

Deep Learning and Computational Modeling for Epitope-Focused Immunogen Design Targeting Monkeypox

Gurudeeban Selvaraj

gurudeeban.selvaraj@avmc.edu.in

Aarupadai Veedu Medical College & Hospital <https://orcid.org/0000-0002-7223-3853>

Satyavani Kaliamurthi

Medical and Bioresearch, Aarupadai Veedu Medical College & Hospital, Vinayaka Mission's Research Foundation (Deemed to be University), Kirumampakkam, Puducherry-607403, INDIA.

<https://orcid.org/0000-0002-3604-8810>

Gilles H Peslherbe

Centre for Research in Molecular Modeling, Department of Chemistry and Biochemistry, Concordia University, Montreal, Canada, H3G 1M8

Dongqing Wei

The State Key Laboratory of Microbial Metabolism, College of Life Sciences and Biotechnology, Shanghai Jiao Tong University, Shanghai 200240, China

Research Article

Keywords: monkeypox virus, mpox, epitope prediction, immunogen design, deep neural network, machine learning, amino acid encoding, reverse vaccinology

Posted Date: October 23rd, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11464247/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: The authors declare no competing interests.

Abstract

Monkeypox (MPX), now widely called mpox, is a zoonotic orthopoxvirus infection that has moved from a neglected tropical disease to a global public health concern. Live-attenuated smallpox vaccines offer cross-protection, but they are produced in limited quantities, are unevenly distributed, and are not suited to every recipient. These constraints motivate the search for rationally designed, epitope-focused immunogens that can be produced at scale and tailored to emerging viral variants. This article describes an *in silico* framework that combines amino acid feature engineering, classical machine learning (ML) benchmarking, and a deep neural network (DNN) to predict potential T-cell and B-cell epitopes from the MPX virus proteome (VP). Epitope and non-epitope sequences, together with antigen information, were collected from public immunology and virology databases, encoded numerically, and described by a panel of 84 amino acid features. Hierarchical heat-map clustering based on Pearson correlation identified 44 of the 84 features as redundant, and these were removed before modeling. Twenty-eight ML algorithms were then screened; the five best performers, namely Random Forest, Label Spreading, Decision Tree, Label Propagation and Extra Trees, were compared using accuracy, F1 score and the Matthews correlation coefficient (MCC), the last two being appropriate for imbalanced data. A DNN with fully connected dense layers, trained by back-propagation with the Adam optimizer, was evaluated with two and four hidden layers. The four-layer network converged faster in terms of loss, while accuracy did not differ significantly between architectures across training epochs. According to the held-out test data, the DNN outperformed the traditional ML models and other deep learning approaches. Because the framework is not specific to one pathogen, it can in principle be retrained to support vaccine design against other emerging infectious diseases. We discuss the biological rationale, methodological choices, limitations, and the experimental validation steps required to move from computational candidates to a multivalent MPX immunogen.

1. Introduction

Monkeypox virus (MPXV) is a large, double-stranded DNA virus of the genus *Orthopoxvirus*, which also includes variola virus (the agent of smallpox), vaccinia virus and cowpox virus. It was first identified in captive monkeys in 1958, and the first human case was recognized in 1970 in the Democratic Republic of the Congo. For decades, human disease was reported mainly in Central and West Africa, with a zoonotic reservoir thought to include rodents and other small mammals [1]. Surveillance data from the Congo Basin showed a marked rise in human cases about thirty years after routine smallpox vaccination ended, consistent with waning population immunity to orthopoxviruses [4]. Reviews of the changing epidemiology of the disease reached a similar conclusion: as cross-protective immunity fades in younger cohorts, the ecological niche for MPXV widens [3]. The situation changed substantially in 2022, when clusters of human infection were reported in dozens of countries with no previous endemic transmission. Clinical series from sixteen countries described a predominantly sexual-contact transmission pattern, with skin lesions, anogenital involvement and systemic symptoms [2]. Genomic analyses of the 2022 outbreak showed a lineage distinct from earlier West African isolates and signs of microevolution during human-to-human spread [7]. For public health authorities, the outbreak underlined

a practical reality: a virus that had been viewed as geographically contained can establish sustained transmission chains when conditions permit.

