

Original Paper

Prediction of Antibody-Antigen Binding via Machine Learning: Development of Data Sets and Evaluation of Methods

Chao Ye¹, MIS; Wenxing Hu², MIT; Bruno Gaeta¹, PhD

¹School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia

²Department of Computer Science, School of Information Science and Technology, Tokyo Institute of Technology, Tokyo, Japan

Corresponding Author:

Bruno Gaeta, PhD

School of Computer Science and Engineering

The University of New South Wales

Computer Science Building (K17)

Engineering Rd, UNSW

Sydney, 2052

Australia

Phone: 61 293857213

Email: bgaeta@unsw.edu.au

Abstract

Background: The mammalian immune system is able to generate antibodies against a huge variety of antigens, including bacteria, viruses, and toxins. The ultradeep DNA sequencing of rearranged immunoglobulin genes has considerable potential in furthering our understanding of the immune response, but it is limited by the lack of a high-throughput, sequence-based method for predicting the antigen(s) that a given immunoglobulin recognizes.

Objective: As a step toward the prediction of antibody-antigen binding from sequence data alone, we aimed to compare a range of machine learning approaches that were applied to a collated data set of antibody-antigen pairs in order to predict antibody-antigen binding from sequence data.

Methods: Data for training and testing were extracted from the Protein Data Bank and the Coronavirus Antibody Database, and additional antibody-antigen pair data were generated by using a molecular docking protocol. Several machine learning methods, including the weighted nearest neighbor method, the nearest neighbor method with the BLOSUM62 matrix, and the random forest method, were applied to the problem.

Results: The final data set contained 1157 antibodies and 57 antigens that were combined in 5041 antibody-antigen pairs. The best performance for the prediction of interactions was obtained by using the nearest neighbor method with the BLOSUM62 matrix, which resulted in around 82% accuracy on the full data set. These results provide a useful frame of reference, as well as protocols and considerations, for machine learning and data set creation in the prediction of antibody-antigen binding.

Conclusions: Several machine learning approaches were compared to predict antibody-antigen interaction from protein sequences. Both the data set (in CSV format) and the machine learning program (coded in Python) are freely available for download on GitHub.

(*JMIR Bioinform Biotech* 2022;3(1):e29404) doi: [10.2196/29404](https://doi.org/10.2196/29404)

KEYWORDS

DNA sequencing; DNA; DNA sequence; sequence data; molecular biology; genomic; random forest; nearest neighbor; immunoglobulin; genetics; antibody-antigen binding; antigen; antibody; structural biology; machine learning; protein modeling; protein; proteomic

Introduction

DNA sequencing technologies are providing new insights into the immune response by allowing for the large-scale sequencing of rearranged immunoglobulin genes that are present in an

individual [1,2]. However, the applications of this approach are limited by the lack of methods for determining the antigen(s) to which a specific immunoglobulin (ie, one encoded by a given sequence) binds. Individual immunoglobulins can be tested experimentally at significant cost; however, the large-scale

characterization of binding properties based on sequence data is currently impossible.

Antigen binding is mediated by the complementarity-determining regions (CDRs) of an antibody, which are shared between heavy and light immunoglobulin chains. Computational methods for predicting antibody-antigen interactions that leverage structure prediction and docking have been proposed [3]. However, the use of these methods requires knowledge of the 3D structures of antibodies and antigens. The direct prediction of antibody-antigen interactions from protein sequences remains an open problem.

Machine learning-based tools, such as mCSM-AB [4] and ADAPT (Assisted Design of Antibody and Protein Therapeutics) [5], have had some success in predicting antibody interactions in other contexts. mCSM-AB is a web server for predicting changes in antibody-antigen affinity upon mutation, using graph-based signatures. ADAPT is an affinity maturation platform that interleaves predictions and testing, and it has been previously validated on monoclonal antibodies.

A more general method for predicting whether an antibody will bind to a protein antigen based on the antibody and antigen sequences remains elusive, in part due to the lack of comprehensive training data for the development of machine learning models. This study is intended as a first step toward this goal and aims to assemble a training data set from a range of sources and evaluate the feasibility of applying machine learning algorithms to identify the binding of antibody-antigen pairs in this data set.

