Bioinformatics, 37(4), 2021, 448–455
doi: 10.1093/bioinformatics/btaa773
Advance Access Publication Date: 11 September 2020
Original Paper

Sequence analysis

EpiDope: a deep neural network for linear B-cell epitope prediction

Maximilian Collatz[1,*,†], Florian Mock [1,*,†], Emanuel Barth[1,2], Martin Hölzer[1,3], Konrad Sachse[1] and Manja Marz[1,2,3,4,*]

[1]RNA Bioinformatics /High Throughput Analysis, Faculty of Mathematics and Computer Science, [2]Bioinformatics Core Facility Jena, Friedrich Schiller University Jena, Jena 07743, Germany, [3]RNA Bioinformatics/High Throughput Analysis, European Virus Bioinformatics Center (EVBC), Jena 07743, Germany and [4]RNA Bioinformatics/High Throughput Analysis, FLI Leibniz Institute for Age Research, Jena 07745, Germany

*To whom correspondence should be addressed.

†The authors wish it to be known that, in their opinion, the first two authors should be regarded as Joint First Authors.

Associate Editor: Cowen Lenore

Received on April 29, 2020; revised on August 6, 2020; editorial decision on August 23, 2020; accepted on September 1, 2020

Abstract

Motivation: By binding to specific structures on antigenic proteins, the so-called epitopes, B-cell antibodies can neutralize pathogens. The identification of B-cell epitopes is of great value for the development of specific serodiagnostic assays and the optimization of medical therapy. However, identifying diagnostically or therapeutically relevant epitopes is a challenging task that usually involves extensive laboratory work. In this study, we show that the time, cost and labor-intensive process of epitope detection in the lab can be significantly reduced using *in silico* prediction.

Results: Here, we present EpiDope, a python tool which uses a deep neural network to detect linear B-cell epitope regions on individual protein sequences. With an area under the curve between 0.67 ± 0.07 in the receiver operating characteristic curve, EpiDope exceeds all other currently used linear B-cell epitope prediction tools. Our software is shown to reliably predict linear B-cell epitopes of a given protein sequence, thus contributing to a significant reduction of laboratory experiments and costs required for the conventional approach.

Availabilityand implementation: EpiDope is available on GitHub (http://github.com/mcollatz/EpiDope).

Contact: maximilian.collatz@uni-jena.de or florian.mock@uni-jena.de or manja@uni-jena.de

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

The public health system is highly dependent on the use of vaccines to protect the population from a range of dangerous infectious diseases. Through decades of systematic vaccination, diseases like measles, mumps, rubella, pertussis, poliomyelitis, diphtheria, tetanus and others have been largely eradicated (Rappuoli *et al.*, 2014; Van Panhuis *et al.*, 2013). Vaccination is also an efficient approach to avoid or reduce prescriptions of antibiotics and, as a consequence, minimize the emergence of ever more multi-resistant strains of microbial pathogens. To assess the degree of protection of vaccination at population level, faster and more efficient serological tools need to be developed. They should be capable of identifying geographical and social heterogeneities in the diversity of population immunity (Arnold *et al.*, 2018; Metcalf *et al.*, 2016).

In addition, it is important to know the status of a patient's immunization to avoid unnecessary vaccinations. In cases where this is not or only incompletely documented, various tests can be used to determine which specific immunities already exist and which vaccinations are missing. These tests are not only slow and expensive, but usually also use whole cell antigens to detect antibodies, which limits their specificity (Sachse *et al.*, 2018).

B-cell antibodies of the immune system of a host are able to detect certain exposed amino acids and subsequently bind the corresponding antigenic proteins. These bound protein regions are called epitopes and represent the interface between infection and immune response (Kringelum *et al.*, 2013; Van Regenmortel, 2009). The antibody part that binds the epitope is called paratope. Epitopes themselves are not intrinsic features of a protein, but rather relational units defined by the interaction with a binding paratope. This relatively vague definition makes it a challenging task to predict epitopes *in silico* (Kringelum *et al.*, 2013; Sanchez-Trincado *et al.*, 2017). Furthermore, epitopes are divided into ~10% linear and 90% conformational epitopes (Zhang *et al.*, 2014). Linear epitopes consist of a contiguous piece of amino acids and conformational epitopes consist of atoms of surface residues that come together by protein folding. In this study, we will focus on the prediction of linear B-cell epitopes.

A frequently used tool to predict linear B-cell epitopes is BepiPred2 (Jespersen *et al.*, 2017). Jespersen *et al.* state an area under the curve (AUC) of 0.57 for their tool's receiver operating characteristic (ROC) curve. This is on par with other prediction scales provided by the 'Immune Epitope Database' IEDB (http://tools.iedb.org/main/bcell/) and demonstrates the difficulty of *in silico* epitope identification. Therefore, we developed EpiDope, a tool based on deep neural networks (DNN) to detect epitopic regions in proteins based on their primary amino acid sequence. Previous tools mostly predict using a sliding window and a likelihood estimation approach. These approaches vary from simple statistical methods to classical techniques of machine learning like SVN or random forests up to DNNs.

DNNs are often used in complex classification problems with limited knowledge about useful features of the objects to be classified. With sufficient data, a DNN can automatically recognize appropriate classification features, making DNNs very suitable for the prediction of linear B-Cell epitopes (Bengio, 2012).

We will show that our DNN-based program EpiDope succeeds in identifying linear B-cell epitopes with a ROC AUC of 0.67 ± 0.07, which significantly exceeds previous methods. EpiDope achieves this using context-sensitive embeddings for the amino acids. Context-sensitive embeddings provide a numerical representation that encodes local information, such as which amino acid is encoded, and information of the context, in this case, the full protein. Therefore, while predicting using a sliding window, not only local features are provided but also context. Hence a higher amount of information than just the single window is provided to the neural network. This is a clear advantage over competing methods.

The high predictive power of EpiDope helps to considerably reduce the number of potential linear epitopes to be validated experimentally and, thus, can accelerate the development of serological assays and immunotherapeutic approaches.

