2	L3-8-2	L3-8-1	0.021
5	8.9	1.3	L3-9-cis7-2	L3-9-cis7-1	0.01
4	7.2	1.2	H1-13-9	H1-13-1	0.016
4	9.5	1.4	H1-13-none	H1-13-1	0.001
4	17.7	4.1	H1-13-1	H1-13-2	0
4	10.7	2.8	L3-8-1	L3-8-2	0.009
4	8.1	1.7	L1-12-2	L1-12-1	0.013

error count	mean simu error count	sd	query cluster	template cluster	significance
significantly better than random and error count<=3					
2	33.5	2.4	L3-9-1	L3-9-cis7-1	0
0	33.1	5.5	L3-9-cis7-1	L3-9-1	0
1	22.0	2.9	L1-11-3	L1-11-1	0
1	21.7	4.4	L1-11-1	L1-11-3	0
3	15.5	3.9	H1-13-1	H1-13-5	0
2	15.0	1.8	H1-13-5	H1-13-1	0
3	11.2	1.5	H1-13-6	H1-13-1	0
0	10.7	2.8	L1-13-1	L1-13-2	0
0	10.7	1.8	L1-13-2	L1-13-1	0
3	10.5	3.2	H2-10-1	H2-10-none	0.008
2	10.3	1.5	H1-13-7	H1-13-1	0
3	10.3	2.1	H2-10-none	H2-10-1	0
3	10.3	3.2	H1-13-1	H1-13-7	0.009
0	9.9	3.1	L1-11-2	L1-11-3	0
0	9.8	2.6	L1-11-3	L1-11-2	0
1	9.3	3.1	L2-8-1	L2-8-4	0.001
2	8.9	2.4	L1-14-2	L1-14-1	0.003
1	8.5	1.7	L1-14-1	L1-14-2	0
3	7.9	2.4	L1-12-1	L1-12-2	0.024
3	7.9	1.7	L3-10-none	L3-10-1	0.003
1	6.6	2.6	L3-9-cis7-1	L3-9-cis7-3	0.007
1	6.2	2.3	H1-14-1	H1-14-none	0.01
0	6.1	2.1	H2-10-3	H2-10-2	0.001
0	6.0	2.4	H2-10-2	H2-10-3	0.005
1	5.6	2.2	L3-10-1	L3-10-cis7,8-1	0.018
0	5.6	1.4	L3-10-cis7,8-1	L3-10-1	0
2	5.4	0.8	L2-8-3	L2-8-1	0.002
0	5.2	1.5	H2-10-4	H2-10-1	0
0	5.0	2.1	L3-11-1	L3-11-cis7-1	0.006
0	4.5	0.7	L3-11-cis7-1	L3-11-1	0
0	3.8	1.2	L3-10-cis8-1	L3-10-1	0.006
0	2.4	1.0	L1-12-3	L1-12-1	0.024
0	2.4	0.7	L2-12-1	L2-12-2	0.006

Table 3-4. Misclassification types by blindBLAST performance group:
Each misclassification is defined with two ordered structure clusters with its corresponding error count being the number of CDRs belonging to the first cluster misclassified to the second cluster.

II. BLAST good at distinguishing some clusters but bad at others

The misclassification categories can be further divided into the ones that are good and the ones that are bad. The explanations are sought for why some misclassifications are good but some are bad.

The criteria for good and bad are based on my opinion that the misclassification with less than 3 error count and significantly smaller than the random assignment are not urgent to be improved and are the good ones. The first 3 categories "Significant improvement over random assignment", "similar to random assignment but with greater than 3 error count" and "significant improvement over random assignment but with more than 3 error count" are considered to be the bad group.

a). Cluster exemplar distances affect classification accuracy

A comparison shows that the good misclassifications generally have greater cluster exemplar to exemplar distances than that of the bad misclassifications(>4 error counts), shown in Table 3-5. Only one exception is found as H1-13-7 and H1-13-1. It has relatively small between-exemplars dihedral distance but is one of the greatly improved ones in the loop H1-13.

b). Similarity score cause misclassification problem

The similarity score is previously defined in equation (1), and determines which template candidate is chosen for a query. Four clusters are picked out to examine the difference between the best similarity score within the cluster the best similarity score outside of the cluster. These Clusters H1-13-1, H2-10-1, L2-8-1, L3-9-2 have bad recovery, especially L3-9-2 has bad recovery of . The The misclassification L3-9-2-L3-9-cis7-1 is the worst one as almost all the L3-9-2 cases are predicted as L3-9-cis7-1 . The highest query-template similarity score in LOOCV setting and in correct-cluster-constrained setting for each query as x and y-axis are plotted for all CDR members in these clusters as Figure 3-1. Many queries in these clusters, especially in L3-9-2, have a large preference to templates from incorrect clusters. And in addition to the error cases, a lot of right cases lie on the border of the diagonal line, suggesting the similarity scores generated by BLAST using PAM30 are not discriminative

between right and wrong clusters. The result indicates similarity scores using PAM30 is effective in cluster identification for some cases but not for the other cases.

Besides the fact that the error cases are directly caused by the favorable similarity score of query-with-wrong-cluster-template-alignment, whether a query in the cluster H1-13-1 would be classified wrong are found to be also related to the relative position of the query structure to its cluster exemplar and how populated is its neighboring structures.

Specifically, for each of the mentioned four clusters, the dihedral distance to the cluster median density distribution is compare between the right cases and the wrong cases. And the density of the number of member CDRs existed within 1/10 of the cluster radius length with the center being a query CDR are compared between the right cases and right cases.

i). Misclassified cases are enriched in the distal end in the spectrum of query to cluster center distances.

Figure 3-2 indicates there is enrichment of unmatched cases in the distal end of to-cluster-center distances of H1-13-1 query cases, meaning the structures on the distal end of H1-13 more frequently find similar sequence in other cluster than from its own. The other three clusters does not show such enrichment. The H2-10 and L2-8 have broad peaks in the middle quantile with L2-8 having a small dent in left which is not conclusive enough.

ii). Misclassified cases are enriched in the smaller number region in the spectrum of number of structure neighbors of all query cases.

Figure 3-3 answers whether the chance of a query being identified with wrong query is affected by how populated is its neighboring structures. Comparison between the density plot of the right cases and wrong cases suggests the wrong cases are enriched in low count region compared to the density of right cases in clusters H1-13-1, while such trend is absent in H2-10-1 and L2-8-1. The matched case number in L3-9-2 is too small to lead to anything conclusive.

These finding suggest that the two factors, relative structural position to the cluster exemplar and the number of structural “neighbors” affects the identification accuracy, if the query CDR comes

from a relatively populated cluster such as H1-13-1. These within cluster factors have less impact if the cluster of the query is not as populated, in which case other clusters offer wrong templates to the query, likely because of its structural proximity to the query cluster.

L3-9		
good		
L3-9-1	L3-9-cis7-1	21
L3-9-cis7-1	L3-9-1	21
L3-9-cis7-1	L3-9-cis7-3	10
bad		
L3-9-2	L3-9-cis7-1	6.9
L3-9-cis7-1	L3-9-cis7-2	9.7
L3-9-cis7-3	L3-9-cis7-1	10.4
L3-9-cis7-2	L3-9-cis7-1	9.7
L3-9-cis7-1	L3-9-2	6.9
H1-13		
good		
H1-13-5	H1-13-1	27
H1-13-6	H1-13-1	23
H1-13-7	H1-13-1	12
H1-13-1	H1-13-5	27
H1-13-1	H1-13-7	12
bad		
H1-13-4	H1-13-1	17
H1-13-1	H1-13-4	17
H1-13-1	H1-13-6	23
H1-13-1	H1-13-9	16
H1-13-2	H1-13-1	11
H1-13-3	H1-13-1	16
H1-13-1	H1-13-3	16

H2-10		
good		
H2-10-3	H2-10-2	20.7
H2-10-2	H2-10-3	20.7
H2-10-4	H2-10-1	19.2
H2-10-none	H2-10-1	
H2-10-1	H2-10-none	
bad		
H2-10-1	H2-10-6	6.8
H2-10-1	H2-10-3	9.4
H2-10-6	H2-10-1	6.8
H2-10-2	H2-10-6	8.8
H2-10-6	H2-10-2	8.8
H2-10-2	H2-10-1	11.3
H2-10-1	H2-10-2	11.3
L2-8		
good		
L2-8-3	L2-8-1	11
L2-8-1	L2-8-4	8.8
bad		
L2-8-2	L2-8-1	4.5
L2-8-1	L2-8-2	4.5

Table 3-5. Between cluster center dihedral distances of misclassification pairs:
"Good" and "bad" which blindBLAST did significantly better versus distances of those not. The first column is the true cluster, the second column is the predicted cluster, the third column is the between cluster dihedral distance, which is defined by equation (2).

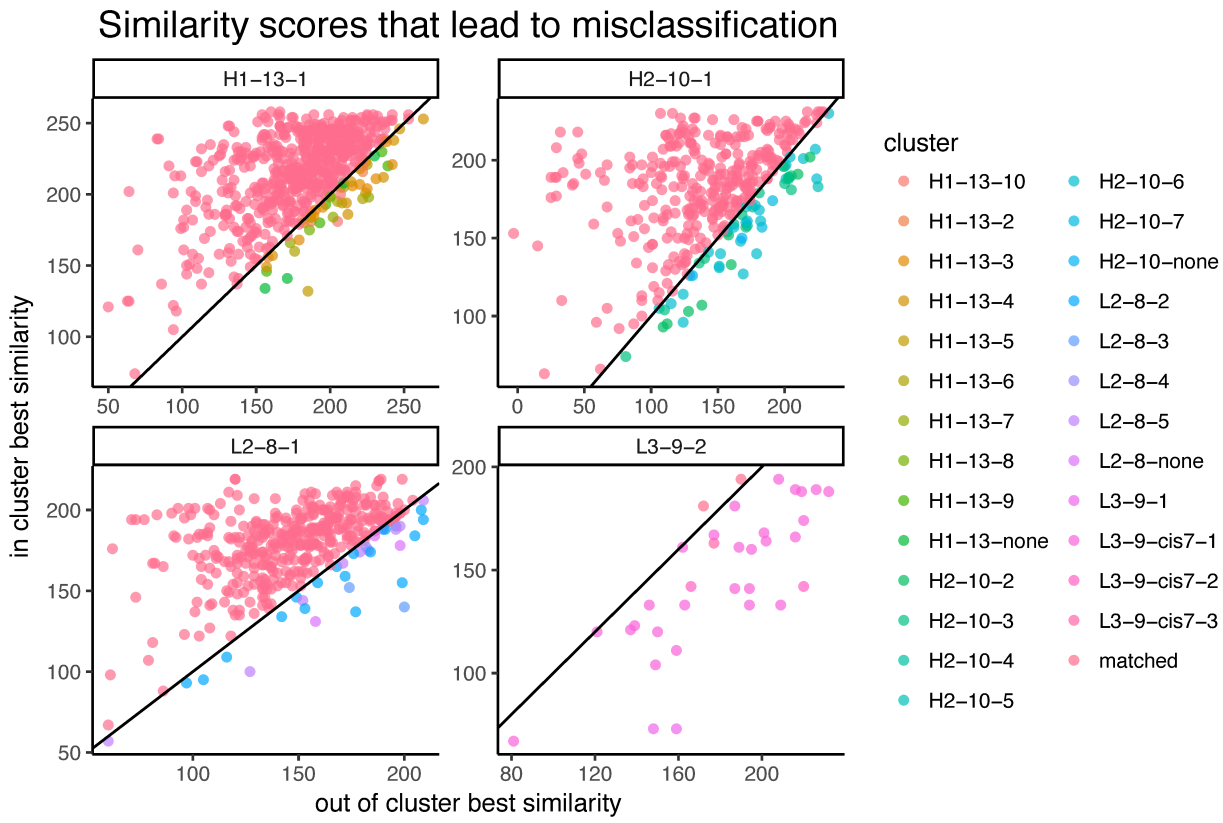


Figure 3-1. Similarity scores that lead to misclassifications

Caption: The dots on the upper left above the diagonal line are the CDRs that find the template in its structure cluster, the points on the lower right are the CDRs with most similar sequence in different clusters, each of the incorrect clusters are color coded as the legend shows.