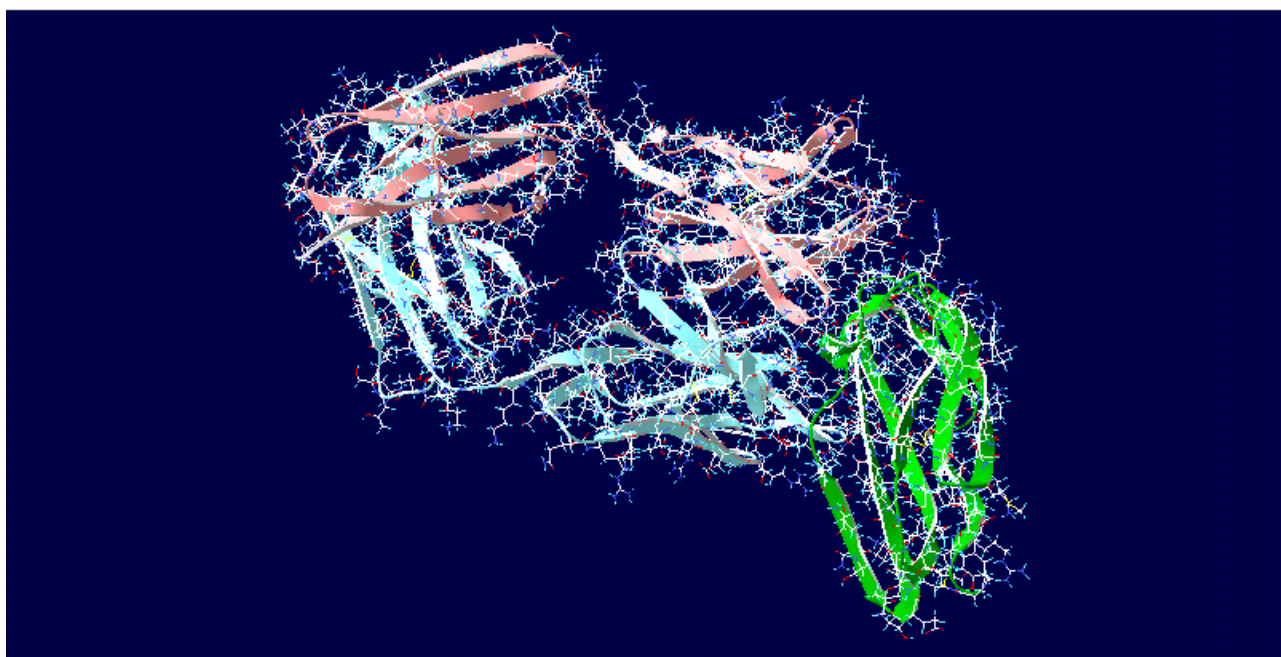
Methods

Data Set

Due to the scarcity of suitable antibody-antigen pairs, computational docking was used to generate some of the data in the training and testing data set. The ClusPro (Boston University) [6-9] and Rosetta (RosettaCommons) [10-12] web servers were used to create a data set of paired antibody-antigen complexes for machine learning. Both ClusPro and Rosetta were used for protein-protein molecular docking. Rosetta uses the SnugDock (RosettaCommons) algorithm [10]. The Swiss-PdbViewer (Swiss Institute of Bioinformatics) [13] was used to examine the resulting protein complex structures.

A total of 50 antibody-antigen complexes were selected randomly from the Protein Data Bank (PDB) [14]. The antibody-antigen complexes were separated by using a Perl script to produce PDB-formatted files as well as sequences for antibodies and antigens. CDRs were located by using the Rosetta antibody modeling web server. Antigens were docked with a range of antibodies by using ClusPro (used only to determine orientation), followed by Rosetta's antibody docking program, SnugDock. In order to keep computation times manageable, not all antibodies were docked. Instead, 10 to 14 antibodies were randomly selected to be docked with each antigen in order to find the best orientation. The resulting complexes were submitted to the Rosetta SnugDock web server in order to calculate the best interface score. This produced structures for between 10 and 14 complexes per antigen, which, when added together with the original antibody-antigen complex, totaled 11 to 15 complexes per antigen. Altogether, 50 antigens were docked with 600 antibodies. An example of a resulting complex is shown in Figure 1.

