BepiPred-3.0: Improved B-cell epitope prediction using protein language models

Clifford, Joakim Nøddeskov; Høie, Magnus Haraldson; Deleuran, Sebastian; Peters, Bjoern; Nielsen, Morten; Marcatili, Paolo

Published in:
Protein Science

Link to article, DOI:
10.1002/pro.4497

Publication date:
2022

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Clifford, J. N., Høie, M. H., Deleuran, S., Peters, B., Nielsen, M., & Marcatili, P. (2022). BepiPred-3.0: Improved B-cell epitope prediction using protein language models. Protein Science, 31(12), [e4497]. https://doi.org/10.1002/pro.4497
BepiPred-3.0: Improved B-cell epitope prediction using protein language models

Joakim Nøddeskov Clifford1 | Magnus Haraldson Høie1 | Sebastian Deleuran1 | Bjoern Peters2 | Morten Nielsen1 | Paolo Marcatili1

1Department of Health Technology, Technical University of Denmark, Kongens Lyngby, Denmark
2La Jolla Institute for Immunology, La Jolla, California, USA

Abstract

B-cell epitope prediction tools are of great medical and commercial interest due to their practical applications in vaccine development and disease diagnostics. The introduction of protein language models (LMs), trained on unprecedented large datasets of protein sequences and structures, tap into a powerful numeric representation that can be exploited to accurately predict local and global protein structural features from amino acid sequences only. In this paper, we present BepiPred-3.0, a sequence-based epitope prediction tool that, by exploiting LM embeddings, greatly improves the prediction accuracy for both linear and conformational epitope prediction on several independent test sets. Furthermore, by carefully selecting additional input variables and epitope residue annotation strategy, performance was further improved, thus achieving unprecedented predictive power. Our tool can predict epitopes across hundreds of sequences in minutes. It is freely available as a web server and a standalone package at https://services.healthtech.dtu.dk/service.php?BepiPred-3.0 with a user-friendly interface to navigate the results.

KEYWORDS

BepiPred-3.0, BepiPred, B-cell epitope prediction, protein language model, machine learning, deep learning, immunology, B-cell epitopes, bioinformatics, immunoinformatics

1 | INTRODUCTION

B-cells are a major component of the adaptive immune system, as they support long-term immunological protection against pathogens and cancerous cells. Their activation relies on the interaction between specialized receptors known as B-cell receptors (BCRs) and their pathogenic targets, also known as antigens. More specifically, BCRs selectively interact with specific portions of their antigens known as epitopes. B-cell epitopes are divided into two types. Linear epitopes are found sequentially along the amino-acid sequence, while conformational epitopes are interspersed in the antigen's primary structure, and brought together in spatial proximity by the antigen's folding. While approximately 90% of B-cell epitopes fall into the conformational category, most of these contain at least a few sequential residue stretches. Epitopes are typically found in solvent-exposed regions of antigens, but physical and chemical features such as hydrophobicity, secondary structure propensity, protrusion indexes, and local amino acid composition, have been shown to affect the likelihood of epitopes.
B-cell epitope identification is of great interest in biotechnological and clinical applications, such as subunit vaccine designs, disease diagnostics, and therapeutic antibody development. Their identification, however, is a costly and time-consuming process requiring extensive experimental assay screening. In silico prediction methods can significantly reduce identification workloads by predicting epitope regions, and because of this they have become critical for such tasks. Structure-based tools have been developed for predicting B-cell epitopes. However, as experimentally determined structural information is often not available, epitope identification must in many cases be performed from amino-acid sequences alone. So far, sequence-based tools have only achieved mediocre results, in general worse than structure-based tools. Recently, accurate protein structure prediction models such as AlphaFold and AlphaFold-Multimer have been proposed, greatly advancing the accuracy for protein structure prediction. However, the application of such models for accurate prediction of structures in emerging pathogens remains challenging due to high computational cost, and limited accuracy for novel protein families.

Thanks to recent developments in the field of machine learning, models trained on large datasets of protein sequences and structures are now available to accurately predict local and global protein structural features from amino acid sequences only. In particular, protein language models (LMs) have been demonstrated to allow for a powerful numeric representation of protein sequences, that in turn can be exploited to substantially increase the accuracy in many different prediction tasks. An interesting advantage that these representations have over traditional amino acid encoding, such as sparse or blosum, is that they capture signals between residues that are far apart in primary sequence, which potentially could be helpful for discovering conformational B-cell epitopes.

Here, we present BepiPred-3.0, a sequence-based tool, which utilizes numerical representations from the protein language model ESM-2, to vastly improve prediction accuracy for linear and conformational B-cell epitope prediction. Furthermore, by carefully selecting the architecture of the predictor, the training strategy, additional input variables to the model, and using an epitope residue annotation strategy adopted from one of our earlier works, performance was further improved, achieving unprecedented predictive performance.