Smallpox vaccination has historically been the main preventive tool against MPXV. Animal studies demonstrated that antibodies induced by smallpox vaccination are necessary and sufficient for protection against monkeypox virus in non-human primates [8], and human studies found that previously vaccinated individuals retained protective immunity against monkeypox years later [9]. On this basis, live-attenuated smallpox vaccines have been approved for the prevention of monkeypox in North America, and prophylaxis is recommended for health workers and for patients after first exposure, so as to prevent the onset of disease. The supply picture is far less encouraging. Vaccine stocks are limited in Latin America, Europe and many developing countries, and the production of live-attenuated products requires specialized manufacturing and cold-chain infrastructure. Live vaccinia-based products also carry contraindications in immunocompromised people, in pregnancy and in individuals with certain skin conditions. These limitations are the central motivation for the work summarized here. A vaccine that is built from a defined set of immunogenic epitopes can, at least in principle, be manufactured by recombinant or synthetic routes, can avoid replicating virus, and can be updated rapidly when new sequence data appear.

The idea of using genome and proteome information to select vaccine antigens, usually called reverse vaccinology, transformed vaccine discovery by allowing candidates to be identified computationally before any laboratory work begins [10]. It has since been extended from whole proteins to individual epitopes, the short peptide regions that are physically recognized by T-cell receptors (when presented by major histocompatibility complex molecules) or by antibodies [11]. Epitope-focused designs can concentrate the immune response on conserved, functionally important regions and can exclude regions that provoke unhelpful responses. The success of nucleic-acid vaccine platforms has also made it far easier to deliver multi-epitope constructs, because the design is encoded in sequence rather than in a complex biological product [12]. The obstacle is prediction. Poxviruses encode roughly two hundred proteins, and the number of candidate peptides is large. Experimental epitope mapping is slow and expensive, so computational filters are needed that can rank candidates reliably. Traditional tools rely on position-specific scoring or shallow neural networks trained on binding affinity data, and their performance depends heavily on the quality of the training data and the representation of the amino acids.

To address the need for a promising MPX immunogen, we propose an in silico deep learning (DL) approach that predicts potential epitopes from the MPX virus proteome, with the longer-term objective of designing a multivalent immunogen. The contributions of the work are as follows. (1) A reproducible pipeline that links public epitope and antigen data to numerical feature generation, feature selection, ML benchmarking and DL modeling; (2) A heat-map clustering step that removes 44 of 84 correlated amino acid features, giving a more compact and less redundant input; (3) A systematic comparison of 28 ML algorithms and of DNN architectures with two and four hidden layers; (4) An evaluation protocol that uses accuracy, F1 score and MCC so that class imbalance does not hide poor performance.

2. Methodology

2.1 The MPXV genome and candidate antigens

The MPXV genome is approximately 197 kilobases long, with a conserved central region that encodes essential replication and structural functions and more variable terminal regions that encode host-interaction proteins [5]. The virus produces two infectious forms, the intracellular mature virion and the extracellular enveloped virion, and each displays different surface proteins. Antibodies against both forms contribute to protection, which is why a multivalent design that includes antigens from each form is attractive. Genomic comparison of human isolates from the Congo Basin showed measurable variation among strains [6], and the 2022 outbreak lineage added further changes [7]. Any immunogen should therefore favor epitopes that are conserved across circulating lineages.

2.2 Public resources for epitope and antigen data

The Immune Epitope Database (IEDB) curates experimentally validated T-cell and B-cell epitopes, along with assay information and host details [13]. The Virus Pathogen Database and Analysis Resource (ViPR) provides sequence, annotation and analysis tools for virology research [14]. The UniProt knowledgebase supplies reviewed protein sequences and functional annotation [16], and the Protein Data Bank (PDB) holds the experimentally determined three-dimensional structures that allow epitope positions to be mapped onto antigens [15]. The workflow in this study draws on these resources together with specialized predictors for peptide-MHC presentation, such as NetMHCpan [17, 18], and with the broader IEDB analysis tools [20].