Figure 1. Example of a docking output. The 3s35 complex was generated by using the ClusPro server (docking results: "YES"; best docking interface score: -0.876).



The Rosetta interface scores were used as estimates of binding affinity in order to identify cognate antibody-antigen pairs to be used as input for machine learning. Complexes with interface scores of higher than -8.0 were classified outright as complexes with poor binding, and those with interface scores of lower than -9.0 were classified outright as complexes with good binding. For complexes with scores that ranged between -8.0 and -9.0 , the docking clusters and positions were examined visually by using SwissDock (Swiss Institute of Bioinformatics). If the top 10 models had their antibodies and antigens in similar relative positions and the structures showed sensible interaction patterns, the pairs were classified as having a good binding affinity.

Rosetta interface scores have been used previously as classifiers to determine binding affinity based on docking results (eg, in an antibody-antigen cross reactivity study [15]).

Additional data were extracted from the Coronavirus Antibody Database (CoV-AbDab) [16]—a database of antibodies against coronaviruses, including SARS-CoV-2, SARS-CoV-1, and MERS-CoV (Middle East respiratory syndrome-related coronavirus). Data (2674 rows) were extracted from the CoV-AbDab on February 14, 2021. After filtering out incomplete data, 2031 rows remained, with each row corresponding to an antibody. The information extracted comprised the antibody names, their binding antigens, and their heavy and light variable region sequences, including the locations of the third CDRs (CDR3s). Each of the variable region sequences were searched against the international ImMunoGeneTics information system database [17] in order to identify the locations of the first CDRs (CDR1s) and second CDRs (CDR2s) from the heavy and light chains. Since a row may contain information about an antibody's interactions with multiple antigens, the data were further split into multiple rows, with each row containing information about the interaction between 1 antibody and 1 antigen.

Additional features were calculated for the sequences, as follows. The isoelectric point for each CDR was calculated by using the Bachem peptide calculator analysis tool (Bachem Holding AG) [18]. The average hydrophilicity of each CDR was also calculated by using the Bachem peptide calculator.

B cell epitopes were predicted by using the IEDB (Immune Epitope Database) antibody epitope prediction analysis tool [19].

The resulting data set can be downloaded from GitHub [20] and is structured with the following column headings: *H chain CDR1 sequence, H chain CDR2 sequence, H chain CDR3 sequence, L chain CDR1 sequence, L chain CDR2 sequence, L chain CDR3 sequence, Hydrophilicity of L CDR1, pI of L CDR1, Hydrophilicity of L CDR2, pI of L CDR2, Hydrophilicity of L CDR3, pI of L CDR3, Hydrophilicity of H CDR1, pI of H CDR1, Hydrophilicity of H CDR2, pI of H CDR2, Hydrophilicity of H CDR3, pI of H CDR3, Antigen Epitope, Rosetta Docking score, Antigen, and Docking result.*

Machine Learning

A weighted K-nearest neighbor (K-NN) classification algorithm [21] for predicting antibody-antigen binding affinity was

implemented in Python. The program can be downloaded from GitHub [20].

For each antigen, the 11 to 15 antibodies that were docked were labeled as “good affinity” or “low affinity,” on the basis of the docking results. Machine learning was then performed, using the sequences of both antigens and antibodies.

Neighbors were determined by using the string distances between the CDR1, CDR2, and CDR3 amino acid sequences of different antibodies. Weights were calculated from distances, so that nearer neighbors were considered to have more weight, as detailed below.

For every antigen, the class (good affinity or low affinity) was learned by using the K-NN method, using a training subset ($N - 1$) of the labeled antigen-antibody sequence pairs and using the CDR string distances as features. The model performance was then evaluated on the remaining antigen-antibody sequence pair that was not used for training (leave-one-out cross-validation).

In order to ensure that the K-NN pairs only included pairs with the same antigen, a fixed penalty of 1000 was added to the distances between antibody-antigen pairs involving different antigens.