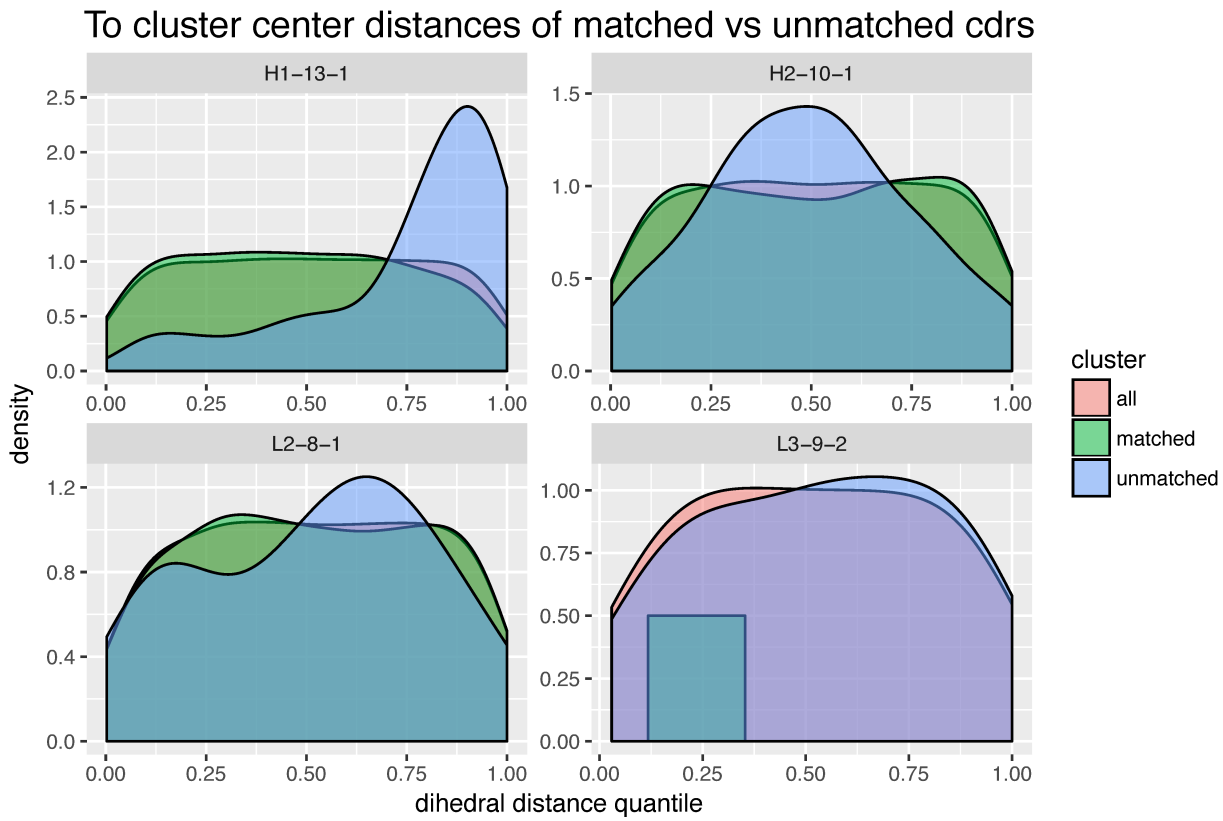


Figure 3-2. To cluster center distances of matched vs unmatched cdrs.
The x axis are the quantile of the dihedral distance of a CDR to the cluster exemplar among all such distances between CDR members in the cluster to lthe cluster exemplar.

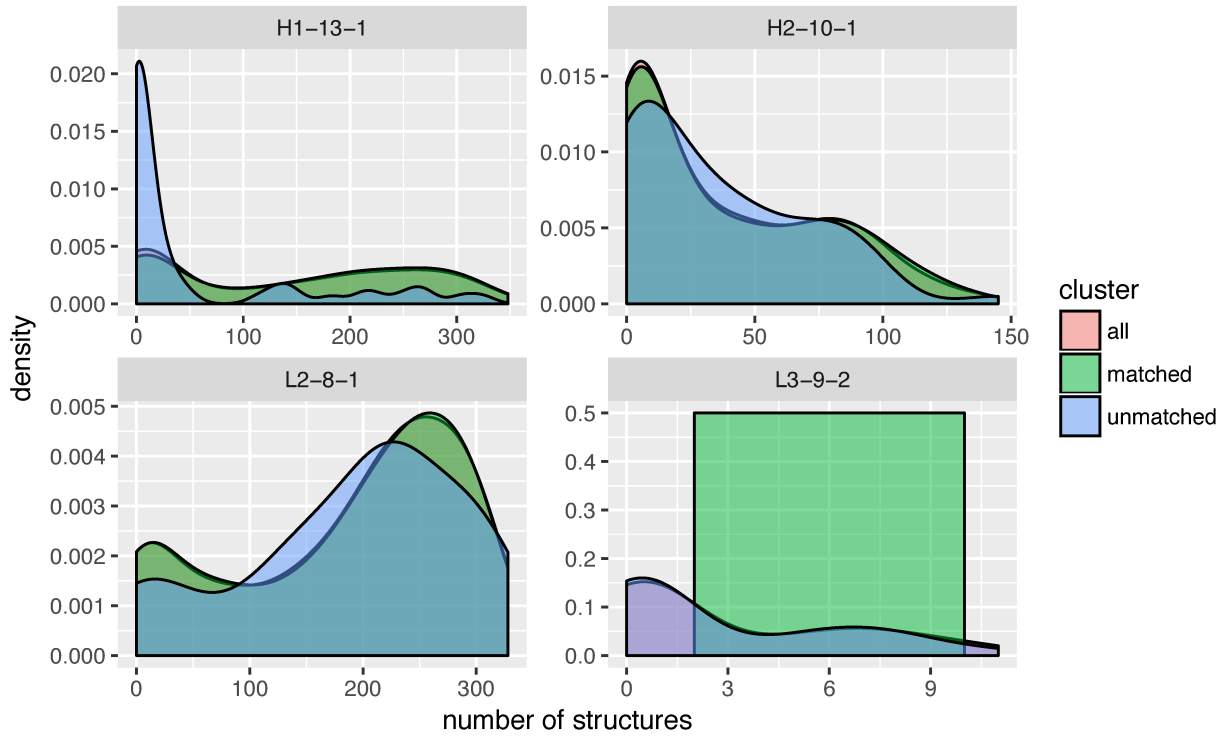


Figure 3-3. Number of structures neighboring the matched and unmatched CDRs: Query-and-template-cluster-unmatched cases are more likely to have sparse structural neighborhood in some loops. This is demonstrated especially in the H1-13 panel of the plot. x axis is the number of query-and-template-cluster matched or unmatched cases in LOOCV blindBLAST searcing, y is the density of the cases among all query cases in the cluster. The unmatched cases are enriched in the low number regions for clusters H1-13-1, while H2-10-1, L2-8-1 and L3-9-2 do not show such a trend.

III. blindBLAST classification accuracy on class member size unbalanced dataset

A machine learning classifier is introduced to improve the classification accuracy. The accuracies from the machine learning classifier and the blndBLAST are compared. The evaluation for both methods are conducted under cross validation scheme of 3 repeats and 10 fold cross validation.

a). blindBLAST cluster identification Accuracy and how it is affected by member size distribution, cluster number, and overall sample size

To validate the utility of the machine learning model, the following questions should be answered. What are the accuracy of the blindBLAST model for each loop and length type? And how is the incorrect template identification affecting the query vs template RMSDs. Shown in Figure 3-4,

accuracies vary according to the loop and length type. A lot of CDR loop and length types have below 90% mean accuracy. This leads the question of why some loop and length type have better accuracy than the others? For this question, I find characteristics of a cluster member size affect its blindBLAST classification accuracy. Loops with greater clusters number such as H1-13 generally have lower accuracy. Loops with more than one popular cluster are likely to have low accuracy. Loops with sparse case number have low accuracies.

H2-10 have lower accuracy compared to H1-13 likely because its cluster two is more balanced to cluster one, therefore increasing chances of misclassification associated with cluster 2. It is the same case between clusters L1-11 and H2-10 with L1-11 more balanced in its member size. The ones with the worst accuracies are L3-10, L3-12, L3-8 and L1-12, each of their cluster one is not the solely dominant cluster as well. Another factor is the small total member size of the loop. Loops L3-10, L3-12, L3-8 and L1-12 are loops with sparse member. This observation however does not apply to L1-13 and L1-14 with L1-13 having accuracy greater than 0.95.

Figure 3-5 shows that substantial increases in mean RMSDs between query and template CDRs can be observed in the set of CDRs with template from wrong clusters in the 3 repeat 10 fold scheme versus if they are aligned with the most similar CDR sequenced template in their own cluster. The misclassifications generally lead to a worse query to template RMSD and the most substantial ones are in the H1-13, H2-10, L1-11, L1-12, L2-8, L3-9. Therefore improving the accuracy of cluster membership of templates can improve the quality of template structure nontrivially.

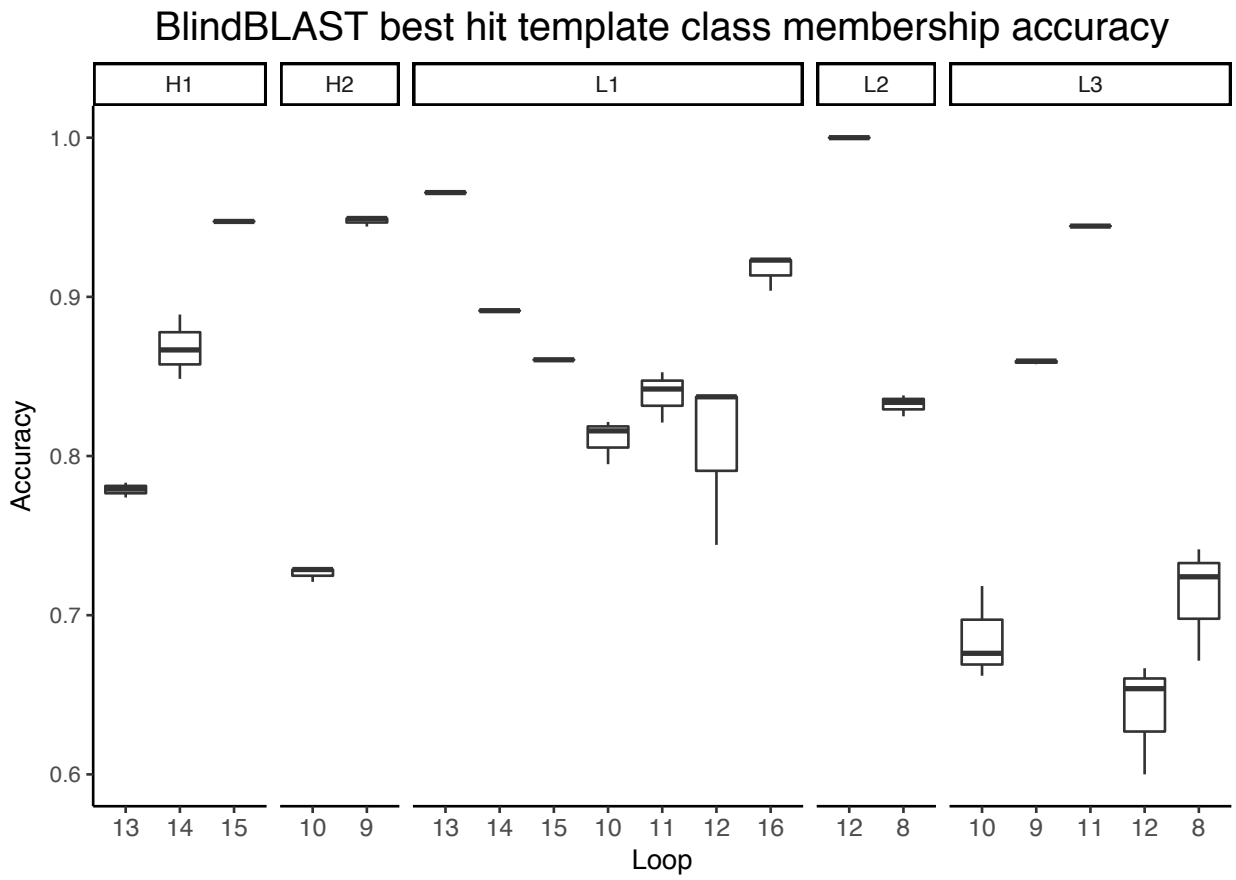


Figure 3-4. Per loop type blindBLAST cluster identification accuracy in 3-repeats-10-fold cross-validation:
Each box is ploted based on 3 accuracy values each from the result of a 10-fold cross validation. The x axis denotes the loop length. These values each combine with the upper panel to indicate one specific loop type.