2 | METHODS

2.1 | Structural datasets

A first dataset, named BP2, consists of the antigens used for training the BepiPred 2.0 server. This dataset contains 776 antigens and is available at the BepiPred-2.0 web server. A second updated dataset, named BP3, was built using the same approach previously adopted in BepiPred 2.0. We first identified crystal structures from the Protein Data Bank deposited before September 29, 2021 that contain at least one complete antibody, and at least one non-antibody (antigen) protein chain. This was done using existing hidden markov model (HMM) profiles developed in-house. We only included crystal structures with a resolution lower than 3 Å and R-factor lower than 0.3. On both datasets, we identified epitope residues using the same approach adopted in our previous paper. On each antigen chain, we labeled every residue that had at least one heavy atom (main-chain or side-chain) at a distance of <4 Å to any heavy atom belonging to residues in antibody chains of the same crystal structure as an epitope residue. We only retained antigen chains with at least one epitope residue and with a minimum sequence length of 39. Epitope residues for antigens which were 100% identical in sequence were merged and only one antigen entry included. Missing residues were not included as part of the epitope annotated antigen chains. After these steps, we obtained a total of 1,466 antigens for the updated BP3 dataset.

2.2 | Redundancy reduction

We used a redundancy reduction approach similar to existing works, which we called the epitope collapse strategy. Here, sequence clusters were first generated using MMseqs2. Next, all antigen sequences belonging to the same cluster were aligned to the MMseqs2 defined cluster representative. The cluster representative sequence then underwent the following modifications: At any position of the alignment, if an epitope was identified in the cluster representative, it was retained as is. If an epitope was found on any of the aligned antigen sequences, it was grafted onto the cluster representative sequence and labelled as epitope. For each cluster, only the sequence of the cluster representative, modified as described above, was retained. The epitope collapse was done at 95% sequence identity for both the BP2 and BP3 datasets, making the size of the BP2 and BP3 datasets 238 and 603 antigens, reduced from 776 and 1,466 antigens, respectively. Furthermore, the strategy was performed at 50% sequence identity for the BP3 dataset. This dataset was called BP3C50ID and contained 358 antigens reduced from 1,466 antigens.

Redundancy was further reduced for two additional datasets. The sequences of the final BP2 and BP3 datasets described above were clustered at 70% sequence identity using MMseqs2, and then only the cluster representatives were incorporated into the reduced datasets. This gave
rise to datasets BP2HR and BP3HR, which contained 190 and 398 antigens, reduced from 238 and 603 antigens, respectively (Table 1).

2.3 | External test sets

Three different independent test sets were built. The first corresponds to the five antigens used as external test evaluation for BepiPred-2.0. Here, however, the antigens were extracted from BP3C50ID, and therefore contained updated and enriched epitope annotations. Any sequence with more than 20% sequence identity to this test set, as calculated by MMseqs2, was removed from the BP2 and BP2HR training sets. After this removal, BP2 and BP2HR training sets contained 233 and 185 antigens. The second external test set comprises the antigens mentioned above, plus 10 additional BP3C50ID antigens selected from the MMseqs2 clusters at 20% identity. Here, any sequence with more than 20% identity to any of the 15 antigens was removed from all BP3 training sets. After this removal, the training sets for BP3, BP3HR and BP3C50ID contained 582, 383, and 343 antigens.

Finally, a third external dataset was constructed by downloading all linear B-cell epitopes from the Immune Epitope Database (IEDB) and then discarding any epitopes containing post translational modifications, as well as epitopes for which the source protein ID, as indicated in the IEDB entry itself, was not a UniProt entry. Epitopes with a perfect match in the source protein were mapped to the relevant region while the rest were discarded. This resulted in 4072 protein sequences with mapped epitopes. Finally, we removed all proteins which had more than 20% sequence identity to the BP3C50ID training set, leaving 3,560 sequences (Table 2).

2.4 | Dataset encoding

Residues were represented either by sparse encoding, BLOSUM62 log-odds scores or by numeric embeddings extracted from the ESM-2 protein language model. For sparse encoding, a single residue can be represented as a vector of size 21, corresponding to a sparse encoding of the 20 amino acids and a padding token. Likewise, a single residue BLOSUM62 encoding is also a vector of size 21, consisting of the residue’s log-odds substitution score paired with all other amino acids (including itself) and a padding token. When employing sparse and BLOSUM62 encodings, the encoding of each residue was generated by concatenating sparse or BLOSUM62 encodings from the residue itself and from its eight neighboring residues, four on each side. For residues close to sequence terminals where an insufficient number of neighbors existed, padding tokens were used to fill in for lacking residues. Thus, each residue was represented by a vector of size 189 (9*21).

ESM-2 encodings were obtained by passing antigen sequences through the pretrained ESM-2 transformer, and extracting the resulting sequence representations from the model. For ESM-2 encoding, no neighboring residue’s ESM-2 encoding was used, as they were not part of the training set. Instead, a set of random ESM-2 encodings were used as a source of diversity to help the model learn more general features.