The similarity between antibodies was measured via a comparison of their CDRs. Each antibody has a heavy chain and a light chain, and each chain contains 3 CDRs. The distance between 2 antibodies was calculated as the Euclidean distance between their CDR distance vectors, as shown in the following equation (equation 1):

$$d(\mathbf{p}, \mathbf{q}) = d(\mathbf{q}, \mathbf{p}) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots + (q_n - p_n)^2} \\ = \sqrt{\sum_{i=1}^n (q_i - p_i)^2}. \quad (1)$$

where $(q_i - p_i)$ represents the string distance between the CDR_i of antibody q and the CDR_i of antibody p .

The Python code is given in [Multimedia Appendix 1](#).

Two different CDR distance calculation methods were tested and compared; one was based on sequence identity, and the other used the BLOSUM62 matrix, as detailed below.

For the identity-based distance measure, pairs of equivalent CDRs were compared with each other based on their Levenshtein string distances [22], as shown in the following equation (equation 2):

$$\text{lev}_{a,b}(i, j) = \begin{cases} \max(i, j) & \text{if } \min(i, j) = 0, \\ \min \begin{cases} \text{lev}_{a,b}(i-1, j) + 1 \\ \text{lev}_{a,b}(i, j-1) + 1 \\ \text{lev}_{a,b}(i-1, j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \quad (2)$$

$Cost=0$ for $a_i=b_i$, $Cost=1$ for $a_i \neq b_i$

The Levenshtein distance only accounts for amino acid identity when it is used for comparing sequences. A more biologically significant distance measure needs to take into account the different properties of amino acids, which means that some amino acid substitutions are more likely to be accepted in an

interaction than others. The BLOSUM62 substitution matrix [23] was used as a proxy for amino acid similarity in the Levenshtein distance calculation. Although the BLOSUM matrices were designed to reflect evolutionary conservation, they can provide an estimate of similarity in interaction potential [24].

The Levenshtein distance was calculated as per equation 2, using the following cost function:

For $a_i=b_i$, $Cost=0$

$$\text{For } a_i \neq b_i, \text{Cost} = D_{ij} = \frac{|S_{ij} - \frac{S_{ii} + S_{jj}}{2}|}{2} \quad (3)$$

where S_{ij} , S_{ii} , and S_{jj} are obtained from the BLOSUM62 matrix.

The following columns from the data set were used to train the model for leave-one-out cross-validation: *H chain CDR1 sequence*, *H chain CDR2 sequence*, *H chain CDR3 sequence*, *L chain CDR1 sequence*, *L chain CDR2 sequence*, *L chain CDR3 sequence*, *Antigen*, and *Docking result*. The trained model was then evaluated on its ability to predict the docking results from the other columns.

A random forest machine learning algorithm incorporating the previous K-NN results was also used for predicting antibody-antigen binding classification. The isoelectric point and net charge at neutral pH (7.0) for each CDR were used as additional features, in addition to the BLOSUM62-derived CDR distances, for training the random forest. Binding was predicted by combining the votes from each of the features, and each individual feature contributed 1 vote, according to the nearest neighbor predictions based on each feature.

The following columns from the data set were used for training the random forest: *String distance (calculate by KNN method)*, *Hydrophilicity of L CDR1*, *pI of L CDR1*, *Hydrophilicity of L CDR2*, *pI of L CDR2*, *Hydrophilicity of L CDR3*, *pI of L CDR3*, *Hydrophilicity of H CDR1*, *pI of H CDR1*, *Hydrophilicity of H*

CDR2, *pI of H CDR2*, *Hydrophilicity of H CDR3*, *pI of H CDR3*, *Antigen*, and *Docking result*. The trained model was then evaluated on its ability to predict the docking results from the other columns.

Each feature was considered as an individual decision tree and contributed 1 vote. For example, the isoelectric point of the CDR1 of an antibody's heavy chain was considered as 1 feature, and the K-NN method was used, as previously described, to find the results of this decision tree. Altogether, there were 13 decision trees, and each tree used the K-NN method to determine its vote, for a total of 13 votes. The final decision was determined based on a simple majority vote. The best results were obtained when the whole forest (all 13 decision trees) took part in the vote.