Table 1. Comparison of the original dataset provided by Jespersen *et al.* (2017), and the redundancy reduced data used as training dataset

	Original	Reduced
No. of verified regions	30 556	24 610
No. of epitopes	11 834	8519
No. of non-epitopes	18 722	16 091
Median length	15	15
Avg. length	13.50	13.27
Avg. length epitope	13.99	13.88
Avg. length non-epitope	13.20	12.95

2 Materials and methods

2.1 Data

The 'IEDB Linear Epitope Dataset' (available at http://www.cbs.dtu.dk/services/BepiPred/download.php), which was also used for the evaluation of the B-cell epitope prediction tool BepiPred2 (Jespersen *et al.*, 2017), served as a training basis for our DNN. It consists of 30 556 protein sequences, in which each sequence contains a marked region, in the following called 'verified regions', that represents an experimentally verified epitope or non-epitope. The subset of epitopes has an average length of 13.99, whereas the subset of non-epitopes has an average length of 13.20 (see Table 1).

2.2 Data preparation

To ensure the best possible training basis for the DNN, we preprocessed the dataset in several steps (see Fig. 1). First, we merged identical protein sequences while keeping the information about their verified regions, resulting in a reduced dataset containing 3158 proteins preserving all 30 556 verified regions (Fig. 1A and B). Second, to reduce redundancy by non-identical but highly similar protein sequences, we clustered all sequences with cd-hit (Li and Godzik, 2006) using an identity threshold of 0.8 (Fig. 1C). This resulted in 1798 protein sequence clusters. From each cluster, only the protein sequence containing the largest number of verified regions was retained, reducing the number of verified regions by 19.46% to 24 610 (see Table 1, Fig. 1D). This reduces the number of very similar and thus overrepresented proteins in the data, as these might bias the training of the DNN.

The clustering step was repeated on the reduced protein sequences using an identity threshold of 0.5, resulting in 1378 sequence clusters. These clusters were then used to build the 10-fold cross-validation for the DNN, where the data were divided into 10 equally sized subsets according to the number of clusters. This sequence

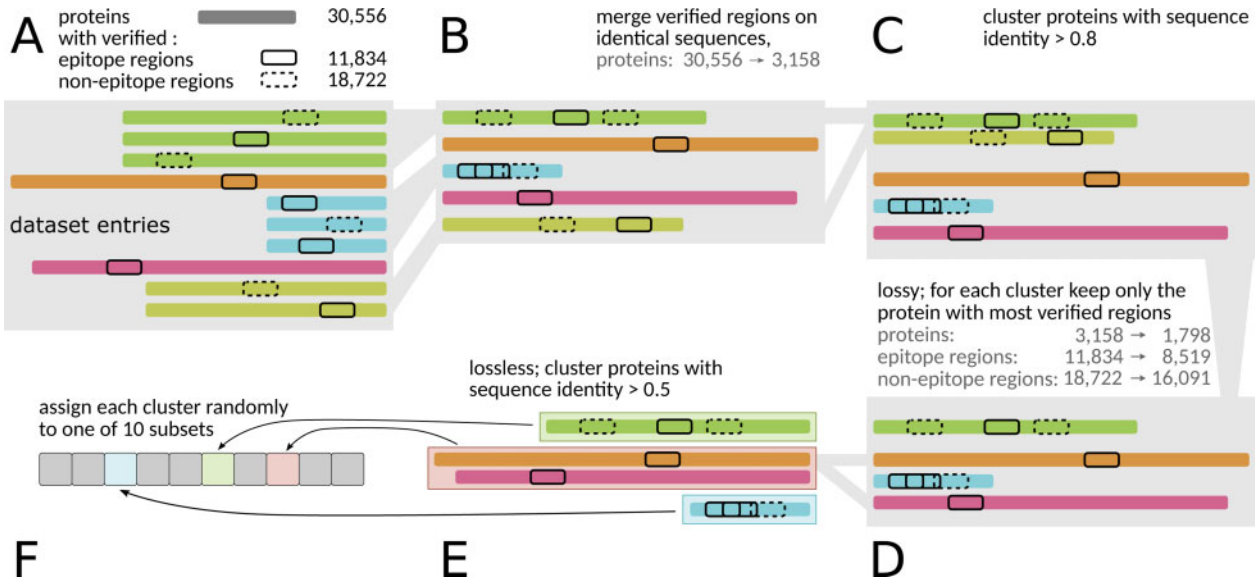

Fig. 1. The data preparation to generate the training and validation set. First all identical amino acid sequences from the raw dataset (**A**) are merged while preserving the verified regions (**B**). In the next step (**C**), all remaining sequences are clustered with a sequence identity of 0.8 and higher. For each of the resulting clusters only the sequence with the highest number of verified regions is selected (**D**). These selected sequences are then clustered again, this time with a sequence identity of 0.5 and higher (**E**). Each cluster is then assigned to only 1 of 10 different subsets for the 10-fold cross-validation (**F**). The data preparation generates training and test data with a low sequence identity of <0.5 while limiting the loss of potential sequences

reduction approach was implemented with the following advantage in mind: the DNN training, and test sets share no proteins with a sequence identity of more than 0.5, ensuring that both sets are as independent as possible from each other, avoiding the DNN from simply memorizing specific epitopes.

2.3 Deep neural network architecture

We compared several DNN architectures, including different ordering, layer types and numbers of nodes. The DNN architecture used in EpiDope consists of two parts (see Fig. 2). The first part (Fig. 2A) uses context-sensitive embeddings of amino acids produced by an ELMo DNN. This DNN was previously trained by Heinzinger *et al.* to encode various chemical, physical and structural information and was demonstrated to be usable for various high-quality predictions on protein sequences (Heinzinger, 2019). The ELMo DNN consists of a CharCNN layer, which in our case, results in a 1024 dimensional, non-context-sensitive embedding for each amino acid. The next two layers consist of 1024 bi-directional LSTM nodes each, which introduces context-specific information by sequentially processing the protein sequence. During this sequential processing, the hidden states of the LSTM nodes are concatenated, resulting in a 1024 dimensional vector per layer. These three layers are then summed up, resulting in a single vector of length 1024. With this each amino acid of a given protein sequence is encoded in a 1024 dimensional vector which encodes the chemical, physical and structural information. These embeddings are the input for a bidirectional LSTM layer with 2×5 nodes (Hochreiter and Schmidhuber, 1997), followed by a dense layer containing 10 nodes.