---

### Table 1

| Antigens | Epi. res. | Epi. ratio | Epi. col. | Redu. red. |
|----------|-----------|------------|-----------|------------|
| BP2 bef. epi collapse | 776 | 6,591 | 0.10 | – |
| BP2 | 238 | 2,106 | 0.103 | 95% | – |
| BP2HR | 190 | 1,679 | 0.103 | 95% | 70% |
| BP3 bef. epi collapse | 1,466 | 12,981 | 0.09 | – | – |
| BP3 | 600 | 6,597 | 0.112 | 95% | – |
| BP3HR | 398 | 4,481 | 0.117 | 95% | 70% |
| BP3C50ID | 358 | 5,011 | **0.134** | 50% | – |

**Note:** Epitope residue ratios were computed as the ratios between the number of epitope residues and total number of residues. The redundancy reduction and epitope collapse columns are the MMseqs2 sequence identity %s, for which the epitope collapse and the second redundancy reduction approaches were used.

### Table 2

| Antigens | Epi. res. | Epi. ratio | Epi. col. |
|----------|-----------|------------|-----------|
| External test set 1 | 5 | 88 | 0.256 | 50% |
| External test set 2 | 15 | 209 | 0.190 | 50% |
| External IEDB test set | 3,560 | 8,818 | 0.116 | – |

**Note:** Epitope residue ratios were computed as the ratios between the number of epitope residues and total number of residues. The epitope collapse strategy column is the MMseqs2 sequence identity %, for which epitope collapse was done.
residues were included and each residue was thus represented by a vector of size 1,280. We also included NetSurfP-3.0 predicted RSA values and protein sequence lengths to ESM-2 encodings.\(^{10}\) The target values were encoded in a position-wise binary manner, resulting in a sequence denoting epitope and non-epitope residues with the same length as the antigen sequence (Figure 1).

### 2.5 Model architectures and hyperparameter tuning

Feed Forward (FFNN), Convolutional (CNN), and Long Short-term Memory (LSTM) neural networks were trained on sparse, BLOSUM62 and ESM-2 encodings, with or without the additional variables (sequence length or NetSurfP-3.0 predicted RSA values). Model weights were initialized and updated using the default PyTorch weight initialization schemes and an Adam optimizer.\(^{13}\) Since we have independent test data, hyperparameter tuning could be performed in a simple five-fold cross validation setup using a grid search on the training data. Hyperparameters were chosen as those which yielded the best validation cross-entropy (CE) loss averaged across all folds. The hyperparameters in question are the learning rate, weight decay, dropout rate and different architectural setups (see supplementary methods Section S5.3 for the exact final model configurations). As a baseline, a Random Forest Classifier (RFC) was trained on sparse and BLOSUM62 encodings. We tested forest sizes ranging from 25 to 300 and determined the optimal size on a validation area under curve (AUC) score basis (see supplementary method Section S5.2).

### 2.6 Training and evaluation

A five-fold cross-validation was used to train the models with the optimized hyperparameters discussed in the previous section. For BP2 and BP3 cross-validation setups, antigens were clustered at 70% sequence identity using MMseqs2, and sequences of the same cluster placed into the same partition. A total of 5 partitions were created.\(^{27}\) For the three remaining datasets, training, and validation splits were made randomly. Each cross-validation setup generated five models, where both validation cross-entropy (CE) loss and AUC were used as an early stopping criteria. For instance, the training procedure for a fold was the following: Model weights were initialized using the default PyTorch weight initialization schemes and the Adam optimizer.\(^{13}\) After each epoch, a cross entropy loss and AUC score was computed on the validation set. Only if both scores improved, were the model parameters stored. We trained for 75 epochs.

Next, the models were evaluated as an ensemble on the external test sets. The evaluative metrics on the external test sets were AUC, AUC10, MCC, recall, precision, F1-score, and accuracy. AUC scores were computed on the averaged probability outputs of the five models. AUC10 scores were computed as the integral of the ROC curve area going from 0 to 0.1 on the false positive rate axis, divided by 0.1, setting the AUC10 score in a range of 0–1. For the remaining threshold-dependent metrics, a majority voting scheme was based on the individual predictions from the five models. The classification threshold used for each fold model was one that maximized the MCC score on respective validation splits. When evaluating our models on the IEDB external test set, a moving average with a window size of 9 was performed on model probability outputs.

For some model comparisons, one-sided \(p\)-values comparing AUC test performances on external test set 1 and 2 were performed. This was done using a bootstrap method and sampling \(N = 10,000\) times with replacement on the external test set residues. There were 971 and 2,670 residues in external test 1 and 2, respectively. The

---

**FIGURE 1** Overview of sequence encoding pipelines, where \(N\) is length of the sequence. Amino acid sequences were encoded using either sparse, BLOSUM62 or ESM-2 derived sequence representation schemes. For the two former approaches, encodings from adjacent residues were concatenated to generate a new set of encodings describing the sequence context of each residue. The encoded sequences were subsequently used for training various models for position-wise antigen prediction.
procedure for comparing two models \((m1 \text{ and } m2)\) was the following: For each bootstrap sample of test set residues, the corresponding \(m1\) and \(m2\) outputs were collected and 2 AUC scores computed. Hereafter, instances where \(m2\) outperformed \(m1\) were counted. We then used Equation (1), to calculate a bootstrap \(p\)-value for \(m1\)'s performance being statistically equal or higher than \(m2\).

\[
p = \frac{\sum \text{m2} \geq \text{m1}}{N}
\]  

If there were no instances of \(m2\) outperforming \(m1\), the bootstrap \(p\)-value was evaluated as being <1/10,000.