The performance of the K-NN and random forest learners was evaluated by using leave-one-out cross-validation on an antigen basis. For each of the 57 antigens, a training data set was constructed by removing 1 row, that is, 1 antibody-antigen pair, from the data set. After training with the remaining antibodies that bound to this antigen, model performance was evaluated based on the removed antibody. The process was repeated until all 5041 antibody-antigen pairs were tested. Model accuracy was calculated as the ratio of the number of correctly predicted antibody-antigen pairs over the total number of pairs in the data set.

Results

Data Set

A total of 600 antibody-antigen complexes were generated via the computational docking of 50 antibody structures with 50 antigen structures. In addition, a total of 4441 antibody-antigen pairs were extracted from the Cov-AbDab. The composition of this section of the data set is shown in Table 1.

In total, the data set contained 5041 antibody-antigen pairs comprising 1157 antibodies and 57 antigens.

Table 1. Number of antibodies and positive and negative antibody-antigen pairs extracted from the Coronavirus Antibody Database.

Antigen	Number of antibodies	Positive samples, n	Negative samples, n
SARS-CoV-2	1943	1912	31
SARS-CoV-1	1241	597	644
MERS-CoV ^a	264	119	145
HCoV-OC43 ^b	257	21	236
HCoV-HKU1 ^c	254	84	170
HCoV-NL63 ^d	258	51	207
HCoV-229E ^e	207	49	158

^aMERS-CoV: Middle East respiratory syndrome-related coronavirus.

^bHCoV-OC43: human coronavirus OC43.

^cHCoV-HKU1: human coronavirus HKU1.

^dHCoV-NL63: human coronavirus NL63.

^eHCoV-229E: human coronavirus 229E.

Machine Learning

The antigen-antibody binding classification methods were evaluated by using leave-one-out cross-validation. For a K value of 2 nearest neighbors, the K-NN method, when the Levenshtein distance was calculated based on sequence identity, achieved an accuracy of 81%. A slight improvement (accuracy of 82%) was observed when using the BLOSUM62 matrix to calculate the Levenshtein string distance.

Different K values were also evaluated when the Levenshtein distance was calculated based on the BLOSUM62 matrix. A K value of 2 provided the best accuracy. For a K value of 1 nearest neighbor, the accuracy was 80%. For a K value of 3, classification accuracy dropped to 79%.

For the random forest predictions, votes were used as the classification prediction results. The accuracy was highest when the whole forest was considered, in which case each feature contributed to the classification results. The performance of the random forest method was best (accuracy of 80%) when all 13 features—the Levenshtein string distance and the isoelectric point and net charge at neutral pH (7.0) for each CDR—took part in the final votes.

Discussion

We created a training and test data set of 5041 antibody-antigen complexes by using a combination of structure modeling and computational docking via Rosetta, together with antibody-antigen pairs extracted from the CoV-AbDab.

We also developed weighted nearest neighbor and random forest approaches to predict antibody-antigen binding based on sequence data. These machine learning procedures can perform classifications to identify antigens that are likely to bind to a given antibody.

Leave-one-out cross-validation testing yielded an accuracy of 82% for classification results that were based on 2 nearest neighbors. The prediction accuracy ranged from around 77% to 82% when varying the number of nearest neighbors. The best prediction results (accuracy of 82%) were obtained with 2 nearest neighbors, using string distance and BLOSUM62 matrices.

This study demonstrates that the interaction between an antibody and a protein antigen can be predicted from the amino acid sequences of both the antibody's variable regions and the antigen by using a relatively simple machine learning approach. Compared to the docking prediction method, which is based on the spatial protein structure, the method proposed in this project does not require a 3D structure and is more suitable for antibodies for which a 3D structure is unavailable.