2.3 Computational epitope prediction

Early T-cell epitope predictors modeled MHC binding with position-specific matrices. Neural-network methods then became standard: NetMHCpan, trained on very large binding-affinity and eluted-ligand datasets, predicts peptide presentation across many MHC alleles [17, 18]. For B-cell epitopes, sequence-based methods such as BepiPred-2.0 use random forest models to separate epitope from non-epitope residues [19]. More recently, convolutional architectures have been applied directly. For example, ranking-based convolutional neural networks have been proposed for peptide-MHC class I binding prediction and reported gains over earlier approaches [21]. These studies show that deep models can learn useful features from sequence, but they also show that performance depends strongly on how peptides are encoded and on how the data are balanced.

2.4 Amino acid encoding

A neural network cannot read letters; each residue must be converted to numbers. Common choices include one-hot encoding, in which each residue becomes a sparse binary vector, and substitution-matrix encodings such as BLOSUM, which represent residues by their evolutionary substitution patterns [23]. A systematic comparison of encoding schemes for deep learning applications found that the choice of representation influences accuracy and that no single scheme is universally best [22]. Beyond simple

encodings, residue-level physicochemical descriptors, such as hydrophobicity, charge, polarity and volume, can be calculated by software packages; the Pepfeature package, for instance, generates a large number of peptide features intended for prediction tasks [24]. The study described here follows this tradition by combining one-hot and BLOSUM encodings with a computed feature table, followed by normalization and feature selection.

2.5 Machine learning for imbalanced biomedical classification

Epitope datasets are almost always imbalanced, because confirmed non-epitopes are rarer or less consistently reported than positives, or the reverse, depending on the assay. Under imbalance, a classifier that predicts the majority class most of the time can show high accuracy while being useless. Resampling methods such as SMOTE [32] and cost-sensitive learning [33] are common remedies, but the more basic requirement is the use of metrics that reflect both classes. The Matthews correlation coefficient [34] summarizes the full confusion matrix and has been argued to be more informative than accuracy or F1 on imbalanced binary problems [35]. For this reason, this study reports accuracy, F1 score and MCC together.

3. Computational Methods

The overall workflow is summarized in the following subsections: data collection, sequence encoding and numerical feature generation, normalization and feature selection, ML model building and validation, DNN construction and training, and model assessment with a view to deployment for predicting epitopes in MPX proteomes.

3.1 Data collection

Training data consisted of antigen records and epitope or non-epitope peptides. Positive examples were T-cell and B-cell epitopes with experimental support, obtained from the IEDB [13] and from related epitope-prediction resources. Antigen sequences were retrieved from ViPR [14] and UniProt [16], and structural information was drawn from the PDB [15]. Peptide-presentation predictors from the NetMHC family [17, 18] were used to supplement and cross-check the T-cell data. Non-epitope peptides were taken as sequences from the same antigens that had been tested and reported as negative, or that were not associated with any positive assay result. The resulting dataset contained a different number of positive and negative elements, which is the reason for the choice of evaluation metrics described below.

3.2 Sequence encoding and feature generation

Peptide sequences were first represented as strings of amino acid letters. They were then transformed into numerical form using one-hot encoding and BLOSUM-based encoding [22, 23]. In parallel, a table of 84 amino acid features was calculated per sequence with a peptide-feature package [24]. These features

capture composition, physicochemical properties and positional characteristics that are plausibly related to immunogenicity, such as hydrophobicity, charge distribution and flexibility. Using several encodings allows both local sequence patterns and aggregate peptide properties to be available to the models.

3.3 Normalization and feature selection

All numerical features were normalized so that variables with large numeric ranges would not dominate the learning process. To produce an efficient and non-redundant feature set, a pairwise correlation matrix was calculated and displayed as a heat map with hierarchical clustering using Pearson correlation as the similarity measure, a visualization approach that has long been used for high-dimensional biological data [25]. Features that fell into tightly correlated blocks carried largely overlapping information. From the 84 calculated features, 44 were excluded because of high correlation with others, leaving 40 features for the training and test datasets. This step reduces dimensionality, lowers the risk of overfitting and shortens training time, and it also makes the final feature set easier to interpret.