## 3 | RESULTS

### 3.1 | Improved performance on BP3 datasets and when the epitope collapse strategy is used

In an initial benchmark study, we investigated the effects of using updated datasets (BP3, BP3HR, and BP3C50ID) versus datasets constructed from the BepiPred2 paper (BP2, BP3HR), as well as various redundancy reduction approaches (see Section 2.2 for more details). We determined that the BP3 datasets (BP3 and BP3HR) led to an improved predictive performance compared to models trained using the smaller BP2 datasets (BP2 and BP2HR). This was observed by training a set of random forest classifiers (RFCs) and evaluating on the five antigen external test set.

We improved performance further by first doing sequence redundancy reduction at a 50% identity threshold as defined by MMsseqs2, and then using the epitope collapse strategy (BP3C50ID dataset) where all epitopes for sequences found in a given cluster are transferred to the cluster representative antigen. We determined that the performance increases from using updated datasets and the epitope collapse strategy, were statistically significant at all common thresholds \((p\)-value <.001\)) (for more details on the epitope collapse strategy refer Section 2.2, and for details on these results see supplementary result Section S6.1). Given these results, we therefore only used the BP3 and BP3C50ID datasets for the subsequent analyses.

### 3.2 | Improved performance using neural networks and ESM-2 sequence embeddings

The main goal of this paper was to demonstrate that a B-cell epitope predictive tool based on LM embeddings will perform better than models using other encoding schemes, such as BLOSUM62 or sparse encoding. To assess the best performing machine-learning architecture and sequence representation for the BP3 and BPC50ID datasets, optimal hyperparameters for four different architectures (RFCs, FFNNs, CNNs, and LSTMs) were identified using a grid search (see Section 2.5). For each architecture, we investigated the performance when representing protein residues with sparse encoding, BLOSUM62 encoding or ESM-2 embedding. The best performing models were selected from optimal cross-validation performance, and tested on the 15 antigen external test set using a classification threshold optimized on the validation splits (Table 3).

We found that all neural networks using ESM-2 sequence embeddings as input performed better. We determined that the performance increase of models using ESM-2 embeddings instead of BLOSUM62 encodings was statistically significant at all common thresholds \((p\)-value <.001\)) for the FFNN, CNN, and LSTM. These \(p\)-values were computed using a bootstrapping method comparing AUC performance on external test set 2. Contrary to our expectations, the FFNN architecture using ESM-2 sequence embeddings as input, performs better than the CNN (ESM-2) and LSTM (ESM-2) models in terms of the vast majority of performance metrics. While CNNs and LSTMs sequentially process residues along the antigen sequence, the FFNN was trained on single residue ESM-2 encodings without applying any sliding window to the input data. This suggests that the ESM-2 residue embeddings already contain sufficient information about the sequence neighborhood, making convolutions over the sequence unnecessary.

We also did an identical analysis for models trained on the BP3 dataset, which gave us slightly worse results than those presented in Table 3 (see supplementary results Section S6.2). And so similar to the initial benchmark using RFCs, we find that the models trained on the BP3C50ID dataset perform better. This points to the epitope collapse strategy improving performance. Importantly, we investigated how the epitope collapse strategy affected predictions on a test set without collapsed epitopes. We note that the collapsing of epitopes also includes the possibility of adding additional epitope residues to the chosen sequence, which in turn would modify the extracted ESM-2 sequence embeddings. Here, we found that there was no apparent decrease in performance (see supplementary results Section S6.5).

To conclude, we find that while part of the improvement can be ascribed to training on a larger dataset and the epitope collapse strategy (see Section 3.1), a massive improvement stems from using LM embeddings (Table 3 and Figure 2).
We uncovered a positive evaluation performance on the external test set of 15 antigens, with a set of RFC, FFNN, CNN, and LSTM models, trained on cross-validation on the BP3C50ID dataset. We evaluated the performance on all test antigen residues, and determined that this performance increase was statistically significant. We also uncovered a negative relationship between RSA and the length of the antigen sequences and their respective epitope residue ratios, with a Spearman correlation coefficient of −0.58 (see supplementary results Section S6.4). We expect this to be due to multiple effects, concerning the larger surface-to-volume ratio of small proteins, the issue of a limited number of antibodies being mapped to individual antigens, as well as possible selection biases in the dataset. Interestingly, we also found that the performance of our best models decreased for longer antigen sequences, suggesting that the models could not infer the protein sequence length from the sequence embeddings. To account for this trend, we trained another FFNN on the BP3C50ID dataset, where sequence lengths were added as an additional input, and the resulting models were evaluated on the same 15 antigen test set. This improved the AUC performance from 0.762 to 0.771 (Table 4), which was found to be statistically significant when computing a bootstrap p-value of .010. This model was used as the final model for the BepiPred-3.0 web server and for further benchmarking. We also compared performance to a model where both NetSurfP-3.0 RSA and sequence lengths were added as additional inputs. When we computed a bootstrap p-value, we found that this model was statistically superior to FFNN (+SeqLen). However, due to the high computational cost of predicting this feature, and its minute contribution to AUC performance (from 0.771 to 0.774), we decided not to add this feature for our final model.