In the absence of large amounts of experimental data on antibody-antigen binding affinities, the Rosetta interface scores, along with the top 10 binding positions, were used to determine the classification for binding affinity. Although this method was unlikely to provide a full representation of the problem, it

provided a data set suitable for comparing a range of approaches. This method will certainly improve as larger data sets become available. The docking data set contained 600 rows of antibody-antigen pairs. Subsets of this data set (200, 300, 400, and 500 rows) were tested during the data collection process. Classification accuracy was quite consistent across all of these subsets. This indicates that while the data set is limited, it provides a good starting point for the development of our approach for the prediction of antibody-antigen binding affinity, which can be further validated as more data become available. The K-NN method was chosen as the initial machine learning method. The best prediction results were obtained with 2 nearest neighbors (K=2). Random forests were also used that incorporated sequence distance as well as the chemical properties of CDRs (isoelectric point and hydrophobicity). The best prediction results (accuracy of 82%) were obtained with the nearest neighbor method when the Levenshtein distance was calculated based on BLOSUM62 matrices. The additional features included in the random forest did not improve classification accuracy, and this was probably due to these features' dependence on the amino acid sequences.

Around 20% (907/5041, 18%) of our method's predictions were inaccurate. These errors mostly occurred with some large antigens. The docking results for these antigens were further examined. The decreased accuracy was likely the result of conformational flexibility in the larger antigens, the presence of multiple epitopes, and the higher number of discontinuous epitopes in larger antigens relative to the number of such epitopes in smaller antigens.

As a step toward the development of a machine learning method suitable for predicting antibody-antigen binding affinities from sequence data, the weighted nearest neighbor and random forest machine learning approaches were applied to the problem. The basic hypothesis was that antibodies with similar sequences may be similar in terms of their ability to bind to a given antigen. A prediction program was coded in Python and evaluated via cross-validation on a data set containing 1157 antibodies and 57 antigens that were combined in 5041 antibody-antigen pairs. The best classification prediction accuracy was around 82% for this data set.

These results provide a useful frame of reference, as well as protocols and considerations, for machine learning and data set creation in the prediction of antibody-antigen binding. Our method is still limited due to the scarcity of training data, but its usefulness for large-scale prediction should increase as more antibody-antigen binding data become available. The ability to predict antibody-antigen binding will allow for a more informed use of data from large-scale immune receptor sequencing. This, in turn, will increase our understanding of the variation in antigen recognition in an organism over time, under a range of conditions and between individuals and populations.

Both the data set (in CSV format) and the machine learning program (coded in Python) are freely available for download on GitHub [20].

Conflicts of Interest

None declared.

Multimedia Appendix 1

Python code for Euclidean distance calculation.

[\[DOCX File , 12 KB-Multimedia Appendix 1\]](#)