The second part of our DNN architecture (Fig. 2B) encodes each amino acid into a vector of length 10. This embedding is not context-sensitive and is trained together with the rest of the DNN. This embedding layer is connected to a bidirectional LSTM layer with 2×10 nodes, again followed by a dense layer with 10 nodes.

Both dense layers are then connected with an additional dense layer containing 10 nodes, which is concluded by the output layer with two nodes representing the two classes, *epitope* and *non-epitope*.

Note that we do not fine-tune the ELMo DNN. Due to the high number of parameters used by ELMo, the limited number of samples for our classification task and to avoid overfitting, we used the weights of the ELMo DNN during the training as given. We combined it with a more traditional approach based solely on the local structure. Thus, the neural network develops embeddings that are explicitly trained for the task of epitope prediction (Fig. 2B). This leads to an architecture that uses embeddings with a broad understanding of proteins in combination with embeddings that are specifically trained to identify epitopes locally.

Note, the number of nodes in this DNN is comparatively low. However, due to the high dimensionality of the context-sensitive embedding (1024 per amino acid), the number of parameters tuned by the DNN is substantial.

2.4 Evaluation on new data

We created a new dataset that contains all verified regions from the IEDB (Vita *et al.*, 2019) that were neither included in the BepiPred2 dataset nor the LBtope 'variable length' dataset (as of November 27, 2019). All areas that were tested positive by at least two assays were stored as epitopes, whereas all areas tested in at least two assays and not tested positive in any assay were stored as non-epitopes. These are the same conditions that were used to create the BepiPred2 dataset (Jespersen *et al.*, 2017). Thus this dataset is entirely independent of our training dataset. This second dataset is from now on called evaluation dataset.

Next we clustered the evaluation dataset, alike the training set, with a sequence identity of 0.8. For each cluster we only kept the sequence with the highest number of validated regions. This reduces the overrepresentation of very similar proteins in the data. As the IEDB also consists of short validated regions, of which a significant number are false-negatives due to weak antibody binding while *in vitro* validation, we filtered for validated regions with length $\geq$ 12 amino acids, as suggested by Rahman *et al.* (2016).

2.5 Evaluation of epitope prediction approaches

We only considered epitope prediction tools that were usable at the time of writing and allow to process multiple protein sequences in a high-throughput manner. As otherwise a statistically solid comparison on large datasets is nearly impossible. Therefore, BcePred (Saha et al., 2004), iBCE-EL (Manavalan *et al.*, 2018), ABCpred (Saha and Raghava, 2006), COBEpro (Sweredoski and Baldi, 2009) and SVMTriP (Yao *et al.*, 2012) could not be compared.

Nonetheless, we compared our tool EpiDope against seven frequently used tools for linear B-cell epitope prediction mainly from the IEDB (http://tools.iedb.org/bcell/) in their latest version. Namely Bepipred 2.0 (Jespersen *et al.*, 2017), LBtope (Singh *et al.*, 2013), Parker Hydrophilicity prediction (Parker *et al.*, 1986), Chou and Fasman beta turn prediction (Pellequer *et al.*, 1993), Emini surface accessibility scale (Emini *et al.*, 1985), Kolaskar and Tongaonkar antigenicity scale (Kolaskar and Tongaonkar, 1990) and the independent prediction tool for intrinsically unstructured protein regions IUPred (Dosztányi *et al.*, 2005).

For each tool, we calculated the corresponding prediction values for the entire protein sequence on each amino acid and sliced out the verified regions. For each of the sliced regions, we calculated the average score as the prediction score to discriminate between epitope and non-epitope. For the antigenicity scale approach, we

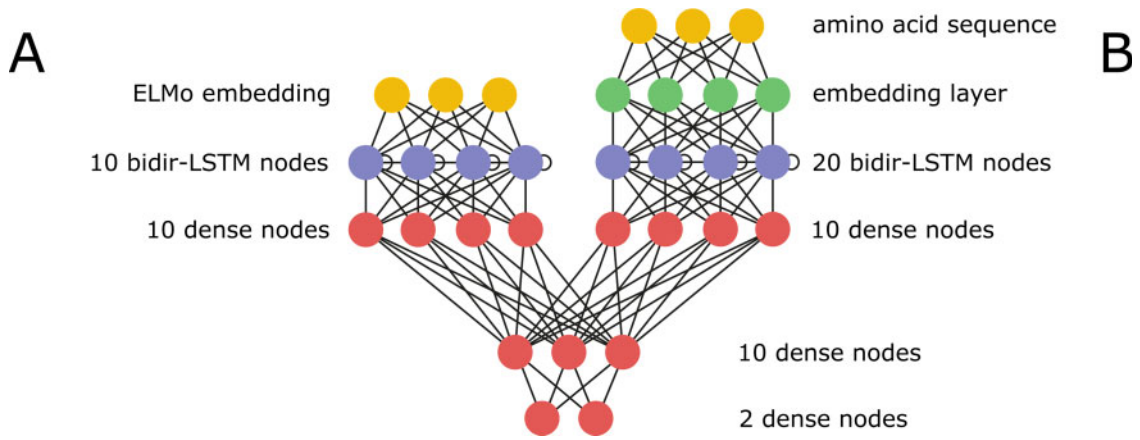

Fig. 2. The DNN architecture of EpiDope consists of two parts. The first (**A**) uses context-sensitive ELMo embeddings for the epitope prediction. These embeddings are previously calculated by an ELMo DNN. The second part (**B**) uses classic embeddings for prediction. These classic embeddings are not context-sensitive. Both parts are then joined to predict two classes, epitopes and non-epitopes

additionally subtracted the mean of the full protein as suggested in the original paper (Kolaskar and Tongaonkar, 1990).