### 3.3 Feature engineering: adding sequence lengths improves performance

In BepiPred-2.0, one of the main factors that contributed to predictive performance was relative solvent accessibility (RSA), an input feature predicted from the antigen sequence using NetSurfP-2.0. We uncovered a positive correlation between NetSurfP-2.0 predicted RSA values and BepiPred-3.0 B-cell epitope probability scores, indicating that information on solvent accessibility is, at least in part, encoded in the ESM-2 embeddings (see supplementary results Section S6.3). To quantify to what extent RSA contributes to epitope predictions in BepiPred-3.0, we compared the performance of an FFNN trained on the BP3C50ID dataset with and without NetSurfP-3.0 RSA added as an input feature. We evaluated the performance on the same external test set of 15 antigens. Here, we found that the added RSA feature improved AUC performance from 0.762 to 0.770 (Table 4). We computed a bootstrapped p-value comparing AUC performances on all test antigen residues, and determined that this performance increase was statistically significant (p-value <.001). We note that NetSurfP-3.0 itself uses ESM embeddings to predict RSA, but these are generated from the earlier version, ESM-1b.

We also uncovered a negative relationship between the length of the antigen sequences and their respective epitope residue ratios, with a Spearman correlation coefficient of −0.58 (see supplementary results Section S6.4). We expect this to be due to multiple effects, concerning the larger surface-to-volume ratio of small proteins, the issue of a limited number of antibodies being mapped to individual antigens, as well as possible selection biases in the dataset. Interestingly, we also found that the performance of our best models decreased for longer antigen sequences, suggesting that the models could not infer the protein sequence length from the sequence embeddings. To account for this trend, we trained another FFNN on the BP3C50ID dataset, where sequence lengths were added as an additional input, and the resulting models were evaluated on the same 15 antigen test set. This improved the AUC performance from 0.762 to 0.771 (Table 4), which was found to be statistically significant when computing a bootstrap p-value of .010. This model was used as the final model for the BepiPred-3.0 web server and for further benchmarking. We also compared performance to a model where both NetSurfP-3.0 RSA and sequence lengths were added as additional inputs. When we computed a bootstrap p-value, we found that this model was statistically superior to FFNN (+SeqLen). However, due to the high computational cost of predicting this feature, and its minute contribution to AUC performance (from 0.771 to 0.774), we decided not to add this feature for our final model.

### 3.4 BepiPred-3.0 web server

BepiPred-3.0 is an easy to use tool for B-cell epitope prediction, as the user only needs to upload protein sequence(s) in fasta format. Furthermore, one can specify the threshold for epitope classification (by default, a...
threshold of 0.1512 is used). Alternatively, the user may specify the number of top residue candidates that should be included for each protein sequence. These options generate two separate fasta formatted output files.

A total of 4 result files are generated. One is a fasta file containing epitope predictions at the set threshold, and another is a similar fasta file, but epitope predictions are instead the number of top residue candidates specified by the user. In each file, epitope predictions are indicated with letter capitalization. The third is a .csv formatted file containing all model probability outputs on the uploaded protein sequence(s). Finally, we provide a .html file, which works as a graphical user interface.

FIGURE 2 ROC-AUC curves for the BP3C50ID FFNN, illustrate the difference of using sparse (a), BLOSUM62 (b), or ESM-2 encodings (c). The x and y axis are the false and true positive rates, respectively. Dashed lines along the diagonal indicate random performance at 50% AUC, and the remaining lines are the performances of different fold models. A confusion matrix illustrates the threshold-dependent performance of the FFNN (ESM-2) ensemble (d). The true negatives or positives and predicted negatives or positives are on the vertical and horizontal axis, respectively.

TABLE 4 A FFNN using ESM-2 encodings as well as additional inputs (NetSurfP-3.0 computed RSA, sequence lengths, or both) was evaluated on the external B test set of 15 antigens

| Models                | AUC  | AUC10 | MCC  |
|-----------------------|------|-------|------|
| FFNN                  | 0.762| 0.209 | 0.309|
| FFNN (+ NSP-3.0 RSA)  | 0.770| 0.205 | 0.310|
| FFNN (+ SeqLen)       | 0.771| 0.196 | 0.332|
| FFNN (NSP-3.0 RSA + SeqLen) | **0.774**| 0.199 | 0.327|

Note: The models were trained using cross-validation on the BP3C50ID dataset and chosen in terms of best validation CE loss and AUC. The evaluative metrics used were AUC, AUC10 a MCC. The best score within a metric category is marked in bold.
that can be opened in any browser. Similar to BepiPred-2.0, this interface can be used for setting a threshold for each antigen and downloading the corresponding B-cell epitope predictions. Due to memory limitations, however, this interface is limited to the first 40 sequences in the uploaded fasta file. We believe this intuitive interface will allow researchers to maximize their precision of B-cell epitope prediction, as a single threshold might not work for all uploaded sequences (Figure 3).