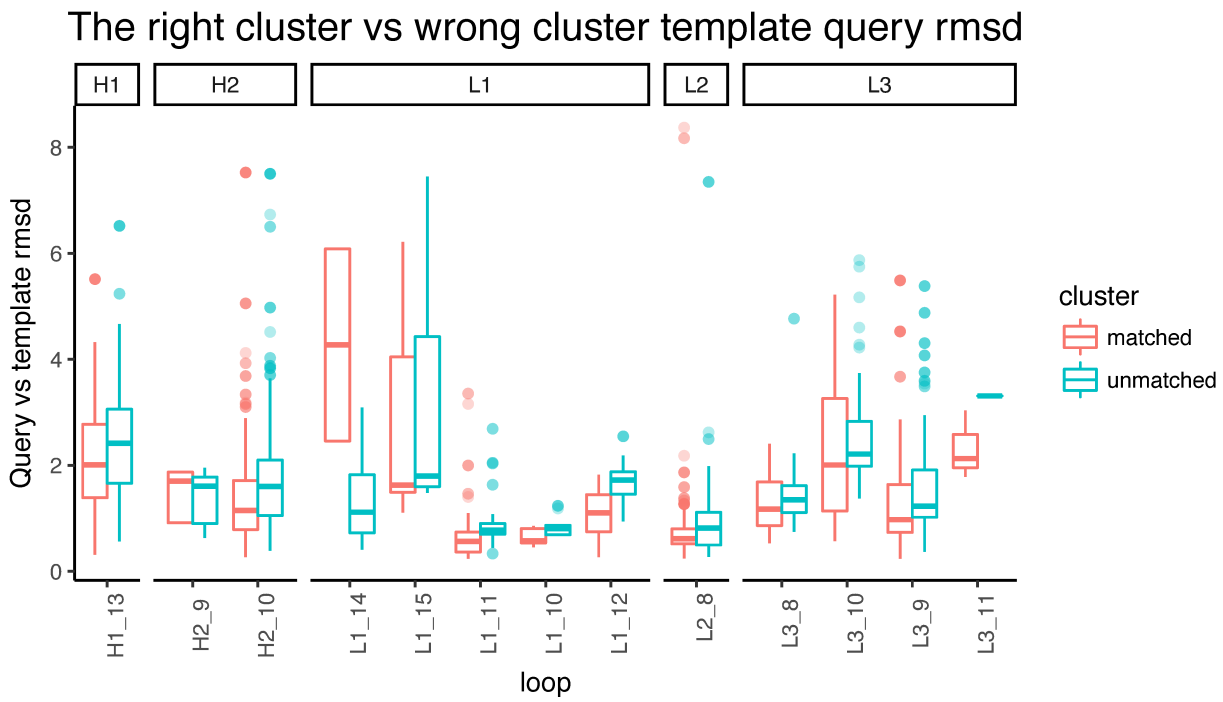


Figure 3-5. The right cluster vs wrong cluster query-template RMSDs:
RMSDs of the blindBLAST failed to predict query-template set and RMSDs of the same set but with templates selected from the right structure cluster.The larger standard deviation of accuracies of loop L1_14, L1_15, L3_10 results from the lower case numbers in these CDR loops.

IV. The guidedBlast achieves higher accuracy in predicting cluster membership from query CDR sequence

There are three questions sought to be answered. Which is the best GBM model for each loop. What are the average accuracy of the left out sets of the trained model? And how is the accuracy from the GBM model compared to that of the original blindBLAST method? are necessary for comparing between the The best GBM model for each loop should be be used for improving cluster identification accuracy compared to blindBLAST, two questions should be thought to answer. One is how to find the best model the training method can acheive. The second are the specific accuracy improvement for each of the misclassification. As described in the methods section. GBM models are tuned to get the best set of hyper-parameters for each loop and length type. The accuracies on all left-out folds in the CV and its corresponding complexity parameters are shown in Figure 3-6 . The tuned set of parameters differs for each CDR loop and length type as listed in Table 3-6.

The hyper-parameter tuning plot Figure 3-7 shows once a specific model complexity is reached for a loop and length type, the model prediction accuracy is plateaued or become worse even with higher model complexity. The tuned models can be divided into two groups based on whether the prediction accuracy is greater than that of blindBLAST and its prediction model stability. In the worse performing group, some loops L1-13, L1-14, L1-15 and L3-10 are plagued by model instability because the standard deviation of accuracies are greater than 5% and even 20% for L1-15 due to the sparsity of the data. This stability problem makes the usefulness of these models to be non-conclusive, although L3-10 has more than 10% accuracy improvement with only a small number of boosting iterations (100 trees) and both L3-11, L3-8 have some modest improvements. Another loop in this group is H2-9 as it has worse than blindBLAST accuracy even at high model complexity.

The remaining loops have more promising results, besides having standard deviation of model accuracies below 5%, compared to blindBLAST, H1-13, H2-10, L3-9, L1-16 all achieve greater accuracies, although by small numbers except for H2_10 which has about 8% improvement. H1-13 takes a greater number of iteration to plateau than H2-10, and L3-9 takes an even smaller number of boosting iterations. H2-10 achieves the second largest accuracy improvement over all the loops except for L3-10.

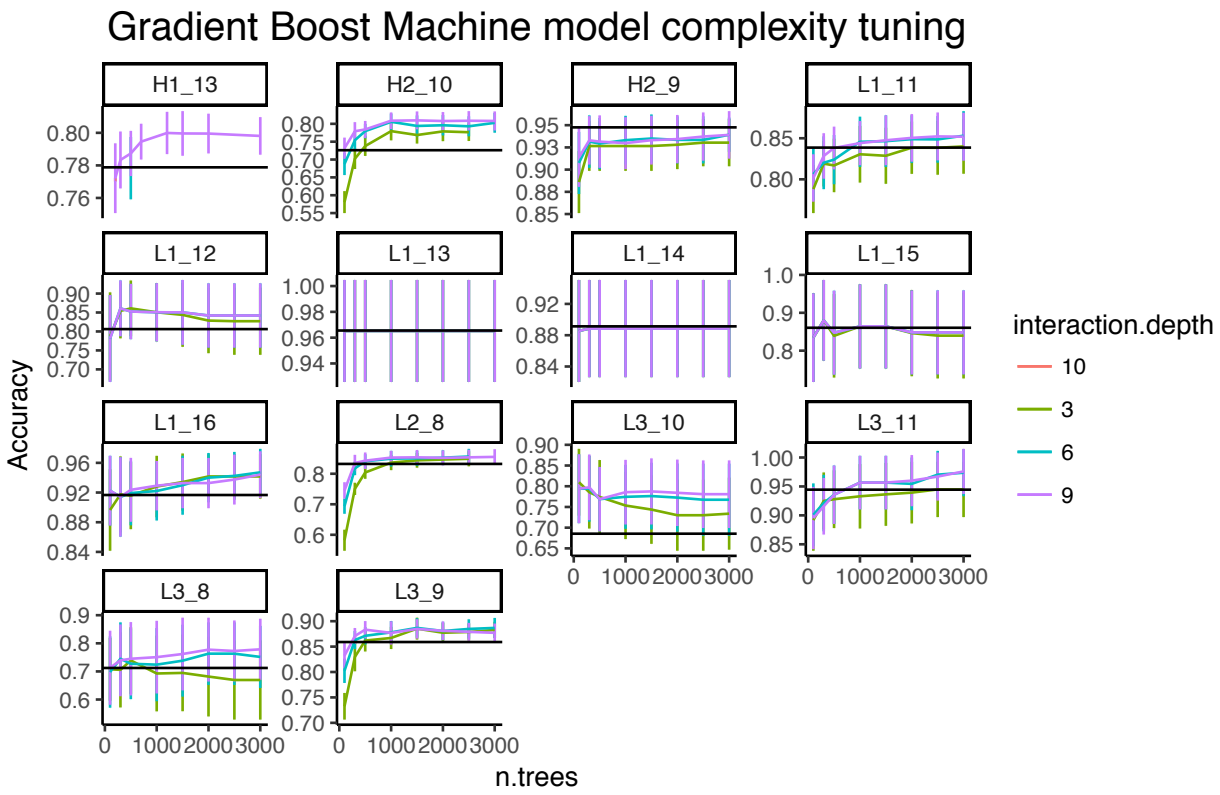


Figure 3-6. Gradient Boost Machine model complexity tuning.
The accuracies on y axis is the average accuracy of the 3 repeats 10 fold cross-validation. Points correspond to different model complexities of the GBM. The parameters that varies include boosting iterations(n.trees), single weak learner complexity(interaction.depth). The shrinkage not shown is set to be $\max(0.01, 0.1*\min(1, nl/10000))$ depending on the case number of the training set nl for each loop and length type. Each panel correspond to the performance of a single CDR loop. Compared to the performance of blindBLAST, the best model achieves higher mean accuracy and generally with a lesser model variance.

	interaction.depth	n.trees	shrinkage	n.minobsinnode	logLoss	Accuracy
H1_13	9	1200	0.01	3	0.873	0.800
H2_10	9	1500	0.01	5	1.372	0.810
H2_9	6	3000	0.01	5	1.718	0.939
L1_11	6	3000	0.01	5	1.404	0.853
L1_12	6	300	0.01	5	0.424	0.861
L1_13	3	100	0.01	5	0.272	0.965
L1_14	3	300	0.01	5	0.475	0.888
L1_15	9	2000	0.001	2	0.131	0.905
L1_16	6	3000	0.01	5	-0.438	0.947
L2_8	6	2500	0.01	5	1.315	0.856
L3_10	9	2000	0.001	5	0.645	0.820
L3_11	9	3000	0.01	5	0.528	0.976
L3_8	9	3000	0.01	5	4.241	0.779
L3_9	6	1500	0.01	5	0.701	0.887

Table 3-6. The finally tuned parameters with the average accuracy of the trained models by CV.

b). Improvement on cis-related classification and its the model variable importance

Because we are interest in distinguishing between cis and trans proline conformation, how is blindBLAST compared to the trained GBM model performance in these misclassifications is studied. Figure 3-7 shows that the GBM rescued most of Proline cis-trans conformation associated blindBLAST mis-classifications in L3-10 and H1-13. But GBM is limited in its power in distinguishing L3-9-cis7-1 and L3-9-2, despite reduce the error count to half of the misclassification case number in the blindBLASTresult. This is because 15 out of the 27 blindBLAST misclassified cases has non-Proline 7th positions, therefore these rescued cases can simply be achieved by setting a filtering rule to prevent all queries with non-Proline 7^{th} positions to be classified to the cis-structural clusters, and as there's no 7^{th} Proline in L3-9-1, they are likely to find templates in L3-9-2, the only other non-cis cluster in L3-9. However the L3-10 has a quite different story, besides some less significant rescues in misclassification types labeled as 1_cis7,8-1, 1_cis8-1, none_cis7,8-1 and none_cis8-1, the model can almost completely distinguish cis8-1 from cis7,8-1 despite they both have Proline at 7^{th} and 8^{th} positions as indicated in Figure 3-7. Those less significant misclassifications are so because the L3-10-1 do not have CDRs with 8^{th} Proline and Seq Logo plot Figure 3-10 shows the L3-10-none cases mostly do not have Proline at 8^{th} position, therefore it's easy to set filtering rules.

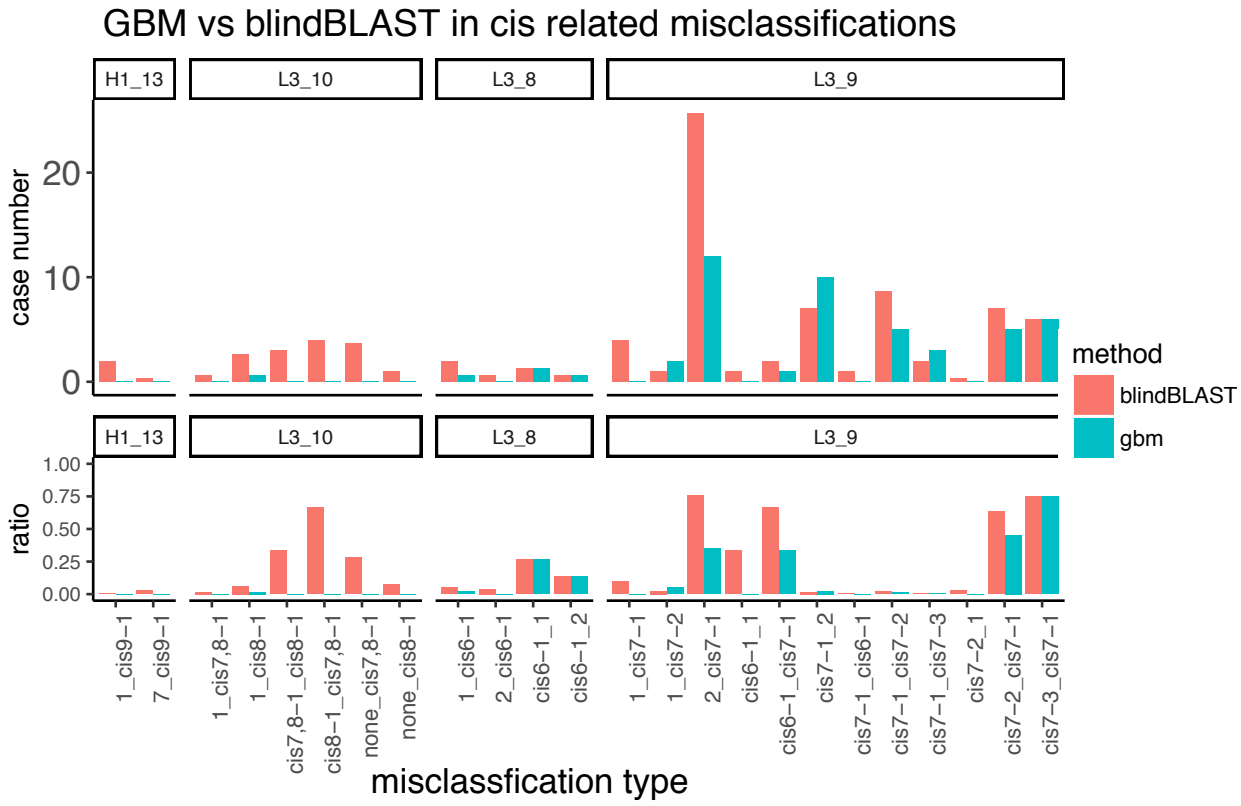


Figure 3-7. GBM vs blindBLAST performace in misclassifications involving cis conformation: The x axis listed all cis conformation related misclassification types which appear mostly in loops H1-13, L3-10, L3-8, L3-9, the height of barplot is the error count in each misclassification in the 10 fold 3 repeats blindBLAST prediction test averaged by repeats.