We compared all the tools using ROC and Precision–Recall curves (Brown and Davis, 2006; Fawcett, 2006), which is good practice for evaluating machine learning approaches in general and unbalanced datasets in particular (Branco *et al.*, 2015). For the EpiDope ROC curve, any subset of the 10-fold cross-validation was predicted by the model that did not include this subset in its training data. We calculated one ROC curve per subset. This resulted in 10 ROC curves, on which the mean ROC curve was calculated. In the mean ROC curve each of the 10 models had an equal influence. The ROC curves of competing tools are calculated on the same data without having to combine multiple predictions, as these tools and scores did not use this data for training.

For the Precision–Recall curve of EpiDope, as with the ROC curve, each subset was predicted by the model that did not use it in the training set. Next, we calculate the Precision–Recall curve on all 10 subsets at once rather than calculating 10 Precision–Recall curves. Calculating the mean curve from 10 Precision–Recall curves could change the balance between the two classes and as such bias the Precision–Recall curve. All other Precision Recall curves were calculated equally.

The ROC curves as well as the Precision–Recall curves were calculated using scikit-learn (Pedregosa *et al.*, 2011).

3 Results and discussion

3.1 The training dataset represents large pathogen variety

To achieve a bias-free prediction of epitopes, it is essential to have a large variety of known epitopes from evolutionarily distinct organisms in the training set. Therefore, we initially analyzed the taxonomic origin of the protein sequences provided by the IEDB and visualized the results with the online tool Pavian metagenomics data explorer (Breitwieser, Florian and Salzberg, 2020), see Figure 3.

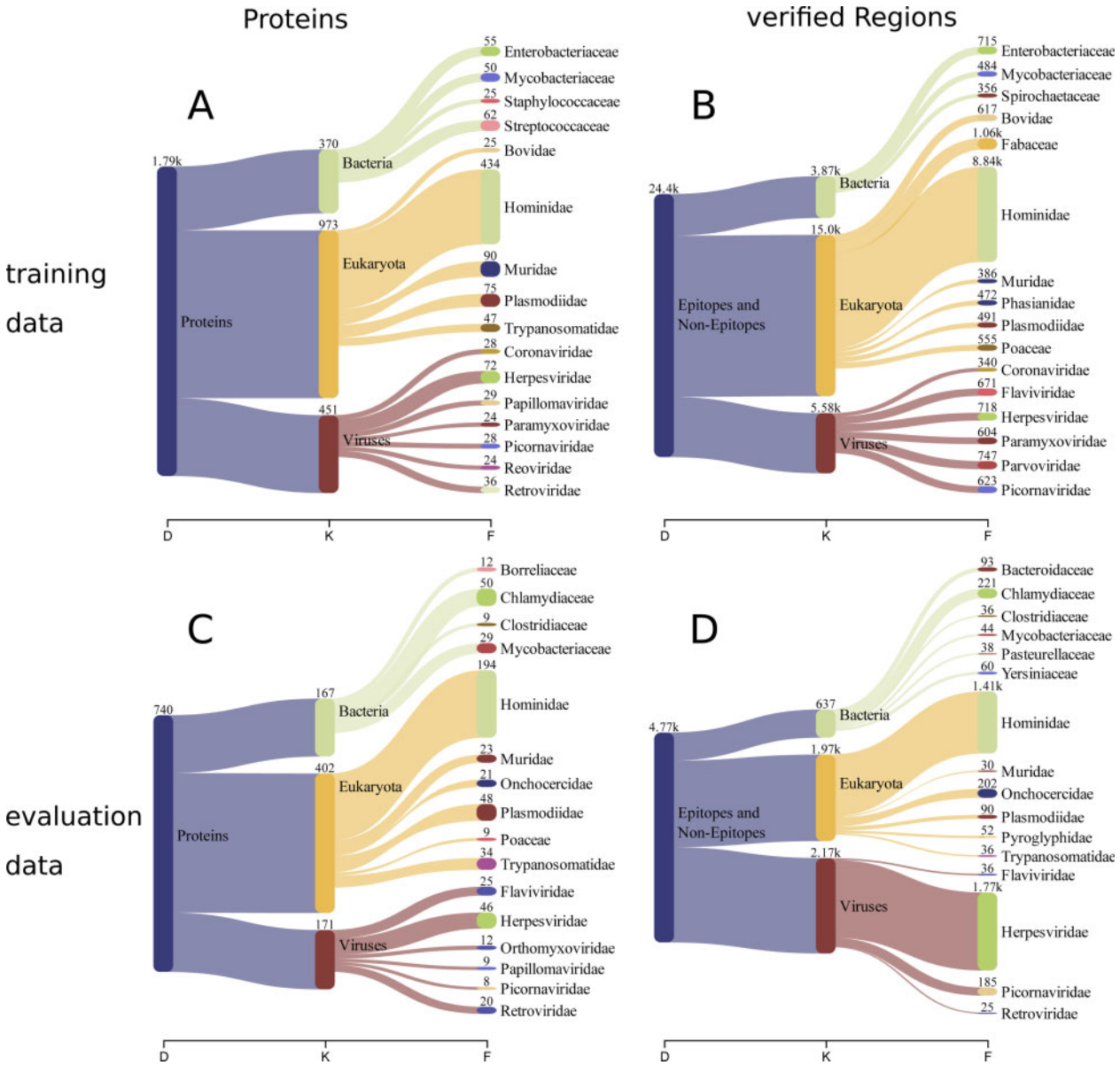

Fig. 3 Taxonomic origin of the 16 most common families of the training dataset (**A** and **B**) and evaluation dataset (**C** and **D**). A shows the origin of all 1798 proteins in the training dataset, covering a wide variety of superkingdoms and families. (B) The taxonomic origin of the epitopes and non-epitopes, with 15.7% from Bacteria, 61% Eukaryota and 23.4% from Viruses. Taxonomic origin of the sequences used in the evaluation dataset: (C) the origin of all 740 proteins and (D) the taxonomic origin of the epitopes and non-epitopes, with 13.2% from Bacteria, 41.3% from Eukaryota and 45.5% from Viruses