3.4 Machine learning screening

To identify strong baselines, we used the LazyPredict package [26], which wraps scikit-learn [27] to fit a large panel of classifiers with default settings on the same train/test split. In total, 28 ML models were developed using different algorithms, spanning linear models, support vector machines, nearest-neighbor methods, naive Bayes variants, tree-based methods, ensembles and semi-supervised graph methods. Models were ranked by accuracy, F1 score and MCC. The top five were Random Forest [28], Label Spreading [31], Decision Tree, Label Propagation [30] and Extra Trees [29]. This screening step serves two purposes: it gives a fair baseline against which the deep model must be compared, and it indicates which families of algorithm suit this feature space.

3.5 Deep neural network architecture

The DNN was implemented as a feed-forward network. Deep learning in general learns hierarchical representations through multiple layers of nonlinear transformation [36, 37]. In this work, the input feature vector passes through a stack of hidden layers; the study compared networks with two and four hidden layers. Hidden units use nonlinear activations of the rectified linear type [42], and regularization options such as dropout [40] and batch normalization [41] are among the hyperparameters that can be tuned. The learned features are then fed into a fully connected dense output layer, which makes the final epitope versus non-epitope prediction. The hypothesis motivating the multi-layer design is that successive layers can mine the relationships between amino acid fragments of the viral proteome in terms of local connectivity and dependence, so that interactions between neighboring residues and between physicochemical properties are captured beyond what a single linear boundary could express.

3.6 Training procedure

All network parameters were trained in a back-propagation fashion [38] with the Adam optimizer, which adapts the learning rate for each parameter using estimates of the first and second moments of the gradients [39]. The loss was the logarithmic loss (log loss, or binary cross-entropy) and training was

monitored over 100 epochs on both training and validation sets, as shown by the accuracy and loss curves. Hyperparameters, including the number of hidden layers, the number of hidden units, the dropout rate and the activation function, were adjusted with reference to the validation behavior. The final model was evaluated on a test set that had not been used for training or tuning.

3.7 Evaluation metrics

Three complementary measures were used. Accuracy is the proportion of correct predictions. The F1 score is the harmonic mean of precision and recall and reflects performance on the positive class. The MCC, defined by the equation below, uses all four cells of the confusion matrix, where TP, TN, FP and FN denote true positives, true negatives, false positives and false negatives, respectively:

$$\text{MCC} = (\text{TP} \times \text{TN} - \text{FP} \times \text{FN}) / \sqrt{[(\text{TP} + \text{FP})(\text{TP} + \text{FN})(\text{TN} + \text{FP})(\text{TN} + \text{FN})]}$$

MCC ranges from -1 to $+1$, with 0 indicating performance no better than chance. Because the dataset has unequal numbers of positive and negative elements, F1 and MCC provide a more honest view of quality than accuracy alone [34, 35].

3.8 Software environment

The work was implemented in Python on a cloud notebook platform. Data handling relied on pandas [46] and NumPy [45], ML on scikit-learn [27] and LazyPredict [26], deep learning on TensorFlow [43] with the Keras interface [44], and plotting on Matplotlib [47] and seaborn [48]. These open-source libraries make the entire pipeline easy to reproduce and to share.

4. Results

4.1 Feature clustering and selection

Heat-map clustering of the 84 calculated amino acid features with hierarchical clustering and Pearson correlation exposed extensive redundancy. Diagonal blocks of strongly correlated features appeared throughout the matrix, which is expected for descriptors derived from related physicochemical scales. Removal of 44 of the 84 features retained 40 that were comparatively uncorrelated. In practical terms, this halved the input dimensionality, which speeds up both ML screening and DNN training and reduces the chance that the models simply memorize correlated noise. The clustering analysis also provided a useful sanity check: the retained features came from several distinct clusters, so different aspects of peptide chemistry remained represented.

4.2 Machine learning benchmarking

Of the 28 ML models, the five with the best metrics were Random Forest, Label Spreading, Decision Tree, Label Propagation and Extra Trees. Several observations arise from the comparison of their accuracy, F1 and MCC values. First, tree-based methods dominated. Random Forest and Extra Trees are ensembles of decision trees that average many high-variance learners, and a single Decision Tree also performed well.