References

1. Dunn-Walters D, Townsend C, Sinclair E, Stewart A. Immunoglobulin gene analysis as a tool for investigating human immune responses. *Immunol Rev* 2018 Jul;284(1):132-147 [FREE Full text] [doi: [10.1111/imr.12659](https://doi.org/10.1111/imr.12659)] [Medline: [29944755](https://pubmed.ncbi.nlm.nih.gov/29944755/)]
2. Boyd SD, Crowe JEJ. Deep sequencing and human antibody repertoire analysis. *Curr Opin Immunol* 2016 Jun;40:103-109 [FREE Full text] [doi: [10.1016/j.coi.2016.03.008](https://doi.org/10.1016/j.coi.2016.03.008)] [Medline: [27065089](https://pubmed.ncbi.nlm.nih.gov/27065089/)]
3. Weitzner BD, Jeliakov JR, Lyskov S, Marze N, Kuroda D, Frick R, et al. Modeling and docking of antibody structures with Rosetta. *Nat Protoc* 2017 Feb;12(2):401-416 [FREE Full text] [doi: [10.1038/nprot.2016.180](https://doi.org/10.1038/nprot.2016.180)] [Medline: [28125104](https://pubmed.ncbi.nlm.nih.gov/28125104/)]
4. Pires DEV, Ascher DB. mCSM-AB: a web server for predicting antibody-antigen affinity changes upon mutation with graph-based signatures. *Nucleic Acids Res* 2016 Jul 08;44(W1):W469-W473 [FREE Full text] [doi: [10.1093/nar/gkw458](https://doi.org/10.1093/nar/gkw458)] [Medline: [27216816](https://pubmed.ncbi.nlm.nih.gov/27216816/)]
5. Vivcharuk V, Baardsnes J, Deprez C, Sulea T, Jaramillo M, Corbeil CR, et al. Assisted Design of Antibody and Protein Therapeutics (ADAPT). *PLoS One* 2017 Jul 27;12(7):e0181490 [FREE Full text] [doi: [10.1371/journal.pone.0181490](https://doi.org/10.1371/journal.pone.0181490)] [Medline: [28750054](https://pubmed.ncbi.nlm.nih.gov/28750054/)]
6. Kozakov D, Hall DR, Xia B, Porter KA, Padhorny D, Yueh C, et al. The ClusPro web server for protein-protein docking. *Nat Protoc* 2017 Feb;12(2):255-278 [FREE Full text] [doi: [10.1038/nprot.2016.169](https://doi.org/10.1038/nprot.2016.169)] [Medline: [28079879](https://pubmed.ncbi.nlm.nih.gov/28079879/)]
7. Comeau SR, Gatchell DW, Vajda S, Camacho CJ. ClusPro: an automated docking and discrimination method for the prediction of protein complexes. *Bioinformatics* 2004 Jan 01;20(1):45-50. [doi: [10.1093/bioinformatics/btg371](https://doi.org/10.1093/bioinformatics/btg371)] [Medline: [14693807](https://pubmed.ncbi.nlm.nih.gov/14693807/)]
8. Kozakov D, Brenke R, Comeau SR, Vajda S. PIPER: an FFT-based protein docking program with pairwise potentials. *Proteins* 2006 Nov 01;65(2):392-406. [doi: [10.1002/prot.21117](https://doi.org/10.1002/prot.21117)] [Medline: [16933295](https://pubmed.ncbi.nlm.nih.gov/16933295/)]
9. Comeau SR, Gatchell DW, Vajda S, Camacho CJ. ClusPro: a fully automated algorithm for protein-protein docking. *Nucleic Acids Res* 2004 Jul 01;32(Web Server issue):W96-W99 [FREE Full text] [doi: [10.1093/nar/gkh354](https://doi.org/10.1093/nar/gkh354)] [Medline: [15215358](https://pubmed.ncbi.nlm.nih.gov/15215358/)]
10. Sircar A, Gray JJ. SnugDock: paratope structural optimization during antibody-antigen docking compensates for errors in antibody homology models. *PLoS Comput Biol* 2010 Jan 22;6(1):e1000644 [FREE Full text] [doi: [10.1371/journal.pcbi.1000644](https://doi.org/10.1371/journal.pcbi.1000644)] [Medline: [20098500](https://pubmed.ncbi.nlm.nih.gov/20098500/)]
11. Lyskov S, Gray JJ. The RosettaDock server for local protein-protein docking. *Nucleic Acids Res* 2008 Jul 01;36(Web Server issue):W233-W238 [FREE Full text] [doi: [10.1093/nar/gkn216](https://doi.org/10.1093/nar/gkn216)] [Medline: [18442991](https://pubmed.ncbi.nlm.nih.gov/18442991/)]
12. Lyskov S, Chou FC, Conchúir S, Der BS, Drew K, Kuroda D, et al. Serverification of molecular modeling applications: the Rosetta Online Server that Includes Everyone (ROSIE). *PLoS One* 2013 May 22;8(5):e63906 [FREE Full text] [doi: [10.1371/journal.pone.0063906](https://doi.org/10.1371/journal.pone.0063906)] [Medline: [23717507](https://pubmed.ncbi.nlm.nih.gov/23717507/)]
13. Guex N, Peitsch MC. SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling. *Electrophoresis* 1997 Dec;18(15):2714-2723. [doi: [10.1002/elps.1150181505](https://doi.org/10.1002/elps.1150181505)] [Medline: [9504803](https://pubmed.ncbi.nlm.nih.gov/9504803/)]
14. RCSB PDB: Homepage. RCSB Protein Data Bank. URL: <https://www.rcsb.org/> [accessed 2019-07-12]
15. Kilambi KP, Gray JJ. Structure-based cross-docking analysis of antibody-antigen interactions. *Sci Rep* 2017 Aug 15;7(1):8145 [FREE Full text] [doi: [10.1038/s41598-017-08414-y](https://doi.org/10.1038/s41598-017-08414-y)] [Medline: [28811664](https://pubmed.ncbi.nlm.nih.gov/28811664/)]
16. Raybould MIJ, Kovaltsuk A, Marks C, Deane CM. CoV-AbDab: The Coronavirus Antibody Database. *Bioinformatics* 2021 May 05;37(5):734-735 [FREE Full text] [doi: [10.1093/bioinformatics/btaa739](https://doi.org/10.1093/bioinformatics/btaa739)] [Medline: [32805021](https://pubmed.ncbi.nlm.nih.gov/32805021/)]
17. IMGT home page. The international ImMunoGeneTics information system. URL: <http://www.imgt.org> [accessed 2020-01-23]
18. Peptide calculator. Bachem. URL: <https://www.bachem.com/knowledge-center/peptide-calculator/> [accessed 2020-03-10]
19. Antibody epitope prediction. Immune Epitope Database. URL: <http://tools.iedb.org/bcell/> [accessed 2020-07-12]
20. Chao Ye. jessye123/ab-ag-seq-machine-learning. GitHub. URL: <https://github.com/jessye123/ab-ag-seq-machine-learning> [accessed 2022-10-19]
21. Taunk K, De S, Verma S, Swetapadma A. A brief review of nearest neighbor algorithm for learning and classification. 2019 Presented at: 2019 International Conference on Intelligent Computing and Control Systems (ICCS); May 15-17, 2019; Madurai, India p. 1255-1260. [doi: [10.1109/iccs45141.2019.9065747](https://doi.org/10.1109/iccs45141.2019.9065747)]
22. Levenshtein VI. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady* 1966 Feb;10(8):707-710 [FREE Full text]
23. Henikoff S, Henikoff JG. Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci U S A* 1992 Nov 15;89(22):10915-10919 [FREE Full text] [doi: [10.1073/pnas.89.22.10915](https://doi.org/10.1073/pnas.89.22.10915)] [Medline: [1438297](https://pubmed.ncbi.nlm.nih.gov/1438297/)]