3.5 Benchmarking: BepiPred-3.0 outperforms its predecessors as well as structure-dependent B-cell epitope prediction tools

BepiPred-3.0 was re-evaluated and compared to its two predecessors on the five antigens from the BepiPred-2.0 paper external test set for a direct benchmarking. Here, we find a drastic improvement in BepiPred-3.0 versus its predecessors with AUC values of 0.57, 0.60, and 0.74 for BepiPred-1.0, 2.0, and 3.0, respectively. Likewise on the 15 antigen and IEDB external test sets, BepiPred-3.0 obtains substantial AUC performance gains of 10 and 14% compared to BepiPred-2.0. We note that the evaluated performance of BepiPred-2.0 on the 15 antigen external test set may be overfit, as these antigens overlap with the BepiPred-2.0 training data. Overall, these comparisons demonstrate an improved ability to generalize on novel datasets when using the ESM-2 protein LM embeddings (Table 5).

We also benchmarked BepiPred-3.0 against a recently developed structure-based B-cell epitope predictive tool, epitope3D, which in their original publication was shown to outperform different other tools.

To allow for a fair comparison, we used a five-fold cross-validation setup on the 200 antigens available at the epitope3D online tool, to retrain BepiPred-3.0. We then evaluated the re-trained model on the external test set composed of 45 antigens provided in the epitope3D publication. The evaluations in the epitope3D paper were done only on surface residues, and so to ensure a fair comparison, we calculated the performance of BepiPred-3.0 and other tools (BepiPred-2.0, Seppa-3.0, Discotope-2.0, Elli-pro) on a subset of surface residues with an RSA above 15%, as defined in the epitope3D paper (Table 6). While epitope3D obtained an AUC of 0.59 on their test set, BepiPred-3.0 obtained an AUC of 0.71. We also tried including non-surface residues in the evaluation as well, which improved the AUC score to 0.75. We find these results surprising, as structure-based methods are generally considered superior to sequence-based methods, indicating the sheer power of using protein language models for B-cell epitope prediction.

4 DISCUSSION

Deep learning methods, such as ESM-2 and AlphaFold, are revolutionizing the field of biology at large, and changing the role of computational tools in numerous tasks. In this paper, we demonstrate that protein
Benchmarking on surface residues of the epitope

BepiPred-3.0 was benchmarked against its predecessors (BepiPred-1.0 and 2.0) on the three external test sets. To this aim, it is possible to frame the epitope unlabeled or potentially positive residues due to data not being false positives, but rather should be considered that in many cases, regions predicted to be epitopes may fraction of all possible pathogenic proteins, and of the antibodies that target them. Due to the underrepresentation of observed epitopes in current datasets, we expect that in many cases, regions predicted to be epitopes may not be false positives, but rather should be considered unlabeled or potentially positive residues due to data paucity. To this aim, it is possible to frame the epitope prediction problem as a positive and unlabeled (PU) training problem. Moreover, we can also argue that the current AUC of the model, around 75%, is an under-estimation, and only by collecting more experimental data will it be possible to fully assess how close we are to the upper limit of the B-cell epitope prediction tools.

It is also interesting to note that a major progress in this class of predictors would be the possibility to include the sequences of individual antibodies, or of antibody libraries, for which we want to identify all the potential epitopes. Language models provide an elegant way to include them in the prediction, by encoding the antibodies together with the antigen. As more data will be available, it will be interesting to test if LMs can also provide a solution to this fundamental problem in immunology and biotechnology.

To conclude, BepiPred-3.0 is available as a web server and as a standalone package. It is easy to use for experts and non-experts alike, and provides state-of-the-art B-cell epitope predictions that will be fundamental to tasks of primary medical and societal importance, such as vaccine development and antibody engineering.

**AUTHOR CONTRIBUTIONS**

Joakim Nøddeskov Clifford: Data curation (lead); formal analysis (lead); investigation (lead); methodology (equal); software (lead); visualization (equal); writing – original draft (lead); writing – review and editing (equal).

Magnus Haraldson Høie: Conceptualization (supporting); investigation (supporting); methodology (supporting); software (supporting); supervision (equal); validation (equal); writing – review and editing (equal).

Sebastian Deleuran: Methodology (supporting); supervision (supporting); visualization (equal); writing – review and editing (equal).

**Note:** These were the 5 and 15 antigen external test sets (TS1 and TS2), as well as the external IEDB test set containing 3,560 sequences (IEDB). The AUC and AUC10 values for BepiPred-1.0 and BepiPred-2.0 on TS1 were extracted from their respective publications. The remaining evaluations for BepiPred-2.0 and BepiPred-3.0 were done using local instalments. The best overall AUC and AUC performance is marked in bold. The data used to train BepiPred-1.0 was constructed by mapping known epitope peptides onto protein sequences. For BepiPred 2.0 and 3.0, the training data was extracted from crystal structures of antibody-antigen complexes. A HMM, RFC, and LM were used for BepiPred-1.0, 2.0, and 3.0, respectively.

Abbreviations: HMM, hidden markov model; LM, protein language model; RFC, random forest classifier.