c). Other non-cis clusters related classification accuracy improvements

So how are the performance of the methods compared besides the discriminating the cis-proline clusters and trans-proline clusters? The cis-cluster related misclassifications constitute most of the improved cases in L3-9 and L3-10. GBM improved misclassifications with over 3 count improvement or compromised misclassification with over 3 count error count increase are shown in Figure 3-9. Many of the improved ones are present in loop H2-10 with no compromised ones from this loop. The best trained model therefore showed better discriminative capability especially between cluster 1 and 6, likewise between cluster 2 and 6.These improvement can't be achieved by setting simple sequence rules. The blindBLAST error cases in misclassification "1-6" have the H2-10-1 query sequences with 7th Glycine, but Glycine is also the dominate 7th residue of H2-10-6. No other residues exist as the dominant residue identity in one cluster but not the other (Figure 3-11). The other improved misclassifications

include less noticeable improvement in L1-11 and the 2_1 misclassification in L2-8. H1_13 by GBM have about 15 less error cases compared to the blindBLAST 166.3 error counts , Figure 5-4 shows discernible recovery improvement for cluster 1, 3, 4 and 5. Figure 5-3 indicates that this improvement comes from many small error count decrease in different misclassifications, within which only misclassification 4_1 have greater than 3 error count decrease from blindBLAST (>10 error count in blindBLAST) misclassification type. An increase of error count greater than 3 is also observed in 6_3 misclassification for H1-13(Figure 3-9).

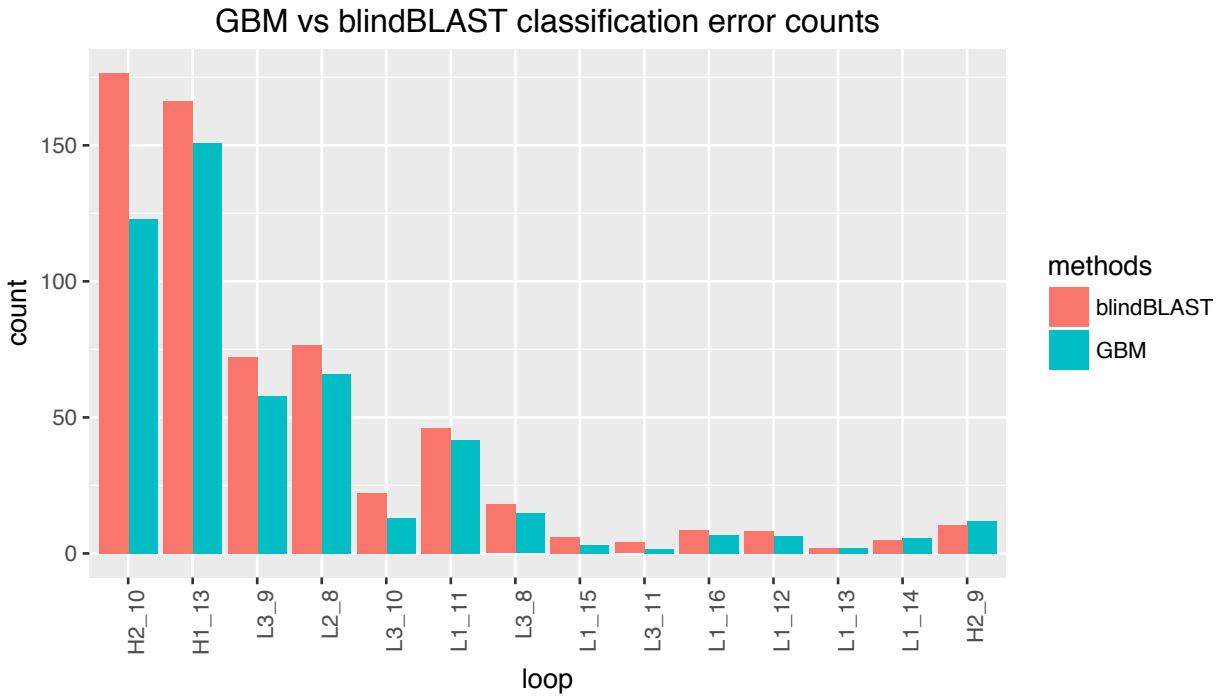


Figure 3-8. Error count of averaged 10-fold CV of blindBLAST and GBM by CDR loop: The misclassification counts for each left-out fold of the 10-folds-CV are collected as the selected template does not belong to the correct cluster or predicted cluster does not agree with the correct cluster. Then the results for all 3 repeats are averaged. The misclassification counts for blindBLAST and GBM are plotted for comparison, sorted on the x-axis by the order of improved classification counts from using GBM instead of blindBLAST. Besides the improved loops, GBM model make two loops worse but by just a few error counts.

Figure 3-9. GBM improved or compromised misclassifications with other 3 error count difference.

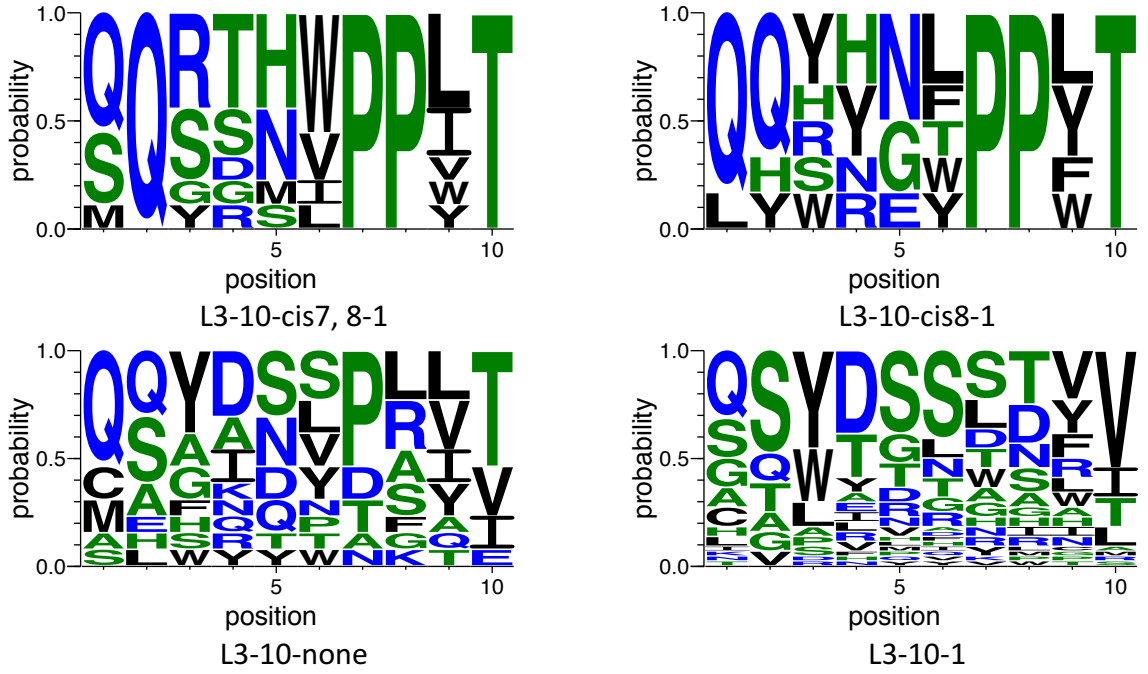
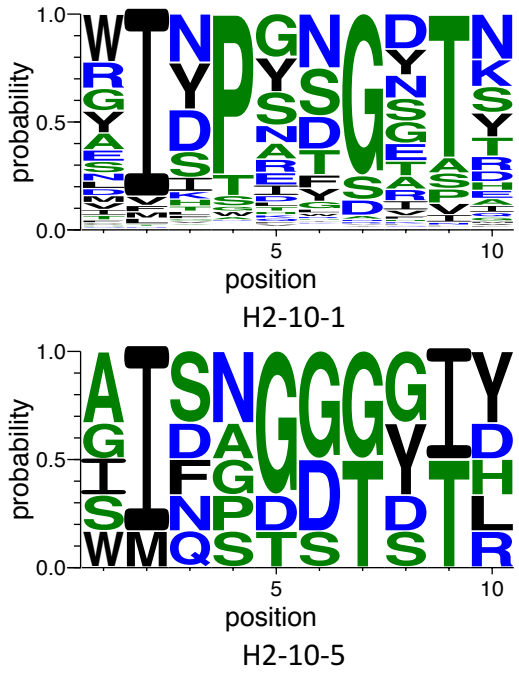


Figure 3-10. Seq logo of the samples in different clusters of L3-10.



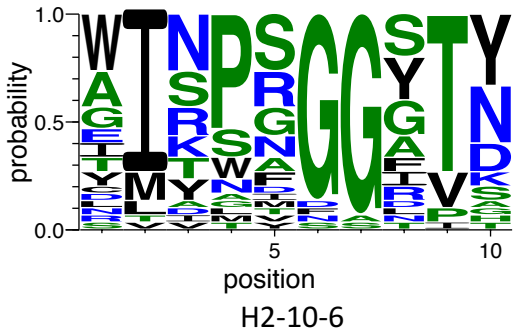
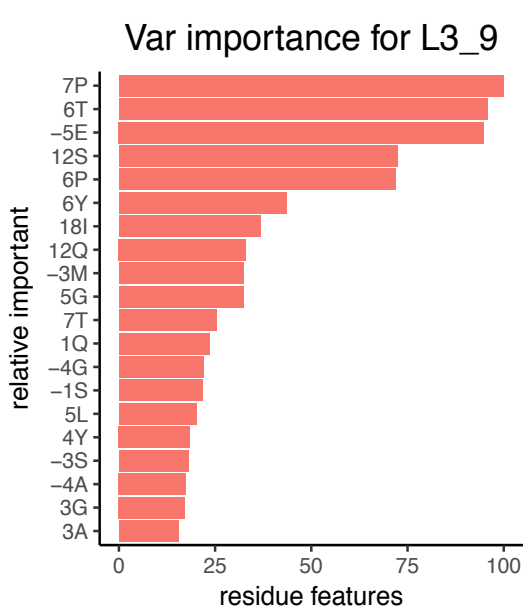
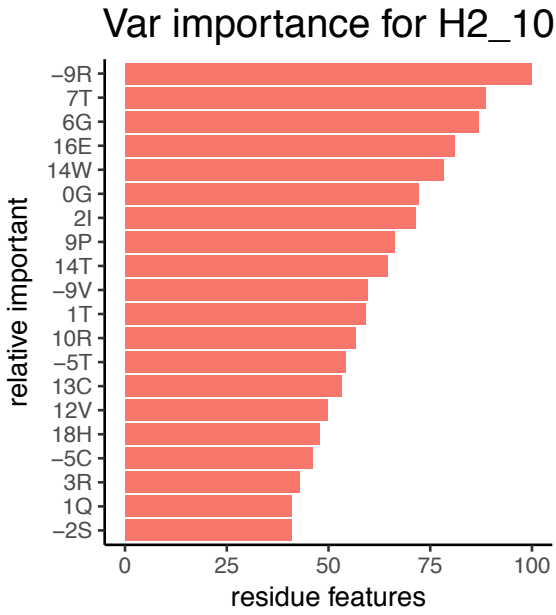


Figure 3-11. Seq logo of samples in different clusters of H2-10.