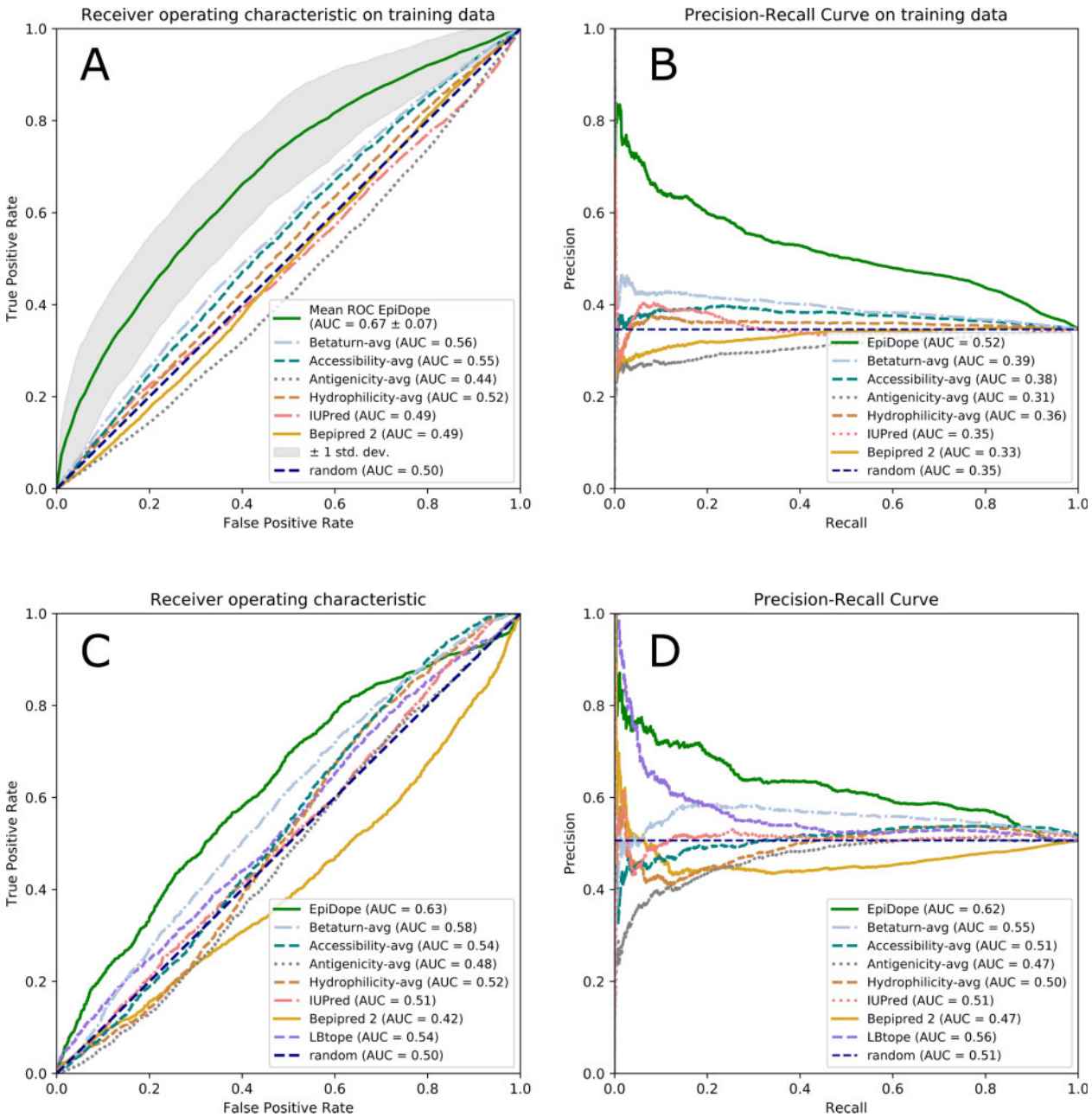

Fig. 4. (**A**) The comparison of the ROC curves between the evaluated tools based on the training dataset. For the mean EpiDope ROC curve, every subset of the 10-fold cross-validation was predicted by the model that did not include this subset in its training dataset. This resulted in 10 ROC curves for which the mean ROC curve was calculated (displayed in green) and the standard deviation area (grey). The other ROC curves were calculated on the same data without having to combine multiple predictions, as these tools and scores did not use this data for training. In (**B**), the precision–recall curve shows the trade-off between the number of false positive predictions compared to the number of false negative predictions. This is important as the number of epitopes and non-epitopes are not balanced in the dataset. (**C**) The ROC curves for the evaluation dataset, which consists of all currently (as of 27.11.2019) well verified regions that are not present in the training dataset. (**D**) The precision–recall curve on the evaluation dataset

Since the number of verified regions per protein varies, we analyzed them separately. Our filtered dataset contains 1798 proteins with 24 610 epitopes and non-epitopes assigned to them.

From all 1798 proteins used while training, 550 were also used for training the ELMo DNN. Identical proteins do not pose a problem, because the training of the ELMo-DNN does not contain any information about epitopes or non-epitopes, only the protein sequence itself is given. Hence using identical proteins does not lead to information leakage.

At the protein level, the data consist of 177 different families, from which the most common 16 are shown in Figure 3A. The 39 families of Bacteria account for 20.6%, the 85 families of Eukaryota for 54.2% and the 53 families of Viruses for 25.1% of all proteins in the training data. At the level of the verified regions, we observed a slight shift, so that Eukaryota with 61.0% are even more pronounced than the Bacteria with 15.7% and the Viruses with 23.4%. Overall, the training data display a clear degree of taxonomic diversity.