Tree methods cope naturally with features on different scales and with nonlinear interactions, which suits a table of physicochemical descriptors. Second, two graph-based semi-supervised algorithms, Label Propagation and Label Spreading, also appeared in the top five. Both propagate labels through a similarity graph built from the feature space, which suggests that epitopes tend to cluster in feature space, so that a peptide's neighbors are informative about its class. Third, the MCC values were visibly lower than accuracy and F1 for all five models. This is the typical signature of an imbalanced dataset: the majority class inflates accuracy and F1, whereas MCC penalizes errors on the minority class. Taken together, accuracy, MCC and F1 confirm the quality of the dataset for classification tasks, while also showing why accuracy alone would be misleading.

4.3 Deep neural network performance

The DNN was trained for 100 epochs and monitored through accuracy and log-loss curves for training and validation data. Accuracy rose rapidly during the first epochs and then plateaued, while the loss fell steeply and then flattened. Training and validation curves remained close to each other, which indicates no marked overfitting under the conditions tested.

To verify convergence and accuracy for different depths, two additional experiments were carried out with two and four hidden layers. The four-layer model had faster convergence in terms of the loss value. Accuracy across epochs, by contrast, did not differ significantly between the two architectures. The practical consequence is that additional depth improved optimization speed without altering the eventual classification accuracy, and it is reasonable to prefer the four-layer network when training time and reliability of convergence matter. On the test dataset, the DNN model outperformed traditional ML models and other DL methods, with remarkable performance and reliability. This statement should be read carefully: the comparison is against the models built within the same study and against selected published approaches, using one dataset. A full independent benchmark would require standardized external test sets.

4.4 Summary of findings

- Hierarchical heat-map clustering reduced 84 features to 40 informative ones.
- Twenty-eight ML models were screened; tree-based ensembles and graph-based semi-supervised methods led the ranking.
- The DNN with four hidden layers converged faster than the two-layer version, with comparable accuracy.
- On the test data, the DNN surpassed the best classical ML models.

5. Discussion

The advantage of the DNN has several plausible sources. Epitope recognition depends on combinations of residue properties across several positions, and stacked nonlinear layers can represent such interactions compactly. Classical ensembles must approximate the same interactions with many axis-

aligned splits. Neural networks also benefit from continuous optimization of all parameters at once through back-propagation, whereas tree methods grow greedily. Finally, deep networks are flexible in handling multiple encodings, so that one-hot, BLOSUM-derived and physicochemical inputs can be merged without hand-crafted combination rules [36, 37]. Related work in peptide-MHC binding prediction has shown similar gains when convolutional layers replace simple scoring schemes [21]. That said, the margin by which deep models beat strong tree ensembles is often modest on tabular data, and gains should be confirmed with repeated cross-validation and external datasets. The finding that the semi-supervised graph methods rank among the best ML models is itself informative, because it indicates that the feature space is well structured and that further gains could come from combining supervised and graph-based approaches.

Correlated features inflate model variance and complicate interpretation. By discarding 44 of 84 features, the pipeline traded a small amount of potentially unique information for a large gain in efficiency. This is a defensible trade in a setting in which the number of positive peptides is limited. Nonetheless, correlation-based filtering is unsupervised with respect to the label: two correlated features may differ in their relevance to the epitope class, and a retained feature may be less useful than the one removed. Complementary approaches, such as permutation importance, recursive feature elimination or attention-based weighting, could be used in future work to examine whether the choice of retained features is optimal.

The observation that four hidden layers converge faster than two while giving similar accuracy deserves comment. Deeper networks can reach good solutions in fewer epochs when gradients flow well, which depends on the activation function and the optimizer. Adam's per-parameter learning rates are known to speed early training on many problems [39]. It is also possible that the dataset is simple enough that two layers already capture most of the learnable signal, so that depth mainly smooths optimization. The close agreement of training and validation curves suggests good generalization within the data distribution, but this does not guarantee generalization to peptides from unseen antigens. Group-aware splitting, in which all peptides from an antigen or a homology cluster are placed in the same fold, is the stricter standard and should be adopted in later versions.