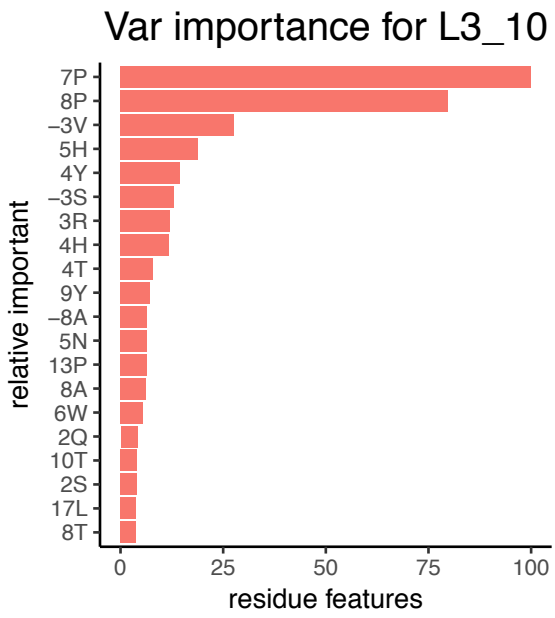


Figure 3-12. Variable importance plots for the model with the best tuned parameters set of different loop and length types：
The y axis is the label of the variable, for example, 7P indicates the importance of the presence of Proline the the 7th position in the loop. -3V indicates the Valine in the 4th upstream position of the loop, taking the direction reversing the order of antibody numbering as the upstream positon. The 11th position is the 1st position of downstream of L3_10.

V. Compare the method to FREAD and Disgro.

I compared the new method with other methodical approaches such as the homology modeling method FREAD and the ab-initio method Disgro using the antibodies from AMAII. The result in Figure 3-13shows that both GBM and blindBLAST find better templates in terms of query vs template RMSD when compared to FREAD and Disgro, and GBM guided template searching rescued the template from the wrong clusters to the right clusters in 2 cases. The FREAD result is in agreement with one of the FREAD paper showing it does not surpass Rosetta Antibody in its modeling result in L1-L3, H1-H2. Therefore blindBLAST still can serve as a good homology modeling template searching strategy despite its simplicity, and GBM-guidedBLAST can correct non-H3 CDR template selection from the wrong canonical cluster .

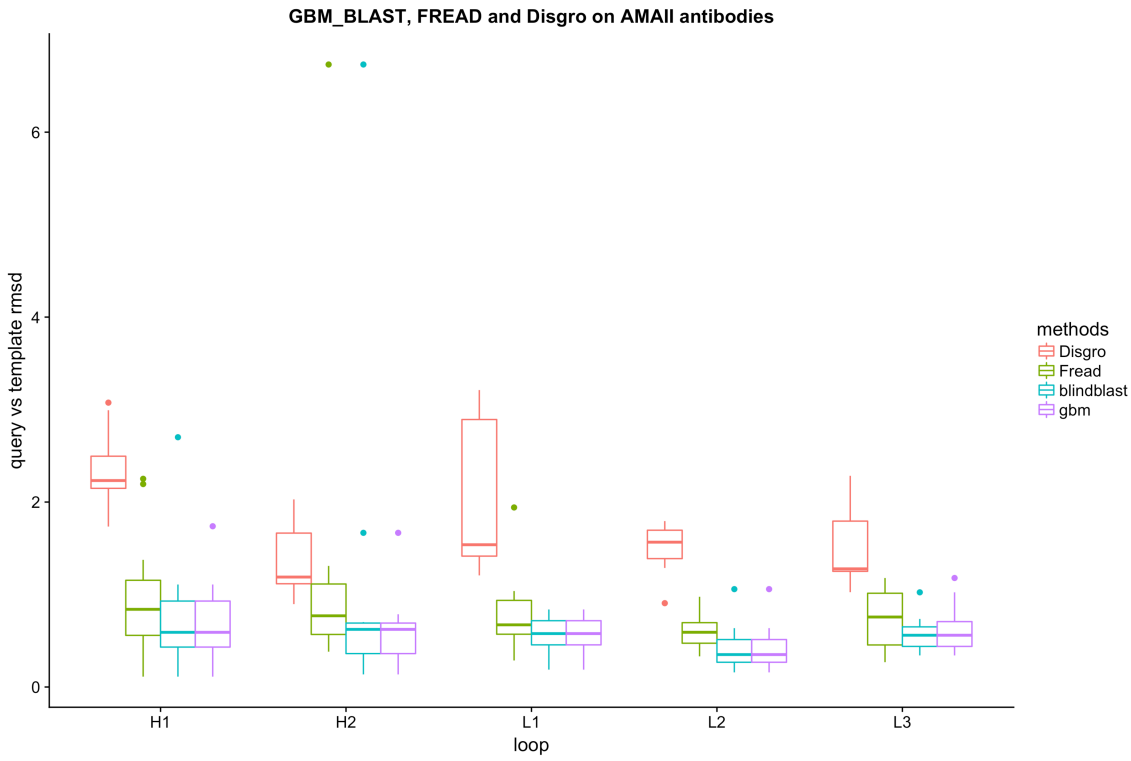


Figure 3-13. GBM, FREAD and Disgro on AMAII antibodies：
The RMSDs of the resulted templates or Disgro-generated models and the query structure are plotted for comparing the performance of the methods. GBM-BLAST searched templates having smaller mean RMSD compared to queries than FREAD searched templates and Disgro generated models. Disgro result is from the scored most confident structure selected from 5000 models generated. Fread is run for each query against the remaining antibody database and select the template giving the best confidence score, which is the similarity score of position specific substitution matrix. As GBM is using blindBLAST by constraining the search databse within its predicted clusters, it may not change the template from blindblast for some queries, the the blindblast query vs template RMSDs are also shown. GBM-BLAST does not affect the template selected in the majority of query cdrs, but it did rescue some cdrs from poor templates, as indicated by one H1 and one H2 cdr, however it gives one template with slightly worse RMSD compared to blindBLAST in L3.

Chapter IV: Discussion:

I. Advantages of GBM:

As stated in Results, some of the best improvements in terms of rescued cases counts come from better accuracies in cis-related clusters of CDR L3 length 9 and 10 loops. The improvement in L3-9 cis-

related clusters can be equally achieved by setting a filter against clustering CDR loops without a proline at the 7th residue position into the L3-9-cis7 clusters. The improvement in the L3-10, especially between its cis7 and cis7,8, was not achievable by simply setting a proline filter as the 7th and 8th positions in both clusters are filled by prolines. Finally, the most prominent improvement of guidedBLAST over blindBLAST was observed in H2-10, reducing misclassification between cluster pairs H2-10-1 and H2-10-6 and H2-10-2 and H2-10-6. These two misclassifications are improved greatly by guidedBLAST because their member sizes are more balanced, relative to the most popular cluster H2-10-1, and their absolute sizes are not as sparse as other loop clusters.

While GBM has improved some of the blindBLAST misclassification cases with error counts no better than random assignment, it has not improved all such misclassifications. For examples, L2-8-2 to L2-8-1, H2-10-1 to H2-10-3, and H1-13-4 to H1-13-1 misclassifications are not improved. The minority clusters (H2-10-3, H1-13-4, L2-8-2) in these cluster pairs are very imbalanced with respect to the majority cluster can be the reason for lack of accuracy improvement. Besides the data distribution problem, the failure to distinguish L2-8-1 and L2-8-2 is partly due to the very small dihedral distance between cluster exemplars, i.e. the clusters are very similar. This raises the question if it is necessary to distinguish these two structural clusters during template searching.

These results suggest that a limiting factor in further improving the identification accuracy by GBM is the unbalanced member size of the minority clusters compared to the size of the majority cluster once the majority cluster is relatively populated (H1-13, H2-10, L2-8, L3-9). Therefore, it could be suggested that as more antibody structures are resolved to enrich some of the minority clusters, the misclassifications associated with those clusters can also be greatly improved as the cases of H2-10-1 and H2-10-6. Moreover, a fine-grained classification might classify structures to its most similar canonical structure, but the small difference between structural clusters might not be overcome even if the minority clusters become more populated.

GBM-trained classifier is a good step to be incorporated into the template selection in non-H3 CDR homology modeling because of the three points: the overall better accuracy achieved; the fact that such improvement could not all be replaced by filtering rules, and that further enrichment in the minority clusters could further improve the accuracy.

II. Future direction:

The limit in the current accuracy partly lies in the data sparsity of clusters with small member size, this limit can be pushed with generating synthetic data for such clusters. A set of structures can be generated that lie in the cluster radius constraint, and Rosetta design can be used to generate the synthetic CDR sequences to emulate the SMOTE method. Another approach of incorporating the antibodies without solved structures in to the model with un-supervised learning can also be attempted, so that the number of cases in the unpopular clusters can be increased. On the other hand, another tweak in the sampling and learning process can be done besides generating synthetic data for sparse cluster, which is building weak learner each time by under-sampling the more popular cluster to obtain cluster distribution balance, which is stated to be a more promising method rather than oversampling the less popular cluster[13].

Chapter V: Supplementary:

The mean dihedral angle is calculated for each dihedral site in each structural cluster. The mean is obtained by finding the value that minimize the variance of all the dihedral angles at a position with this value being its mean. The standard deviation is calculated from the minimized variance. The equation is:

$$\theta_i < -\min\ \sigma^2(\theta_i)\ \ |\ \sigma^2 = \sum_{i=1}^{n_{member}} \min\left(abs(\mu - \theta_i), 360 - abs(\mu - \theta_i)\right)^2 \quad (6)$$

$$sd(abs(\mu - \theta_i), 360 - \mathrm{abs}(\mu - \theta_i)) \quad (7)$$

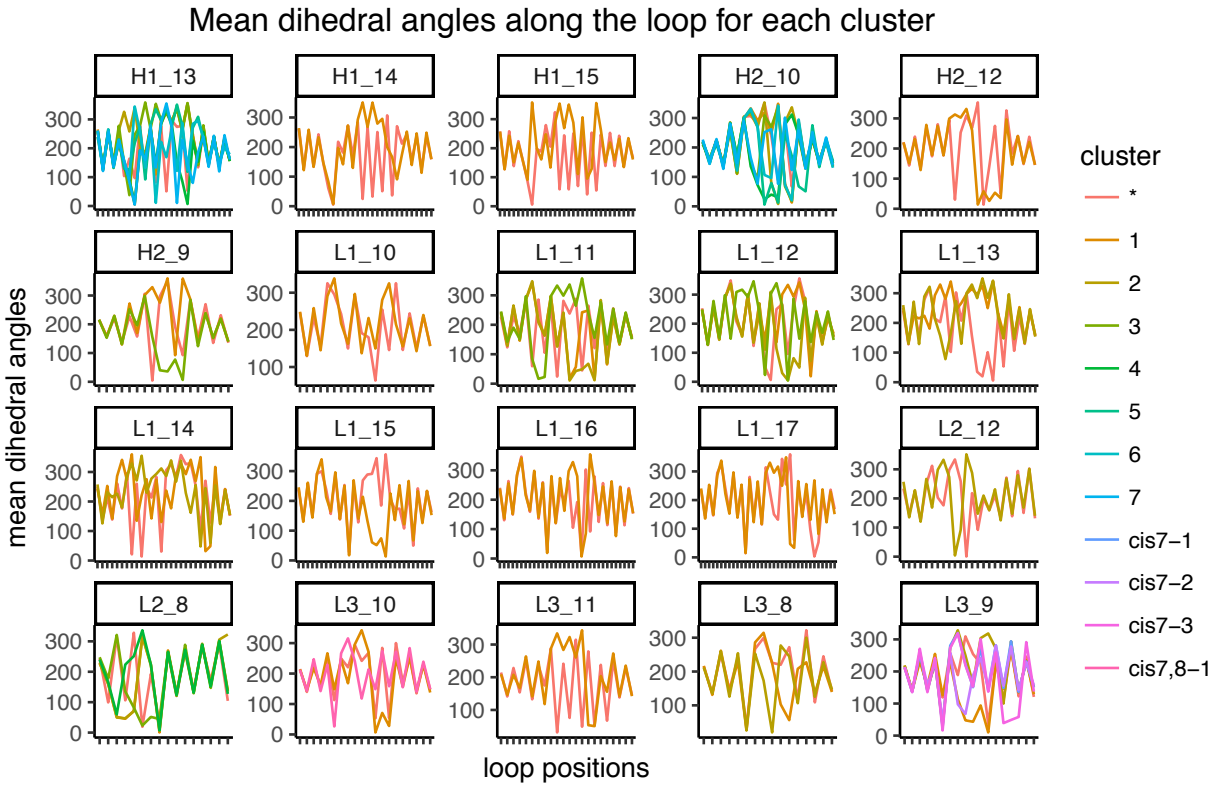


Figure 5-1. Mean dihedral angle on each position of all members in the cluster along the cdr loops:

Clusters exhibit varying degrees of uniformity for the mean dihedral angle for each site. For some sites, the mean dihedral angles are close to each other, while for other sites, one of more clusters differ from each others by more than 20 degrees The plots suggest that the two ends of loops have less variable mean dihedral angles between different clusters than the middle positions of loops.