3.2 Evaluation of EpiDope via 10-fold cross-validation

As described in the Section 2, we evaluated the performance of EpiDope with competing methods on the training dataset, using 10-fold crossvalidation. Each of the 10 subsets consists of proteins that

Table 2. Comparison of the AUC and AUC10% of the ROC curve calculated for multiple tools, on the training dataset. AUC shows the value on the complete ROC curve while AUC10% is the area for a False Positive Rate (FPR)<0.1 corrected by multiplying with 10. EpiDope is compared using 10-fold cross validation.

Training dataset		
Tool	AUC	AUC10%
EpiDope	0.670	0.151
Betaturn	0.562	0.070
Accessibility	0.548	0.058
Antigenicity	0.443	0.034
Hydrophilicity	0.521	0.053
IUPred	0.489	0.060
Bepipred2	0.490	0.038
random	0.500	0.050

Note: AUC shows the value on the complete ROC curve while AUC10% is the area for a False Positive Rate (FPR) <0.1 corrected by multiplying with 10. EpiDope is compared using 10-fold cross validation. The best performing method is highlighted in grey.

Table 3. Comparison of the AUC and AUC10% of the ROC curve calculated for multiple tools on the evaluation dataset. AUC shows the value on the complete ROC curve, while AUC10% is the area for a false-positive rate (FPR)<0.1 corrected by multiplying with 10.

Evaluation data		
Tool	AUC	AUC10%
EpiDope	0.625	0.120
Betaturn	0.579	0.054
Accessibility	0.537	0.039
Antigenicity	0.477	0.023
Hydrophilicity	0.517	0.040
IUPred	0.514	0.044
Bepipred2	0.416	0.050
LBtope	0.542	0.083
Random	0.500	0.050

Note: AUC shows the value on the complete ROC curve, while AUC10% is the area for a false-positive rate (FPR) < 0.1 corrected by multiplying with 10. The best performing method is highlighted in grey.

have a sequence identity below 0.5 to all proteins in the other 9 subsets (for details, see the Section 2.2).

The ROC-curve (Fig. 4A) shows that EpiDope is with an AUC of 0.67 ± 0.07 clearly outperforming competing prediction approaches, all of which achieved an AUC $\leq$ 0.56. Despite multiple requests from us to the developers of BepiPred2, we could not confirm their stated AUC performance of 0.57 on our reduced and less redundant dataset (see Table 1).

Usually, it is not necessary to find all immunodominant epitopes in a proteome. Instead, only a small number of functioning epitopes are required, ideally without having to scan a large number of regions. Therefore the prediction performance in the highest-rated regions is of particular interest. To evaluate the performance of these regions, we used AUC10%. The AUC10% is the AUC of the ROC curve for a False Positive Rate (FPR) < 0.1, normalized, as suggested by the BepiPred2 developers by multiplying with 10 (Jespersen *et al.*, 2017).

EpiDope reaches a high AUC10% of 0.151, compared to the second best method (Betaturn) with an AUC10% of 0.070 (see Table 2). Note that Jespersen *et al.* state an AUC10% of 0.08 for BepiPred2 (Jespersen *et al.*, 2017) on the original unreduced dataset. The performance of EpiDope relies on the high precision of the top predictions, notable also in the Precision–Recall curve, see Figure 4B.

3.3 Analysis of the evaluation dataset

In comparison with the training dataset the evaluation dataset is of reduced size (see Fig. 3B and D), with the training dataset having over 24 600 verified regions and the evaluation dataset having 4767 verified regions. The evaluation dataset has a lower proportion of eukaryotic verified regions (41.3%) and a higher proportion of viral verified regions (45.5%) compared the training dataset (eukaryotic 61.0%, viral 23.4%).

From the 16 most common families (Fig. 3D), 8 are not in the most common families of the training dataset (Fig. 3A). These families are *Bacteroidetes*, *Chlamydiaceae*, *Clostridiaceae*, *Pasteurellaceae*, *Yersiniaceae*, *Pyroglyphidae*, *Trypanosomatidae* and *Retroviridae*, representing between 0.5 and 4.2% of the evaluation dataset.

Furthermore, the ratio of epitope regions versus non-epitope regions changed dramatically, from 35% epitopes in the training dataset to 51% in the evaluation dataset.

As with the training dataset, we calculated the predictions of several tools (see Section 2.5 for further details) for the verified regions. We calculated the ROC curve and the Precision–Recall curve (see Fig. 4C/D) and evaluated the AUC and AUC10% of EpiDope and the competing tools (see Table 3).

The ROC-curve (see Fig. 4C) shows that EpiDope is, with an AUC of 0.63, surpassing competing prediction approaches. We can observe a similar result for nearly all tools in comparison with the evaluation on the training dataset. The AUC of EpiDope decreased by 0.045, which is within the calculated standard deviation of 0.071.

The AUC10% indicates the usability of all methods for practical applications. EpiDope outperforms all competing methods with an AUC10% of 0.120 (see Table 3). The second best method LBtope reaches an AUC10% of 0.083. For a comparison on the evaluation dataset without length curation or sequence similarity reduction see Supplementary Figure S1.

3.4 Evaluation recap

The results, shown in Figure 4, indicate that EpiDope can predict even data that is relatively distinct from the training data and has a higher usability than competing methods.

The evaluation showed that previous methods are struggling to reliably predicting linear B-cell epitopes. Interestingly, for our data, simple property-based methods performed comparably well. These simple methods calculate the physical/chemical properties of protein sequences. These properties seem to be relatively robust indicators for epitopes. Methods that use machine learning are only as good as the data on which they are trained. This may explain why BepiPred2, which was trained on crystallographic data and thus trained on approximately 90% conformational epitopes, performs poorly in predicting linear epitopes.