The study uses metric choice rather than resampling as the main response to imbalance. Metric choice is necessary but not sufficient. Future iterations could add class-weighted loss functions, oversampling with SMOTE [32] or focal-loss variants, and they should report precision-recall curves, which are often more informative than receiver operating characteristic curves when positives are rare [33, 35]. Calibration of predicted probabilities also matters if scores will be used to rank peptides for synthesis. A predicted epitope is not yet a vaccine. Translating predictions into a multivalent MPX immunogen requires several additional steps. Conservation analysis: Candidate epitopes should be mapped against all available MPXV genomes, including the 2022 lineage [7] and earlier clades [6], and those conserved across lineages should be prioritized. Population coverage: T-cell epitopes must be presented by MHC alleles that are common in the target populations; predictors such as NetMHCpan can estimate this [17, 18]. Structural accessibility: For B-cell epitopes, surface exposure and conformational context should be

examined using solved structures [15] and, where structures are lacking, computed models from methods such as AlphaFold [49]. Safety screening: Candidates should be checked for similarity to human proteins to avoid autoimmunity, and for allergenicity and toxicity. Construct design: Selected epitopes can be joined with suitable linkers and adjuvant sequences, and the construct expressed through recombinant or mRNA platforms [12]. Experimental validation. Binding assays, T-cell activation assays and neutralization or animal challenge studies are the final arbiters. Earlier protection studies establish the benchmark, since antibody responses were shown to be a correlate of protection in primates [8, 9].

Relation to other approaches

Protein language models trained on hundreds of millions of sequences learn representations that capture structure and function without labels [50], and could replace hand-crafted features in future versions of this pipeline. Structure-aware predictors using computed protein structures [49] could improve B-cell epitope prediction, which depends heavily on three-dimensional shape. The framework described here should be viewed as a lightweight, transparent baseline, easily trained on modest hardware, onto which such components can be added. Nothing in the pipeline is specific to MPXV apart from the data. The same sequence of steps (retrieval of epitope data, encoding, correlation-based reduction, ML screening, DNN training and evaluation with imbalance-aware metrics) can be run for another pathogen. This is the basis of the claim that the model has the potential to help design vaccines against other emerging infectious diseases. The speed of the approach is important for outbreak response: it can generate a ranked list of candidate peptides within days of a new genome sequence becoming available, in contrast with the months or years required for conventional antigen discovery [10, 11].

6. Limitations

The work has limitations that should be stated plainly. First, it is entirely computational, and no wet-laboratory confirmation of predicted epitopes is reported. Second, the training data come from public databases that vary in assay type, host species and reporting quality, and non-epitope labels are particularly uncertain, because an untested peptide is not necessarily a true negative. Third, the data are imbalanced, and although appropriate metrics were used, the minority class remains small. Fourth, the comparison with other deep learning methods is limited to the datasets and settings of this study. Fifth, a correlation-based filter may remove features that are individually informative. Sixth, the evaluation used a single train/test partition; repeated cross-validation, group-wise splitting and independent external validation would give stronger evidence. Finally, an epitope-focused immunogen may need adjuvants, delivery systems and an appropriate balance between cellular and humoral immunity, none of which can be addressed by sequence prediction alone.

7. Future Directions

Future development of the model will focus on integrating protein language model embeddings [50] and structure-derived features [49] with physicochemical descriptors to improve epitope prediction accuracy. Model robustness and generalizability will be enhanced by replacing single-split testing with nested, group-aware cross-validation and independent external benchmark datasets. Explainable AI methods will be incorporated to identify the sequence features driving individual predictions, thereby facilitating interpretation by immunologists. Furthermore, the model will be extended to a unified multitask learning framework capable of predicting MHC allele-specific antigen presentation and conformational B-cell epitopes. Epitope prioritization will consider conservation across MPXV lineages and related orthopoxviruses to identify candidates with broad protective potential. Finally, collaboration with experimental laboratories will enable the validation of top-ranked peptides, with experimental results incorporated into an active-learning loop to iteratively refine model performance and accelerate the development of broadly protective immunogens.

8. Conclusion

This article has described a DNN framework for designing an immunogen from the MPX virus proteome. After encoding, heat-map clustering provided a reliable and compact set of amino acid features, which were used as input to the deep learning model. A network with four layers was able to mine the relationships between amino acid fragments of the viral proteome in terms of local connectivity and dependence, and the learned features were passed to a fully connected dense layer for prediction. All network parameters were trained by back-propagation with the Adam optimizer. According to the test dataset, the DNN outperformed traditional ML and other DL methods, with remarkable performance and reliability. The framework is not limited to MPX vaccine design; with appropriate data, it has the potential to help design vaccines against other emerging infectious diseases. Experimental validation of the predicted epitopes remains the essential next step before they can contribute to a multivalent immunogen.