**TABLE 6** Benchmarking on surface residues of the epitope3D external test set of 45 antigens

| Models       | AUC | AUC10 | Data    | Method | Test  |
|--------------|-----|-------|---------|--------|-------|
| BepiPred-3.0 | 0.71|       | Peptides| HMM    | TS1   |
| Epitope3D    | 0.59|       | PDB     | RFC    | TS2   |
| BepiPred-2.0 | 0.58|       | Peptides| LM     | TS1   |
| Seppa-3.0    | 0.55|       | PDB     | LM     | TS2   |
| Discotope-2.0| 0.51|       | LM      | LM     | IEDB  |
| Ellipro      | 0.49|       |         | LM     | IEDB  |

LMs can vastly improve B-cell epitope prediction, and using only the antigen sequence as an input, outperform existing tools, including structure-based ones. We can envision that, by using a similar approach on structure based embeddings calculated on solved antigen structures, or on structural models created using AlphaFold, it will be possible to further improve the current results.

We also want to argue on the fact that the current BepiPred-3.0 results are likely affected by the limited availability of experimental structures. The available solved antibody-antigen complexes are just a minute fraction of all possible pathogenic proteins, and of the antibodies that target them. Due to the underrepresentation of observed epitopes in current datasets, we expect that in many cases, regions predicted to be epitopes may not be false positives, but rather should be considered unlabeled or potentially positive residues due to data paucity. To this aim, it is possible to frame the epitope

Table 6: Benchmarking on surface residues of the epitope3D external test set of 45 antigens

| Models       | AUC | AUC10 | Data     | Method | Test  |
|--------------|-----|-------|----------|--------|-------|
| BepiPred-3.0 | 0.71|       | Peptides | HMM    | TS1   |
| Epitope3D    | 0.59|       | PDB      | RFC    | TS2   |
| BepiPred-2.0 | 0.58|       | Peptides | LM     | TS1   |
| Seppa-3.0    | 0.55|       | PDB      | LM     | TS2   |
| Discotope-2.0| 0.51|       | LM       | LM     | IEDB  |
| Ellipro      | 0.49|       |          | LM     | IEDB  |

Note: The AUC values for epitope3D were extracted from its publication. The remaining evaluations for BepiPred-3.0 were done using local installations. The best overall AUC and AUC performance is marked in bold. The data used to train BepiPred-1.0 was constructed by mapping known epitope peptides onto protein sequences. For BepiPred 2.0 and 3.0, the training data was extracted from crystal structures of antibody-antigen complexes. A HMM, RFC, and LM were used for BepiPred-1.0, 2.0, and 3.0, respectively. Abbreviations: HMM, hidden markov model; LM, protein language model; RFC, random forest classifier.
and editing (supporting). Morten Nielsen: Methodology (supporting); supervision (equal); validation (equal); writing – review and editing (equal). Paolo Marcatili: Conceptualization (lead); data curation (equal); formal analysis (supporting); investigation (supporting); methodology (lead); project administration (equal); resources (equal); software (supporting); supervision (lead); validation (equal); writing – original draft (equal); writing – review and editing (equal).

DATA AVAILABILITY STATEMENT
BepiPred-3.0 is available both as a web server, and as stand-alone software including datasets at the below URLs. Webserver, DTU: https://services.healthtech.dtu.dk/service.php?BepiPred-3.0; Webserver, biolib: https://biolib.com/DTU/BepiPred-3/