3.5 EpiDope output and visualization

EpiDope produces multiple outputs. As an easily readable and interpretable format, EpiDope visualizes its results in an interactive html plot using the Python bokeh package (Bokeh Development Team, 2019). This allows large proteins to be displayed without impairing readability. For an example output plot see Figure 5. By default, EpiDope highlights regions of at least eight consecutive amino acids that have predicted values above the threshold of 0.818. This threshold corresponds to a 15% recall rate with a precision of 0.635 on the training dataset. Furthermore the user can provide amino acid sequences (text file with one sequence per line) that are either classified as epitopes or non-epitopes to highlight them in the html plot as blue or red regions, respectively.

Additionally, EpiDope produces simple computer parsable output. The file epidope_scores.csv lists the predicted score per

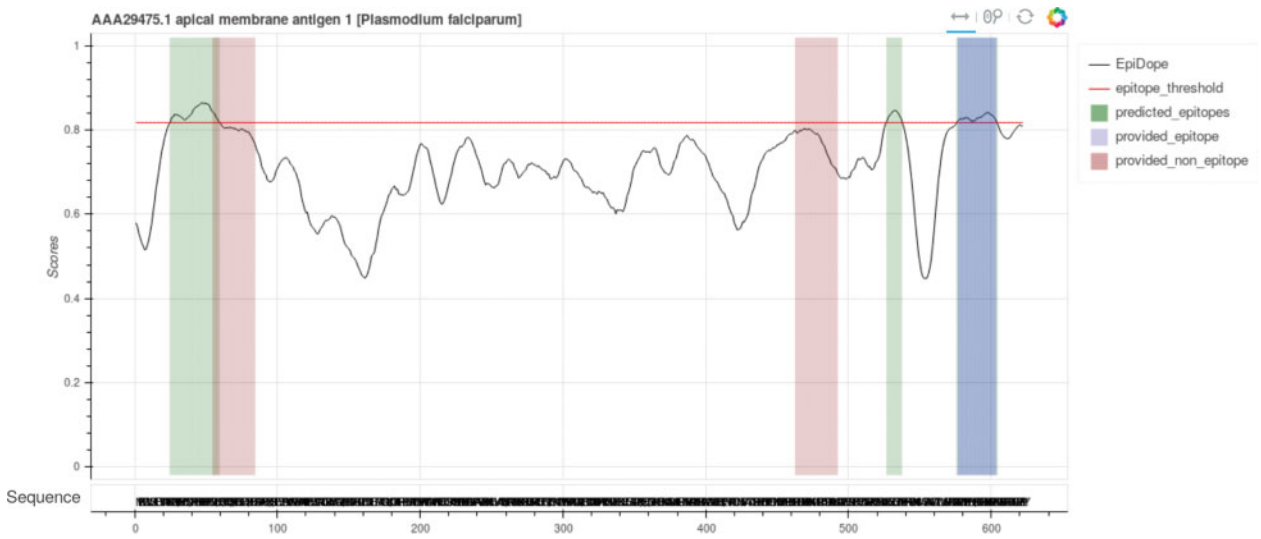

Fig. 5. Graphical output of EpiDope as an interactive html plot for the apical membrane antigen 1 of *Plasmodium falciparum*. The black line displays the predicted values of EpiDope per position (amino acid). A higher value means that EpiDope certifies this region a higher chance of being an epitope. The red line is the default threshold for the predicted epitopes. The green regions are the predicted parts that are above the threshold for at least eight consecutive amino acids. The blue and green regions are user provided regions that are known to be epitopes and non-epitopes, respectively

amino acid, predicted_epitopes.csv lists all regions with a score higher than the defined threshold and predicted_epitopes_sliced.faa is a multi-fasta file of potential epitopes with a user defined size and overlap, that can be used to scan for epitopes in wet lab experiments.

4 Conclusion

In this study, we have developed the linear B-cell epitope prediction tool EpiDope (Epitope Deep learning predictor).

While only requiring a protein's amino acid sequence, EpiDope has been shown to be the best-performing among currently available B-cell epitope prediction tools. EpiDope is based on a DNN. We trained EpiDope on almost 25 000 experimentally verified epitope and non-epitope regions. We have used two different datasets to compare the performance of EpiDope with numerous different tools, including the currently probably most used tool, BepiPred2. For the training dataset, we performed 10-fold cross-validation to ensure the reliability of the benchmarks. In addition, all proteins of a subset have a sequence identity of less than 50% to the proteins in the other nine subsets. This ensures that all 10 subsets are independent of each other. The second dataset (evaluation dataset) consists of 4769 new verified epitopes and non-epitopes, which therefore have not been present in the training dataset. On both datasets EpiDope outperformed all competing methods. Especially for the AUC10%, which represents the performance on the practically relevant top predictions. The AUC10% was corrected by multiplying with 10 [as described in BepiPred2 (Jespersen *et al.*, 2017)] and varies between 0.151 and 0.120.

The data from IEDB (Vita *et al.*, 2019) contains many short validated regions for which a high false-negative rate is expected (Rahman *et al.*, 2016). This affects the reliability of any *in silico* evaluation. This results in wrongly assigning true positive predictions as false positive predictions when comparing *in silico* predictions to *in vitro* tests. Therefore, we filtered the evaluation dataset for validated regions with length $\geq$ 12 amino acids, as suggested by Rahman *et al.* (2016).

The high predictive power of EpiDope enables a much more precise search for epitopes and, thus, faster and more cost-effective development of medical treatments or diagnostic methods. In future approaches, it could be interesting to investigate the use of binding distances of antigen-antibody crystal structures for the training of a conformational epitope predicting tool using the architecture presented in this work.

Funding

This work was funded in the framework of the national research network InfectControl, project "Molecular serology for rapid determination of vaccination titers (STIKO Serology)", which was financially supported by the Federal Ministry of Education and Research (BMBF) of Germany under grant 03ZZ0820A. This work was further supported by the Collaborative Research Center/Transregio 124 (FungiNet; number 210879364), project B5, funded by Deutsche Forschungsgemeinschaft (DFG). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Conflict of Interest: none declared.