Declarations

Acknowledgements

This work was carried out at the Centre for Research in Molecular Modeling (CERMM) and the Department of Chemistry and Biochemistry, Concordia University, Montreal, Quebec, Canada. Support from the Digital Research Alliance of Canada, Calcul Québec and the Natural Sciences and Engineering Research Council of Canada (NSERC) is acknowledged.

References

1. Alakunle E, Moens U, Nchinda G, Okeke MI (2020) Monkeypox virus in Nigeria: infection biology, epidemiology, and evolution. *Viruses* 12(11):1257

2. Thornhill JP, Barkati S, Walmsley S et al (2022) Monkeypox virus infection in humans across 16 countries — April–June 2022. *N Engl J Med* 387(8):679–691
3. Bunge EM, Hoet B, Chen L et al (2022) The changing epidemiology of human monkeypox—a potential threat? A systematic review. *PLoS Negl Trop Dis* 16(2):e0010141
4. Rimoin AW, Mulembakani PM, Johnston SC et al (2010) Major increase in human monkeypox incidence 30 years after smallpox vaccination campaigns cease in the Democratic Republic of Congo. *Proc Natl Acad Sci USA* 107(37):16262–16267
5. Shchelkunov SN, Totmenin AV, Safronov PF et al (2002) Analysis of the monkeypox virus genome. *Virology* 297(2):172–194
6. Kugelman JR, Johnston SC, Mulembakani PM et al (2014) Genomic variability of monkeypox virus among humans, Democratic Republic of the Congo. *Emerg Infect Dis* 20(2):232–239
7. Isidro J, Borges V, Pinto M et al (2022) Phylogenomic characterization and signs of microevolution in the 2022 multi-country outbreak of monkeypox virus. *Nat Med* 28:1569–1572
8. Edghill-Smith Y, Golding H, Manischewitz J et al (2005) Smallpox vaccine-induced antibodies are necessary and sufficient for protection against monkeypox virus. *Nat Med* 11(7):740–747
9. Hammarlund E, Lewis MW, Carter SV et al (2005) Multiple diagnostic techniques identify previously vaccinated individuals with protective immunity against monkeypox. *Nat Med* 11(9):1005–1011
10. Rappuoli R (2000) Reverse vaccinology. *Curr Opin Microbiol* 3(5):445–450
11. Sette A, Rappuoli R (2010) Reverse vaccinology: developing vaccines in the era of genomics. *Immunity* 33(4):530–541
12. Pardi N, Hogan MJ, Porter FW, Weissman D (2018) mRNA vaccines — a new era in vaccinology. *Nat Rev Drug Discov* 17(4):261–279
13. Vita R, Mahajan S, Overton JA et al (2019) The Immune Epitope Database (IEDB): 2018 update. *Nucleic Acids Res* 47(D1):D339–D343
14. Pickett BE, Sadat EL, Zhang Y et al (2012) ViPR: an open bioinformatics database and analysis resource for virology research. *Nucleic Acids Res* 40(D1):D593–D598
15. Berman HM, Westbrook J, Feng Z et al (2000) The Protein Data Bank. *Nucleic Acids Res* 28(1):235–242
16. The UniProt Consortium (2021) UniProt: the universal protein knowledgebase in 2021. *Nucleic Acids Res* 49(D1):D480–D489
17. Jurtz V, Paul S, Andreatta M, Marcatili P, Peters B, Nielsen M (2017) NetMHCpan-4.0: improved peptide–MHC class I interaction predictions integrating eluted ligand and peptide binding affinity data. *J Immunol* 199(9):3360–3368
18. Reynisson B, Alvarez B, Paul S, Peters B, Nielsen M (2020) NetMHCpan-4.1 and NetMHCIIpan-4.0: improved predictions of MHC antigen presentation by concurrent motif deconvolution and integration of MS MHC eluted ligand data. *Nucleic Acids Res* 48(W1):W449–W454