H1_13	1_phi	2_psi	3_phi	3_psi	4_phi	4_psi	5_phi	5_psi	6_psi	7_phi	7_psi	8_phi	8_psi	9_phi	9_psi	10_phi	10_psi	11_phi	12_phi	12_psi	13_phi	13_psi
H1_14	4_phi	5_phi	5_psi	6_psi	7_psi	8_psi	9_phi	9_psi	10_phi	10_psi	11_phi	11_psi										
H1_15	5_psi	6_phi	6_psi	7_phi	7_psi	8_phi	8_psi	9_phi	9_psi	10_phi	10_psi	11_phi	11_psi	13_phi	13_psi							
H2_10	3_phi	3_psi	4_psi	5_phi	5_psi	6_phi	6_psi	7_phi	7_psi	8_phi	8_psi	9_phi	9_psi	10_psi								
H2_12	5_psi	6_phi	6_psi	7_psi	8_phi	8_psi	9_phi	10_phi	10_psi													
H2_9	3_phi	3_psi	4_psi	5_phi	5_psi	6_phi	6_psi	7_psi	8_phi													
L1_10	2_phi	3_phi	3_psi	4_phi	5_psi	6_phi	6_psi	7_phi	7_psi	8_phi												
L1_11	2_phi	2_psi	3_psi	4_phi	4_psi	5_phi	5_psi	6_phi	6_psi	7_phi	7_psi	8_phi	8_psi	9_psi	10_phi							
L1_12	2_phi	3_psi	4_phi	4_psi	5_psi	6_phi	6_psi	7_phi	7_psi	8_phi	8_psi	9_phi	9_psi	10_phi	10_psi	11_psi	12_phi	12_psi				
L1_13	2_phi	2_psi	3_phi	3_psi	4_phi	4_psi	5_phi	5_psi	6_phi	6_psi	7_psi	8_phi	8_psi	9_phi	9_psi	10_psi	11_phi	11_psi	12_phi			
L1_14	1_psi	2_phi	2_psi	3_phi	3_psi	4_phi	4_psi	5_phi	5_psi	6_phi	6_psi	7_psi	8_phi	8_psi	9_phi	9_psi	10_phi	10_psi	11_psi	12_phi	12_psi	13_phi
L1_15	3_psi	4_phi	8_phi	8_psi	9_phi	9_psi	10_phi	11_psi	12_phi	13_phi												
L1_16	7_phi	8_phi	8_psi	9_phi	9_psi	10_phi	11_phi	11_psi	12_psi													
L1_17	9_psi	10_phi	10_psi	11_phi	11_psi	12_phi	12_psi	13_phi	14_phi	14_psi	15_phi	15_psi	16_phi									
L2_12	1_phi	3_psi	4_phi	4_psi	5_phi	5_psi	6_phi	7_phi	7_psi	9_phi												
L2_8	1_phi	1_psi	2_phi	2_psi	3_phi	3_psi	4_phi	4_psi	6_phi	7_psi	8_psi											
L3_10	2_phi	3_phi	3_psi	4_phi	4_psi	5_phi	5_psi	6_phi	6_psi	7_phi	7_psi	8_phi	9_phi									
L3_11	2_phi	4_psi	5_psi	6_phi	6_psi	7_phi	7_psi	8_phi	9_phi	9_psi	10_phi											
L3_8	2_psi	3_phi	4_phi	4_psi	5_phi	5_psi	6_phi	6_psi	7_phi	7_psi	8_phi											
L3_9	2_phi	2_psi	3_phi	3_psi	4_phi	4_psi	5_phi	5_psi	6_phi	6_psi	7_phi	7_psi	8_phi	8_psi	9_phi	9_psi						

Table 5-1. Important sites of loops:
For each loop site, the corresponding mean dihedral angles of clusters of a loop length and type group are calculated, if any angle values from two clusters have difference greater than 20, they are chosen as important sites for classification.

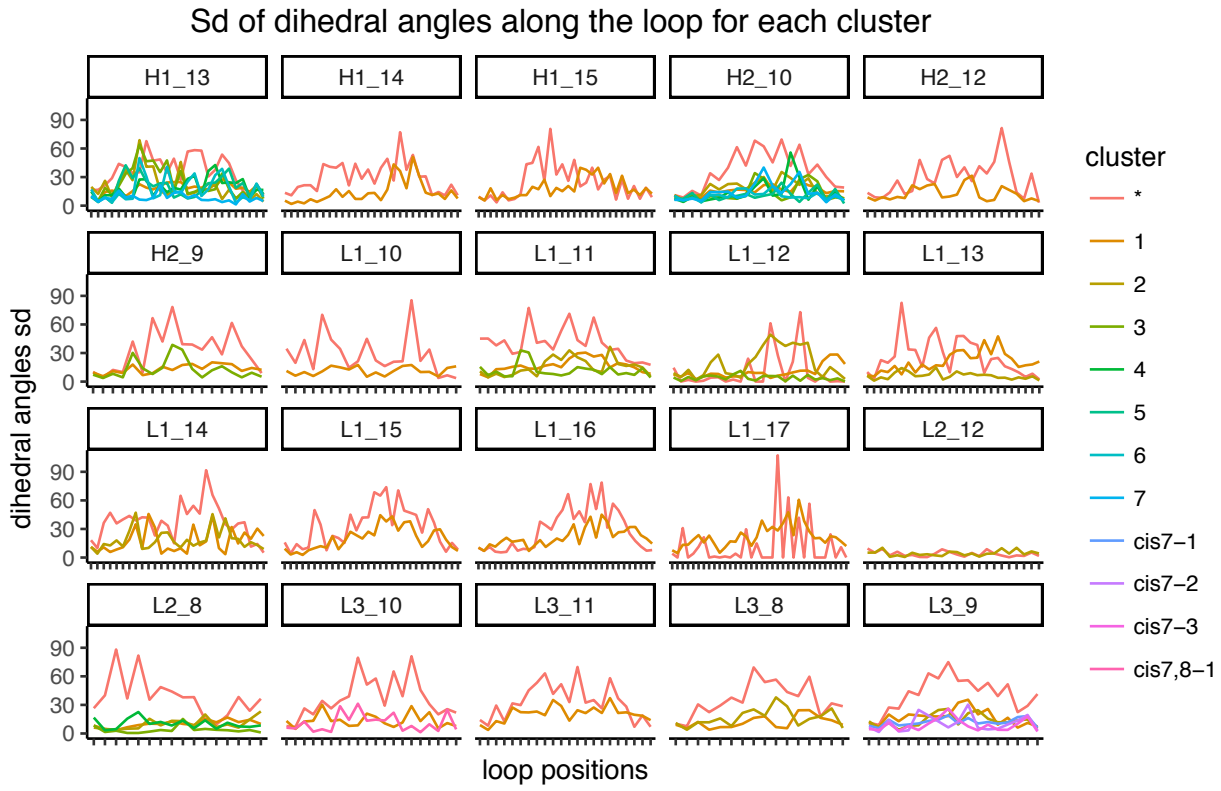


Figure 5-2. the standard deviation of dihedral angle at each position along the loop of all members in the cluster :
Most of the positions have standard deviation below 40 degrees except for the none clusters as they are the fail-to-classified ones. The dihedral angles in the loop center generally have greater sd than the positions near the stem.

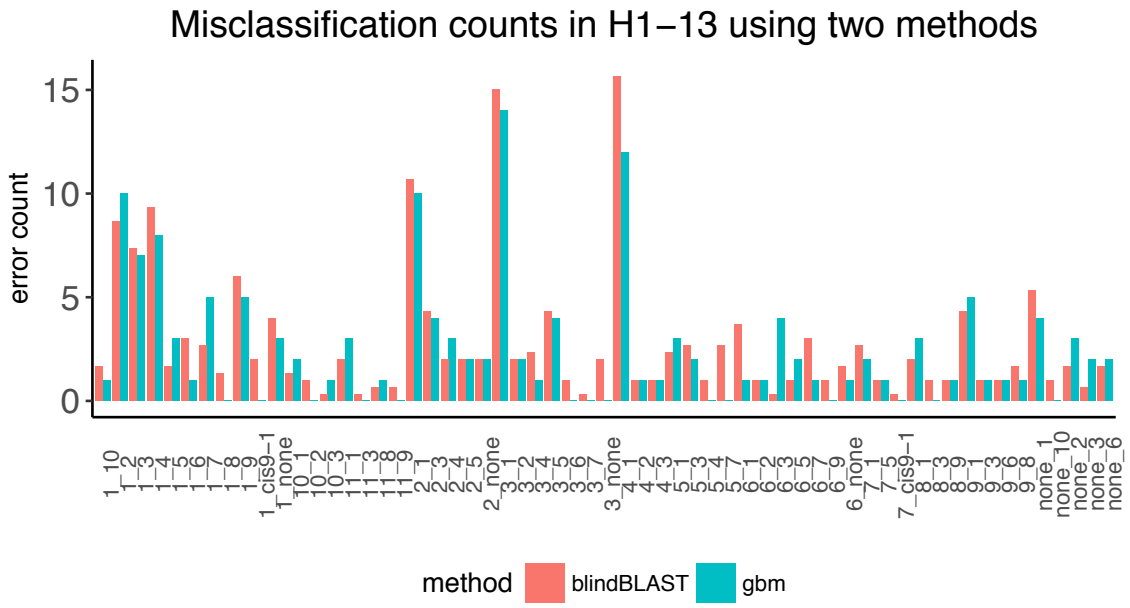


Figure 5-3. Misclassification counts in H1-13:
Most of the misclassifications with larger error counts (>5) do not have obvious decrease in the error counts using GBM trained classification method. The types with decreased error count, however, sometimes have its reverse pair having increased error count using GBM compared to blindBLAST.

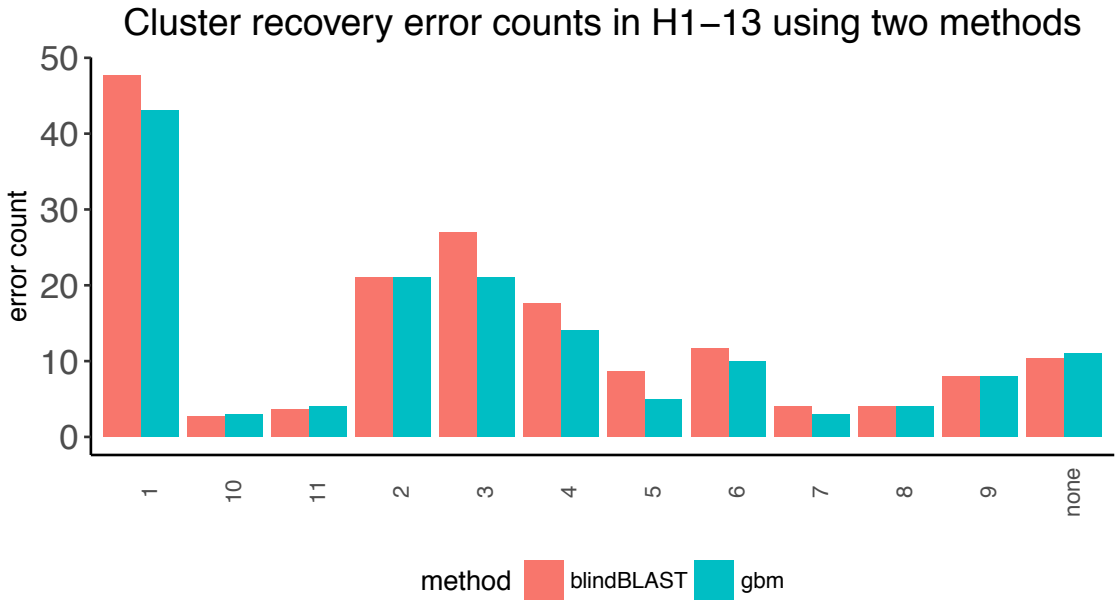


Figure 5-4. The recovery improvement in every cluster in loop H1-13. .