ORCID
Joakim Nøddeskov Clifford https://orcid.org/0000-0002-8126-9209

REFERENCES
1. Andersen P, Nielsen M, Lund O. Prediction of residues in discontinuous b-cell epitopes using protein 3d structures. Protein Sci. 2006;15(11):2538–2567.
2. Bateman A, Martin MJ, Orchard S, et al. Uniprot: The universal protein knowledgebase in 2021. Nucleic Acids Res. 2021; 49(1):D480–D489.
3. Behmard E, Soleymani B, Najafi A, Barzegari E. Immunoinformatic design of a covid-19 subunit vaccine using entire structural immunogenic epitopes of sars-cov-2. Sci Rep. 2020;10(1): 20864.
4. Berman HM, Westbrook J, Feng Z, et al. The Protein Data Bank. Nucleic Acids Res. 2000;28(1):235–242.
5. Brandes N, Ofer D, Peleg Y, Rappoport N, Linial M. Proteinbert: A universal deep-learning model of protein sequence and function. Bioinformatics. 2022;38(8):2102–2110.
6. Edgar RC. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. Nucleic Acids Res. 2004;32(5):1792–1797.
7. Da Silva BM, Myung Y, Ascher DB, Pires DE. Epitope3d: A machine learning method for conformational B-cell epitope prediction. Brief Bioinform. 2022;23(1):bbab423.
8. Erik J, Lund O, Nielsen M. Improved method for predicting linear B-cell epitopes. Immunome Res. 2020;6(2):2.
9. Evans R, O'Neill M, Pritzel A, et al. Protein complex prediction with alphafold-multimer. bioRxiv. 2021. doi:10.1101/2021.10.04.463034
10. Haie MH, Kiehl EN, Petersen B, et al. NetSurfP-3.0: Accurate and fast prediction of protein structural features by protein language models and deep learning. Nucleic Acids Res. 2022;50: gkaa439.
11. Jespersen MC, Peters B, Nielsen M, Marcatili P. Bepipred-2.0: Improving sequence-based b-cell epitope prediction using conformational epitopes. Nucleic Acids Res. 2017;45(W1): W24–W29.
12. Jumper J, Evans R, Pritzel A, et al. Highly accurate protein structure prediction with alphafold. Nature. 2021;596(7873): 583–589.
13. Kingma DP, Ba J. Adam: A method for stochastic optimization. 2017.
14. Klausen MS, Anderson MV, Jespersen NC, Nielsen M, Marcatili P. Lyra, a webserver for lymphocyte receptor structural modeling. Nucleic Acids Res. 2015;43(W1):W349–W355.
15. Klausen MS, Jespersen MC, Nielsen H, et al. Netsurf-p2.0: Improved prediction of protein structural features by integrated deep learning. Proteins Struct Funct Bioinform. 2019;87(6): 520–527.
16. Kozlova EEG, Cerf L, Schneider FS, et al. Computational b-cell epitope identification and production of neutralizing murine antibodies against atrolysin-i. Sci Rep. 2018;8(1):14904.
17. Krangel JM, Nielsen M, Padkjær SB, Lund O. Structural analysis of b-cell epitopes in antibody: Protein complexes. Mol Immunol. 2013;53(1–2):24–34.
18. Kulkarni-Kale U, Bhosle S, Kolaskar AS. Cep: A conformational epitope prediction server. Nucleic Acids Res. 2005;33(2): W168–W171.
19. Lin Z, Akin H, Rao R, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. bioRxiv. 2022. doi:10.1101/2022.07.20.500902
20. Mucci JJ, Carmona S, Volovich R, Altcheh J, Bracameod EM, Marco J, Nielsen MA, Buscaglia C, Agüero F. Next-generation ELISA diagnostic assay for chagas disease based on the combination of short peptidic epitopes. PLOS Negl Trop Dis. 2017;11(10):e0005972.
21. Ponomarenko J, Bui HH, Li W, et al. Ellipro: A new structure-based tool for the prediction of antibody epitopes. BMC Bioinform. 2008;9(1):514.
22. Rives A, Meier J, Sercu T, et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. Proc Natl Acad Sci U S A. 2021;118(15): e2016239118.
23. Saha S, Raghava GP. Prediction of continuous b-cell epitopes in an antigen using recurrent neural network. Proteins Struct Funct Genet. 2013;53(1):242.
24. Sanchez-Trincado JL, Gomez-Perosanz M, Reche PA. Fundamentals and methods for t- and b-cell epitope prediction. J Immunol Res. 2017;2017:2680160.
25. Shirai H, Prades C, Vita R, et al. Antibody informatics for drug discovery. Biochem Biophys Acta Protiomes. 2014; 1844(11):2002–2015.
26. Singh H, Ansari HR, Raghava GP. Improved method for linear b-cell epitope prediction using antigen’s primary sequence. PLoS ONE. 2013;8(5):e62216.
27. Steinegger M, Söding J. Mmesq2 enables sensitive protein sequence searching for the analysis of massive data sets. Nat Biotechnol. 2017;35(11):1026–1028.
28. Strokach A, Becerra D, Corbi-Verge C, Perez-Riba A, Kim PM. Fast and flexible protein design using deep graph neural networks. Cell Syst. 2020;11(4):402–411.e4.
29. Sun J, Wu D, Xu T, et al. Seppa: A computational server for spatial epitope prediction of protein antigens. Nucleic Acids Res. 2009;37(2):W612–W616.
30. Sweredoski MJ, Baldi P. Pepito: Improved discontinuous b-cell epitope prediction using multiple distance thresholds and half sphere exposure. Bioinformatics. 2008;24(12):1459–1460.
31. Kringelum JV, Lundegaard C, Lund O, Nielsen M. Reliable B cell epitope predictions: Impacts of method development and improved benchmarking. PLoS Comput Biol. 2012;8(12):e1002829.
32. Vita R, Mahajan S, Overton JA, et al. The immune epitope database (iedb): 2018 update. Nucleic Acids Res. 2019;47(1):D339–D343.
33. Yao B, Zhang L, Liang S, Zhang C. Svmtrip: A method to predict antigenic epitopes using support vector machine to integrate tri-peptide similarity and propensity. PLoS One. 2012;7(9):e45152.
34. Zhou C, Chen Z, Zhang L, et al. Seppa 3.0—Enhanced spatial epitope prediction enabling glycoprotein antigens. Nucleic Acids Res. 2019;47(1):W388–W394.

SUPPORTING INFORMATION
Additional supporting information can be found online in the Supporting Information section at the end of this article.

How to cite this article: Clifford JN, Høie MH, Deleuran S, Peters B, Nielsen M, Marcatili P. BepiPred-3.0: Improved B-cell epitope prediction using protein language models. Protein Science. 2022;31(12):e4497. https://doi.org/10.1002/pro.4497