24. Huang YA, You ZH, Gao X, Wong L, Wang L. Using Weighted Sparse Representation Model Combined with Discrete Cosine Transformation to Predict Protein-Protein Interactions from Protein Sequence. *Biomed Res Int* 2015;2015:902198 [FREE Full text] [doi: [10.1155/2015/902198](https://doi.org/10.1155/2015/902198)] [Medline: [26634213](https://pubmed.ncbi.nlm.nih.gov/26634213/)]

Abbreviations

ADAPT: Assisted Design of Antibody and Protein Therapeutics

CDR: complementarity-determining region

CDR1: first complementarity-determining region

CDR2: second complementarity-determining region

CDR3: third complementarity-determining region

CoV-AbDab: Coronavirus Antibody Database

IEDB: Immune Epitope Database

K-NN: K-nearest neighbor

MERS-CoV: Middle East respiratory syndrome-related coronavirus

PDB: Protein Data Bank

Edited by A Mavragani; submitted 06.04.21; peer-reviewed by Z Qiu, Y Xiao, ME Ackerman, H Sundaramoorthi; comments to author 20.05.21; revised version received 23.09.21; accepted 18.10.22; published 28.10.22

Please cite as:

Ye C, Hu W, Gaeta B

Prediction of Antibody-Antigen Binding via Machine Learning: Development of Data Sets and Evaluation of Methods

JMIR Bioinform Biotech 2022;3(1):e29404

URL: <https://bioinform.jmir.org/2022/1/e29404>

doi: [10.2196/29404](https://doi.org/10.2196/29404)

PMID:

©Chao Ye, Wenxing Hu, Bruno Gaeta. Originally published in JMIR Bioinformatics and Biotechnology (<https://bioinform.jmir.org>), 28.10.2022. This is an open-access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work, first published in JMIR Bioinformatics and Biotechnology, is properly cited. The complete bibliographic information, a link to the original publication on <https://bioinform.jmir.org/>, as well as this copyright and license information must be included.