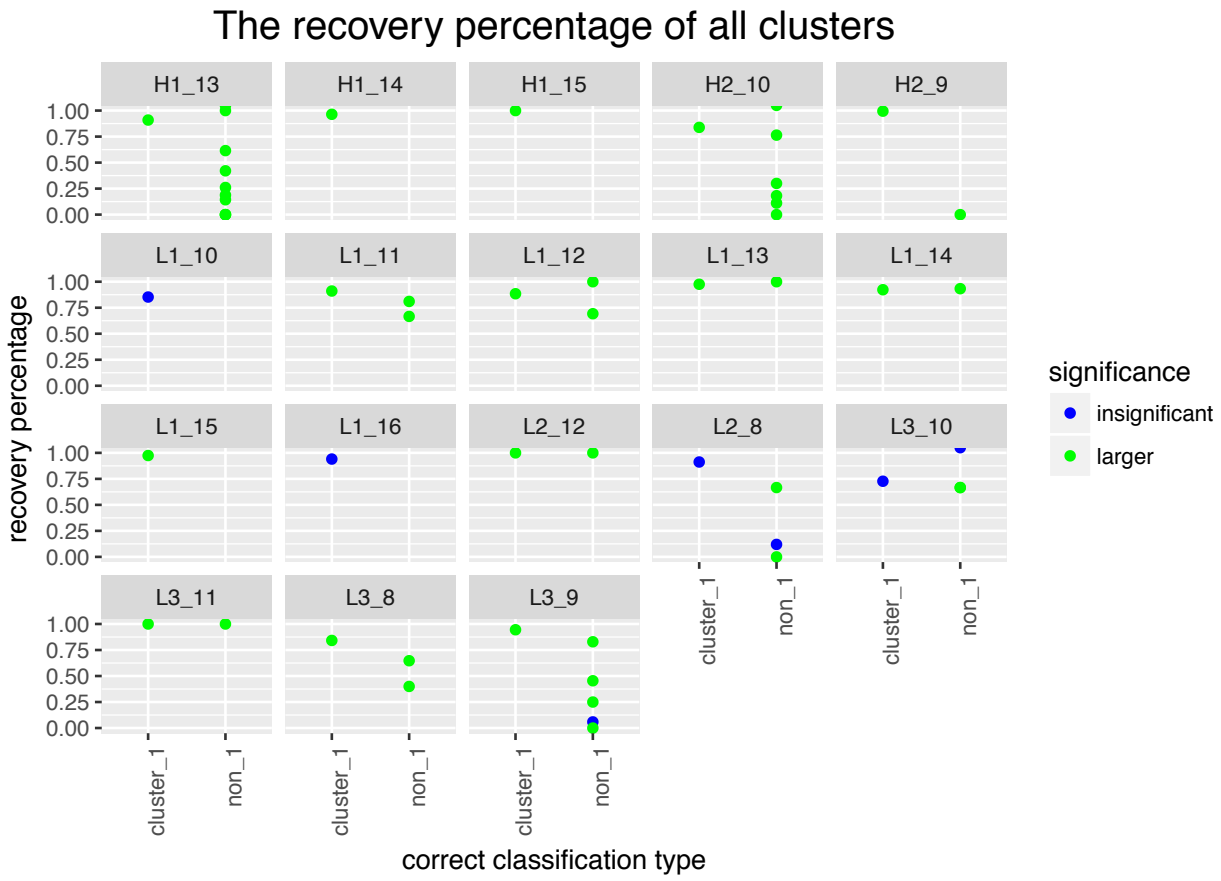


Figure 5-5. The effect size of the correct classification counts:
Improvement by blindBLAST compared to the random assignment simulation. With the point color coded based on which side it falls into the significance test on the empirical correct counts distribution. The “cluster_1” and “non_1” x axis label separate the most popular cluster from other clusters. The y axis is the effect size of the differences.

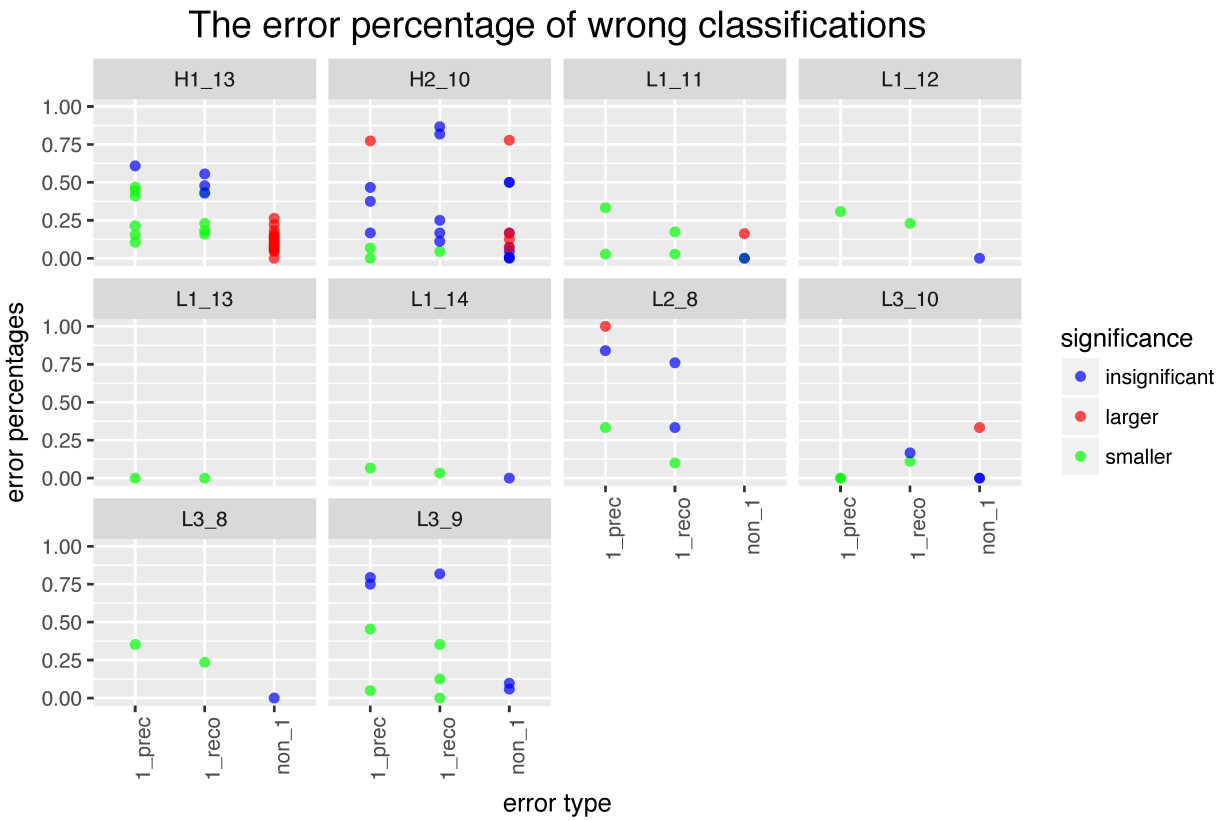


Figure 5-6. The error percentage of the misclassifications:
Plot is generated for ordered cluster pairs with both the query cluster case number and the template cluster case number greater than 5 to eliminate the trivial cases.

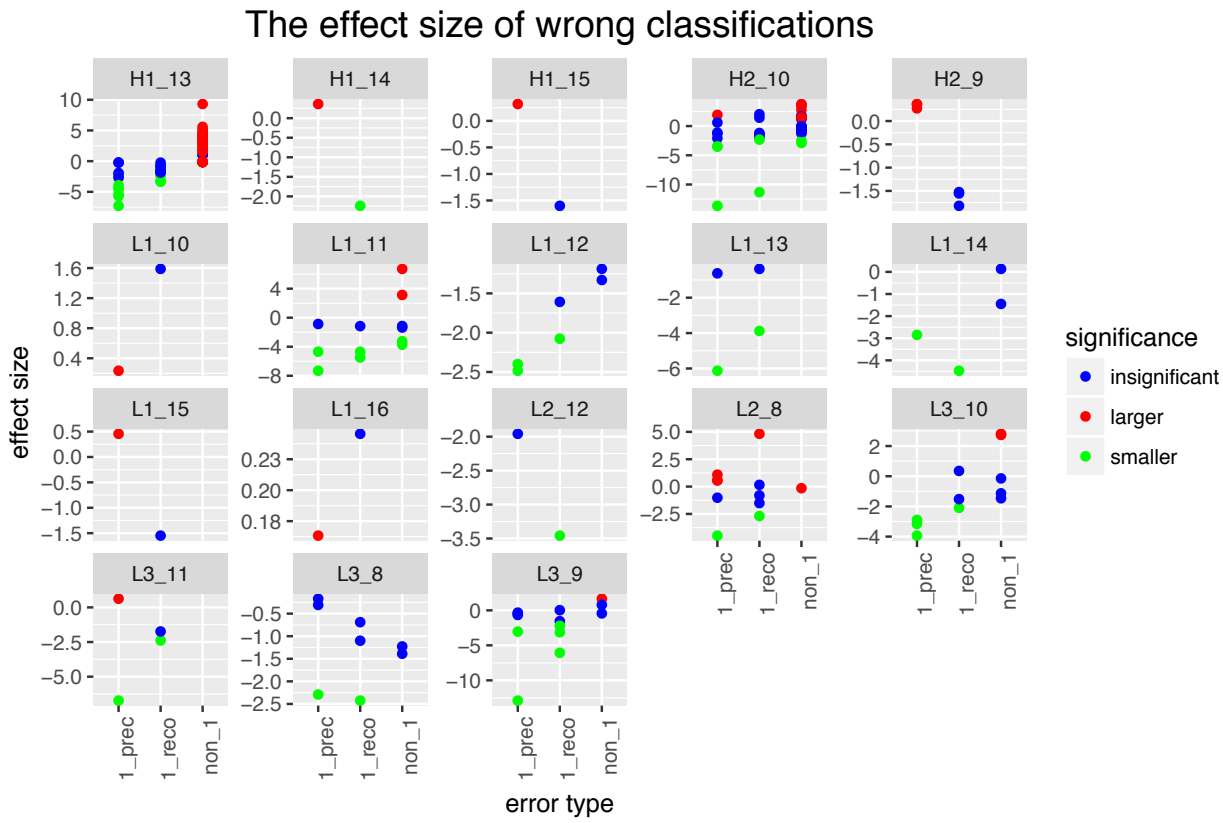


Figure 5-7. The effect size of wrong classifications:
All misclassification types with mean simulated error count greater than one is included in the plot. The x axis separates the misclassifications into three types, “1_prec” are misclassification types that cluster non-cluster-1 into it, “1_reco” are types that cluster cluster-1 to other non cluster-1 clusters. The “non_1” are types of misclassifications between non-cluster-1 clusters.

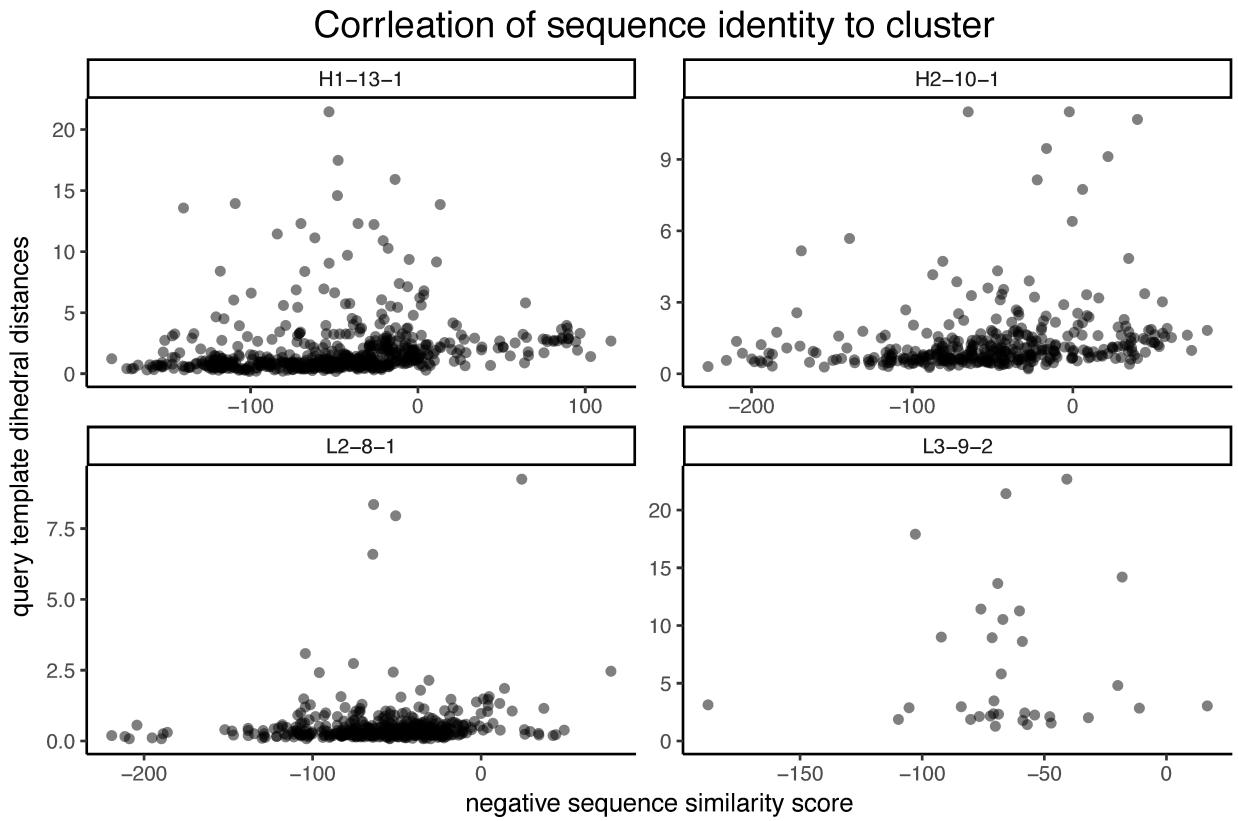


Figure 5-8. Correlation of sequence similarities and dihedral distance:
The x axis is the negative BLAST similarity score, with a smaller score corresponding to greater sequence similarity based on PAM30. The y axis is the dihedral angle distance between a query and template pair. The points include all CDR query and template pairs regardless of whether they match or not.