Data Availability

The possibility to run EpiDope locally allows for high data security without the need to submit your sequences to external servers. It also enables the user to scan through entire proteomes without being dependant on server load and speed. For even easier usability for a broad research community, we plan to establish an EpiDope Online version. Until then EpiDope can be downloaded from GitHub (https://github.com/mcollatz/EpiDope) or installed via Conda or Docker. The training datasets are available in the open science framework (https://doi.org/10.17605/OSF.IO/KRW2J).

References

Arnold,B.F. *et al.* (2018) Integrated serologic surveillance of population immunity and disease transmission. *Emerg. Infect. Dis.*, **24**, 1188–1194.

Bengio,Y. (2012) Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, pp. 17–36.

Bokeh Development Team. (2019) *Bokeh: Python Library for Interactive Visualization.*

Branco,P. *et al* (2016) A Survey of Predictive Modeling on Imbalanced Domains. ACM Computing Surveys, **49**, 1–50. 10.1145/2907070

Breitwieser, Florian,P., and Salzberg,S. L. (2020) Pavian: Interactive analysis of metagenomics data for microbiome studies and pathogen identification. Bioinformatics, **36**, 1303.

Brown,C.D. and Davis,H.T. (2006) Receiver operating characteristics curves and related decision measures: a tutorial. *Chemom. Intell. Lab. Syst.*, **80**, 24–38.

Dosztányi,Z. *et al.* (2005) IUPred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content. *Bioinformatics*, **21**, 3433–3434.

Emini,E.A. *et al.* (1985) Induction of hepatitis a virus-neutralizing antibody by a virus-specific synthetic peptide. *J. Virol.*, **55**, 836–839.

Fawcett,T. (2006) An introduction to roc analysis. *Pattern Recogn. Lett.*, **27**, 861–874.

Heinzinger,M. *et al* (2019) Modeling aspects of the language of life through transfer-learning protein sequences. BMC Bioinformatics, **20**, 10.1186/s12859-019-3220-8.

Hochreiter,S. and Schmidhuber,J. (1997) Long short-term memory. *Neural Comput.*, **9**, 1735–1780.

Jespersen,M.C. *et al.* (2017) Bepipred-2.0: improving sequence-based B-cell epitope prediction using conformational epitopes. *Nucleic Acids Res.*, **45**, W24–W29.

Kolaskar,A. and Tongaonkar,P.C. (1990) A semi-empirical method for prediction of antigenic determinants on protein antigens. *FEBS Lett.*, **276**, 172–174.

Kringelum,J.V. *et al.* (2013) Structural analysis of B-cell epitopes in antibody: protein complexes. *Mol. Immunol.*, **53**, 24–34.

Li,W. and Godzik,A. (2006) Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, **22**, 1658–1659.

Manavalan,B. *et al.* (2018) IBCE-EL: a new ensemble learning framework for improved linear B-cell epitope prediction. *Front. Immunol.*, **9**, 1695.

Metcalf,C.J.E. *et al.* (2016) Use of serological surveys to generate key insights into the changing global landscape of infectious disease. *Lancet*, **388**, 728–730.

Parker,J. *et al.* (1986) New hydrophilicity scale derived from high-performance liquid chromatography peptide retention data: correlation of predicted surface residues with antigenicity and X-ray-derived accessible sites. *Biochemistry*, **25**, 5425–5432.

Pedregosa,F. *et al.* (2011) Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.*, **12**, 2825–2830.

Pellequer,J.-L. *et al.* (1993) Correlation between the location of antigenic sites and the prediction of turns in proteins. *Immunol. Lett.*, **36**, 83–99.

Rahman,K.S. *et al.* (2016) Inadequate reference datasets biased toward short non-epitopes confound B-cell epitope prediction. *J. Biol. Chem.*, **291**, 14585–14599.

Rappuoli,R. *et al.* (2014) Vaccines, new opportunities for a new society. *Proc. Natl. Acad. Sci. USA*, **111**, 12288–12293.

Sachse,K. *et al.* (2018) A novel synthetic peptide microarray assay detects chlamydia species-specific antibodies in animal and human sera. *Sci. Rep.*, **8**, 1–13.

Saha S., Raghava G.P.S.(2004) BcePred: Prediction of Continuous B-Cell Epitopes in Antigenic Sequences Using Physico-chemical Properties. In: Nicosia G., Cutello V., Bentley P.J., Timmis J. (eds) *Artificial Immune Systems. ICARIS* **2004**. Lecture Notes in Computer Science, vol **3239**. Springer, Berlin, Heidelberg.

Saha,S. and Raghava,G.P.S. (2006) Prediction of continuous B-cell epitopes in an antigen using recurrent neural network. *Proteins Struct. Funct. Bioinf.*, **65**, 40–48.

Sanchez-Trincado,J.L. *et al.* (2017) Fundamentals and methods for T- and B-cell epitope prediction. *J. Immunol. Res.*, **2017**, 1–14.

Singh,H. *et al.* (2013) Improved method for linear B-cell epitope prediction using antigen's primary sequence. *PLoS One*, **8**, e62216.

Sweredoski,M.J. and Baldi,P. (2009) COBEpro: a novel system for predicting continuous B-cell epitopes. *Protein Eng. Des. Select.*, **22**, 113–120.

Van Panhuis,W.G. *et al.* (2013) Contagious diseases in the united states from 1888 to the present. *N. Engl. J. Med.*, **369**, 2152–2158.

Van Regenmortel,M.H.V. (2009) What is a B-cell epitope? Methods in Molecular Biology (Clifton, N.J.), **524**, 3–20. 10.1007/978-1-59745-450-6_1 19377933.

Vita,R. *et al.* (2019) The immune epitope database (IEDB): 2018 update. *Nucleic Acids Res.*, **47**, D339–D343.

Yao,B. *et al.* (2012) SVMTriP: a method to predict antigenic epitopes using support vector machine to integrate tri-peptide similarity and propensity. *PLoS One*, **7**, e45152.

Zhang,J. *et al.* (2014) Conformational B-cell epitopes prediction from sequences using cost-sensitive ensemble classifiers and spatial clustering. *BioMed Res. Int.*, **2014**, 1–12.