19. Jespersen MC, Peters B, Nielsen M, Marcatili P (2017) BepiPred-2.0: improving sequence-based B-cell epitope prediction using conformational epitopes. *Nucleic Acids Res* 45(W1):W24–W29
20. Kim Y, Ponomarenko J, Zhu Z et al (2012) Immune epitope database analysis resource (IEDB-AR). *Nucleic Acids Res* 40(W1):W525–W530
21. Chen Z, Min MR, Ning X (2021) Ranking-based convolutional neural network models for peptide-MHC class I binding prediction. *Front Mol Biosci* 8:634836
22. ElAbd H, Bromberg Y, Hoarfrost A, Lenz T, Franke A, Wendorff M (2020) Amino acid encoding for deep learning applications. *BMC Bioinformatics* 21:235
23. Henikoff S, Henikoff JG (1992) Amino acid substitution matrices from protein blocks. *Proc Natl Acad Sci USA* 89(22):10915–10919
24. Khan E (2021) Pepfeature: a package for calculating epitope/peptide features for prediction purposes. <https://pypi.org/project/Pepfeature/>
25. Eisen MB, Spellman PT, Brown PO, Botstein D (1998) Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA* 95(25):14863–14868
26. Pandala S (2022) LazyPredict: a comprehensive machine learning package. <https://github.com/shankarpandala/lazypredict> (accessed 12 January)
27. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: machine learning in Python. *J Mach Learn Res* 12:2825–2830
28. Breiman L (2001) Random forests. *Mach Learn* 45(1):5–32
29. Geurts P, Ernst D, Wehenkel L (2006) Extremely randomized trees. *Mach Learn* 63(1):3–42
30. Zhu X, Ghahramani Z (2002) Learning from labeled and unlabeled data with label propagation. Technical Report CMU-CALD-02-107. Carnegie Mellon University
31. Zhou D, Bousquet O, Lal TN, Weston J, Schölkopf B (2004) Learning with local and global consistency. In: *Advances in Neural Information Processing Systems 16*
32. Chawla NV, Bowyer KW, Hall LO, Kegelmeyer WP (2002) SMOTE: synthetic minority over-sampling technique. *J Artif Intell Res* 16:321–357
33. He H, Garcia EA (2009) Learning from imbalanced data. *IEEE Trans Knowl Data Eng* 21(9):1263–1284
34. Matthews BW (1975) Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochim Biophys Acta* 405(2):442–451
35. Chicco D, Jurman G (2020) The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics* 21:6
36. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. *Nature* 521(7553):436–444
37. Goodfellow I, Bengio Y, Courville A (2016) Deep Learning. MIT Press, Cambridge, MA
38. Rumelhart DE, Hinton GE, Williams RJ (1986) Learning representations by back-propagating errors. *Nature* 323:533–536
39. Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. *arXiv:1412.6980*

40. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. *J Mach Learn Res* 15:1929–1958
41. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: *Proceedings of the 32nd International Conference on Machine Learning*. :448–456
42. Nair V, Hinton GE (2010) Rectified linear units improve restricted Boltzmann machines. In: *Proceedings of the 27th International Conference on Machine Learning*. :807–814
43. Abadi M, Barham P, Chen J et al (2016) TensorFlow: a system for large-scale machine learning. In: *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation*. :265–283
44. Chollet F et al (2015) Keras. <https://keras.io>
45. Harris CR, Millman KJ, van der Walt SJ et al (2020) Array programming with NumPy. *Nature* 585:357–362
46. McKinney W (2010) Data structures for statistical computing in Python. In: *Proceedings of the 9th Python in Science Conference*. :56–61
47. Hunter JD (2007) Matplotlib: a 2D graphics environment. *Comput Sci Eng* 9(3):90–95
48. Waskom ML (2021) seaborn: statistical data visualization. *J Open Source Softw* 6(60):3021
49. Jumper J, Evans R, Pritzel A et al (2021) Highly accurate protein structure prediction with AlphaFold. *Nature* 596:583–589
50. Rives A, Meier J, Sercu T et al (2021) Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proc Natl Acad Sci USA* 118(15):e2016239118