Bibliography

1. Hirano, M., Das, S., Guo, P. & Cooper, M. D. in *Advances in immunology* **109,** 125–157 (2011).
2. DeKosky, B. J., Lungu, O. I., Park, D., Johnson, E. L., Charab, W., Chrysostomou, C., Kuroda, D., Ellington, A. D., Ippolito, G. C., Gray, J. J. & Georgiou, G. Large-scale sequence and structural comparisons of human naive and antigen-experienced antibody repertoires. *Proc. Natl. Acad. Sci. U. S. A.* **113,** E2636-45 (2016).
3. Glanville, J., Zhai, W., Berka, J., Telman, D., Huerta, G., Mehta, G. R., Ni, I., Mei, L., Sundar, P. D., Day, G. M. R., Cox, D., Rajpal, A., Pons, J. & Lerner, R. A. Precise determination of the diversity of a combinatorial antibody library gives insight into the human immunoglobulin repertoire.
4. Karanicolas, J. & Kuhlman, B. Computational design of affinity and specificity at protein–protein interfaces. *Curr. Opin. Struct. Biol.* **19,** 458–463 (2009).
5. Bradbury, A. & Plückthun, A. Reproducibility: Standardize antibodies used in research. *Nature* **518,** 27–29 (2015).
6. Weiser, M., Vega-Saenz de Miera, E., Kentros, C., Moreno, H., Franzen, L., Hillman, D., Baker, H. & Rudy, B. Differential expression of Shaw-related K+ channels in the rat central nervous system. *J. Neurosci.* **14,** 949–72 (1994).
7. Wojciak, J. M., Zhu, N., Schuerenberg, K. T., Moreno, K., Shestowsky, W. S., Hiraiwa, M., Sabbadini, R. & Huxford, T. The crystal structure of sphingosine-1-phosphate in complex with a Fab fragment reveals metal bridging of an antibody and its antigen. *Proc. Natl. Acad. Sci. U. S. A.* **106,** 17717–22 (2009).
8. He, K., Du, X., Sheng, W., Zhou, X., Wang, J. & Wang, S. Crystal Structure of the Fab Fragment of an Anti-ofloxacin Antibody and Exploration of Its Specific Binding. *J. Agric. Food Chem.* **64,** 2627–2634 (2016).
9. Lippow, S. M., Wittrup, K. D. & Tidor, B. Computational design of antibody-affinity improvement beyond in vivo maturation. *Nat. Biotechnol.* **25,** 1171–6 (2007).
10. Dunbar, J., Krawczyk, K., Leem, J., Marks, C., Nowak, J., Regep, C., Georges, G., Kelm, S., Popovic, B. & Deane, C. M. SAbPred: a structure-based antibody prediction server. *Nucleic Acids Res.* **44,** W474-8 (2016).
11. Weitzner, B. D., Jeliazkov, J. R., Lyskov, S., Marze, N., Kuroda, D., Frick, R., Adolf-Bryfogle, J., Biswas, N., Dunbrack, R. L. & Gray, J. J. Modeling and docking of antibody structures with Rosetta. *Nat. Protoc.* **12,** 401–416 (2017).
12. Marcatili, P., Olimpieri, P. P., Chailyan, A. & Tramontano, A. Antibody structural modeling with prediction of immunoglobulin structure (PIGS). *Nat. Protoc.* **9,** 2771–2783 (2014).
13. Kuhn, M. & Johnson, K. Applied Predictive Modeling [Hardcover]. (2013). doi:10.1007/978-1-4614-6849-3
14. Baran, D., Pszolla, M. G., Lapidoth, G. D., Norn, C., Dym, O., Unger, T., Albeck, S., Tyka, M. D. & Fleishman, S. J. Principles for computational design of binding antibodies. *Proc. Natl. Acad. Sci. U. S. A.* **114,** 10900–10905 (2017).
15. Choi, Y. & Deane, C. M. Predicting antibody complementarity determining region structures without classificationw. *Mol. BioSyst. Mol. BioSyst* **7,** 3327–3334 (2011).
16. Shi, J., Blundell, T. L. & Mizuguchi, K. FUGUE: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties11Edited by B. Honig. *J. Mol. Biol.* **310,** 243–257 (2001).
17. Yamashita, K., Ikeda, K., Amada, K., Liang, S., Tsuchiya, Y., Nakamura, H., Shirai, H. & Standley, D.

M. Kotai Antibody Builder: automated high-resolution structural modeling of antibodies. *Bioinformatics* **30,** 3279–3280 (2014).
18. Dunbar, J., Krawczyk, K., Leem, J., Baker, T., Fuchs, A., Georges, G., Shi, J. & Deane, C. M. SAbDab: the structural antibody database. *Nucleic Acids Res.* **42,** D1140–D1146 (2014).
19. Lefranc, M.-P., Pommié, C., Kaas, Q., Duprat, E., Bosc, N., Guiraudou, D., Jean, C., Ruiz, M., Da Piédade, I., Rouard, M., Foulquier, E., Thouvenin, V. & Lefranc, G. IMGT unique numbering for immunoglobulin and T cell receptor constant domains and Ig superfamily C-like domains. (2004). doi:10.1016/j.dci.2004.07.003
20. Sequences of proteins of immunological interest /. (U.S. Dept. of Health and Human Services, Public Health Service, National Institutes of Health, 1991).
21. Standard conformations for the canonical structures of immunoglobulins1. *J. Mol. Biol.* **273,** 927–948 (1997).
22. North, B., Lehmann, A. & Dunbrack, R. L. A New Clustering of Antibody CDR Loop Conformations. *J. Mol. Biol.* **406,** 228–256 (2011).
23. Honegger, A., Plu, A. & Ckthun, È. Yet Another Numbering Scheme for Immunoglobulin Variable Domains: An Automatic Modeling and Analysis Tool. doi:10.1006/jmbi.2001.4662
24. Al-Lazikani, B., Lesk, A. M. & Chothia, C. Standard conformations for the canonical structures of immunoglobulins 1 1Edited by I. A. Wilson. *J. Mol. Biol.* **273,** 927–948 (1997).
25. Weitzner, B. D., Kuroda, D., Marze, N., Xu, J. & Gray, J. J. Blind prediction performance of RosettaAntibody 3.0: Grafting, relaxation, kinematic loop modeling, and full CDR optimization. *Proteins Struct. Funct. Bioinforma.* **82,** 1611–1623 (2014).
26. Lorenzen, S., Peters, B., Goede, A., Preissner, R. & Fr??mmel, C. Conservation of cis prolyl bonds in proteins during evolution. *Proteins Struct. Funct. Genet.* **58,** 589–595 (2005).
27. Exarchos, K. P., Papaloukas, C., Exarchos, T. P., Troganis, A. N. & Fotiadis, D. I. Prediction of cis/trans isomerization using feature selection and support vector machines. *J. Biomed. Inform.* **42,** 140–149 (2009).
28. Sarkar, P., Reichman, C., Saleh, T., Birge, R. B. & Kalodimos, C. G. Proline cis-trans Isomerization Controls Autoinhibition of a Signaling Protein. *Mol. Cell* **25,** 413–426 (2007).
29. Kado, Y., Mizohata, E., Nagatoishi, S., Iijima, M., Shinoda, K., Miyafusa, T., Nakayama, T., Yoshizumi, T., Sugiyama, A., Kawamura, T., Lee, Y. H., Matsumura, H., Doi, H., Fujitani, H., Kodama, T., Shibasaki, Y., Tsumoto, K. & Inoue, T. Epiregulin recognition mechanisms by anti-epiregulin antibody 9E5: Structural, functional, and molecular dynamics simulation analyses. *J. Biol. Chem.* **291,** 2319–2330 (2016).
30. Andreotti, A. H. Native state proline isomerization: An intrinsic molecular switch. *Biochemistry* **42,** 9515–9524 (2003).
31. Jain, P., Garibaldi, J. M. & Hirst, J. D. Supervised machine learning algorithms for protein structure classification. *Comput. Biol. Chem.* **33,** 216–223 (2009).
32. Blagus, R., Lusa, L., Bishop, C., He, H., Garcia, E., Daskalaki, S., Kopanas, I., Avouris, N., Ramaswamy, S., Ross, K., Lander, E. et al, S. SMOTE for high-dimensional class-imbalanced data. *BMC Bioinformatics* **14,** 106 (2013).
33. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16,** 321–357 (2002).
34. Adolf-Bryfogle, J., Xu, Q., North, B., Lehmann, A. & Dunbrack, R. L. PyIgClassify: a database of antibody CDR structural classifications. *Nucleic Acids Res.* **43,** D432–D438 (2015).

Curriculum Vitae

EDUCATION

Johns Hopkins University **Dec 2017**
M.S.E. Chemical & Biomolecular Engineering
University of Illinois at Chicago **Aug 2015**
Bioinformatics
Illinois Institute of Technology **May 2013**
BS Biochemistry

RESEARCH EXPERIENCE

Computational protein lab at Johns Hopkins University 05/2016 ~ current
Principle Investigator: Prof. Jeffrey Gray

- Graduate student with research project to improve template selection method for knowledge-based CDR loop modeling in the Rosetta antibody protocol; Contributor to Rosetta software
- Profiled current Rosetta CDR loop prediction accuracy with metric of whether the selected template belongs to the correct CDR loop structural classification clustered by RMSD previously in literature with all current available human and mouse antibodies on Sabdab.
- Attempted per position residue frequency profiling method PSSM, various machine learning methods and energy based method for improving the accuracy of the prediction, recognizing the major difficulty is in allowing clusters with a small number of members to have good recall, while classes with a large number of members maintain good specificity.
- Rewrite python code to c++ for some functions linking Rosetta and pymol for graphical demonstration of the protein such as per residue energy and structural information.

University of Illinois at Chicago and University of Chicago Institute for Genomics and Systems Biology 01/2015 ~ 10/2015
Research Assistant in PsychEncode Project
Principle Investigator: Prof. Chunyu Liu, Prof. Elliot Gershon

- Performed preprocessing for over 200 whole genome RNA sequencing samples to filter out low-quality reads and samples.
- Communicated with wet lab colleagues to optimize the sample extraction protocol so that the sequencing depth is around 50 million reads and with good coverage in intronic and noncoding regions of the genome.
- Confirmed the validity of wet lab protocol for studying non-coding RNA expression by getting the statistics of the assembled RNA transcripts in terms of their length, and distribution of the genome.

University of Illinois at Chicago Bioinformatics Lab **03/2014 ~ 8/2015**

Principle Investigator: Prof. Yang Dai

Graduate student, participated in studying the coupling between methylation and gene isoform expression in cancer

- Constructed a graph model for selecting out isoforms with highly correlated splicing patterns alteration in cancer, the same model could be applied to DNA methylation level data as well.
- Added another layer to the model by coupling the two graphs constructed by isoform expression and methylation, and constructed the coupled heavy subgraph for the coupled graph by optimizing the parameters of the objective function. Such coupling leads to implications of a global effect of methylation of DNA changes alter the splicing pattern during tumorigenesis.
- Applied the coupled model to 13 types of cancer with more than 20 samples for each type of cancer from both healthy and tumor tissues in TCGA.

BGI China **5/2014 ~ 8/2014**

Intern as data analyst in a group developing capturing circulating tumor cells (CTCs)

- Analyzed RNAseq raw data from a set of prostate tumors and a set of healthy individual prostate tissues to characterize the differential expression between the two sets. Compared RNAseq bulk analysis tools cufflinks, MATs, and Htseq for possible adoption to single-cell sequencing analysis.
- Generated differential expression profile of healthy and prostate tumor samples. Selected out the significantly differently expressed genes and cluster the samples into different groups using the subset of differentially expressed genes.

"Muscle" Lab at Illinois Institute of Tech **06/2012~05/2013**

Research volunteer for a project designing an improved exon skipping strategy to restore the reading frame of a mutated dystrophin gene, which leads to Duchene muscular dystrophy.

Principle Investigator: Prof. Nick Menhart

- Performed genetic engineering, protein expression and purification, protease digestion and circular dichroism profiling on some variants among a library of deletion variants of dystrophin with mutation in disease hotspots spanning exons 45-46,45-47, 45-48.; With findings of exon deletion 45-48 results in the most stable repairs as a result of less disrupted helicity of the spliced repeats.
- Helped with assessing neural NO Synthase (nNOS) binding moiety in some variants among a total of constructed 35 variants in the STR1617 region, with each corresponding to a bin of neighboring charged residues substituted by alanine. Studied how the charges in each bin affected the binding activity individually or in combination using a bio-layer interferometer.