University of Brasilia

Institute of Exact Sciences
Department of Computer Science

A Hierarchical Domain Adaptation Method in Neural Language Models

With Application to Taxonomy-Aware Linear B-cell Epitope Prediction

Lindeberg Pessoa Leite

A thesis submitted to the University of Brasília in partial fulfillment of the requirements for the degree of Doctor of Philosophy (PhD) in Computer Science

Supervisor
Prof. Dr. Teófilo Emidio de Campos

Co-Supervisor
Prof. Dr. Felipe Campelo França Pinto

Brasilia
2025

University of Brasilia

Institute of Exact Sciences
Department of Computer Science

A Hierarchical Domain Adaptation Method in Neural Language Models

With Application to Taxonomy-Aware Linear B-cell Epitope Prediction

Lindeberg Pessoa Leite

A thesis submitted to the University of Brasília in partial fulfillment of the requirements for the degree of Doctor of Philosophy (PhD) in Computer Science

Prof. Dr. Teófilo Emidio de Campos (Supervisor)
CIC/UnB

Prof. Dr. Felipe Campelo França Pinto (Co-Supervisor)
University of Bristol

Prof. Dr. Luis Paulo Faina Garcia
CIC/UnB

Prof. Dr. Cristiano Leite de Castro
UFMG

Dr. João Luís Reis Cunha
University of York

Prof. Dr. Rodrigo Bonifacio De Almeida
Computer Science Graduate Program Coordinator

Brasilia, July 16, 2025

Abstract

Domain adaptation aims to enable classifiers trained on a source domain to perform effectively on a target domain. Single domain adaptation methods are typically designed to transfer knowledge from a single source domain, where all observations are implicitly assumed to bear the same level of relationship to the target domain. However, in real scenarios, domain adaptation frequently involves source domains with internal, often hierarchical, structures. For instance, this occurs in phylogenetic branching in biological datasets, the evolutionary progression of languages, interconnected thematic structures in scientific literature, offensive language identification, and fact-checking. A common yet simplistic strategy is to merge these sources into a single domain. However, this strategy neglects the distinct relationships between individual sources and the target domain and also noisy data in multi-level source domain. Creating a unified source dataset for this heterogeneous collection can eliminate the informative characteristics of individual domains and may result in negative transfer effects. Although multi-source domain adaptation is a well-studied approach for this type of problem, less research has been conducted when the source domains have hierarchical relationships.

This thesis investigates the hierarchical relationships of source domains to enhance predictions at the target domain level. Specifically, the proposed method captures the hierarchical relationships and their relative importance across different levels, improving the adaptability of neural language models. By explicitly modeling these hierarchical dependencies, the method enhances the model's ability to generalize throughout diverse hierarchical levels, ensuring more accurate predictions at the target level. To demonstrate its efficiency, the method is applied to a case study on epitope prediction, a critical problem in immunoinformatics. Experimental results reveal significant performance gains, which outperforms three state-of-the-art methods in identifying linear B-cell epitopes (LBCE), as evaluated across eight different metrics.

Keywords: Hierarchical Domain Adaptation, Neural Language Models.

Resumo Expandido

A adaptação de domínio tem como objetivo permitir que classificadores treinados em um domínio de origem tenham um bom desempenho em um domínio-alvo. Tradicionalmente, métodos de adaptação consideram um domínio de origem único, no qual todas as observações assumem implicitamente o mesmo nível de relacionamento com o domínio-alvo. Entretanto, em cenários reais, a adaptação de domínio frequentemente envolve domínios de origem com estruturas internas, muitas vezes hierárquicas. Exemplos incluem ramificações filogenéticas em conjuntos de dados biológicos, evolução das línguas, estruturas temáticas interconectadas na literatura científica, identificação de linguagem ofensiva e verificação de fatos. Uma estratégia comum, porém simplista, é mesclar fontes heterogêneas em um único domínio. Essa abordagem negligencia as relações distintas entre fontes individuais e o domínio-alvo, além de introduzir ruídos que prejudicam a transferência. Criar um conjunto de dados unificado para uma coleção heterogênea pode eliminar informações críticas, resultando em efeitos negativos de transferência. Embora a adaptação de domínio com múltiplas fontes seja amplamente estudada, pouca pesquisa aborda explicitamente adaptações que consideram estruturas hierárquicas internas nos domínios de origem.

Esta tese investiga as relações hierárquicas entre os domínios de origem por meio de uma adaptação de domínio hierárquica, que captura explicitamente dependências hierárquicas para aprimorar a generalização e precisão das predições no nível do domínio-alvo. O método proposto emprega uma *Hierarchical Weighted Cross-Entropy Loss*, que ajusta dinamicamente a contribuição relativa dos diferentes níveis hierárquicos, e corrige desbalanceamentos entre classes. Essa estratégia permite uma transferência de conhecimento mais robusta e adaptável, especialmente adequada para cenários com poucos dados rotulados e estruturas hierárquicas de vários níveis.

Para contextualizar esta proposta, a tese revisa inicialmente a evolução dos modelos de linguagem: desde n-gramas e modelos ocultos de Markov (HMMs), passando por redes neurais recorrentes (RNNs, LSTMs e GRUs), até arquiteturas modernas baseadas em atenção, como Transformers. O surgimento de modelos pré-treinados, como BERT, GPT e RoBERTa, revolucionou o processamento de linguagem natural, enquanto sua adap-

tação para sequências biológicas resultou em modelos como ESM, ProtBERT, ProtT5, ProteinBERT e Ankh. Apesar dos avanços, persistem limitações relevantes, como o viés nos dados, especialmente em cenários com escassez de dados rotulados no domínio-alvo. Essas limitações reforçam a necessidade de métodos robustos de adaptação de domínio, particularmente em contextos com vários níveis hierárquicos.

Este trabalho também apresenta conceitos fundamentais relacionados à adaptação de domínio, destacando métodos tradicionais baseados em discrepância, métodos adversariais, métodos de reconstrução e abordagens de normalização. Cada técnica busca reduzir a diferença entre domínios com graus variados de robustez, estabilidade e aplicabilidade. Adicionalmente, são abordados os regimes de supervisão, incluindo adaptação supervisionada, semi-supervisionada e não supervisionada.

A tese também discute a adaptação de múltiplas fontes, destacando tanto benefícios quanto desafios dessa abordagem, como transferência negativa e maior custo computacional. Finalmente, é aprofundada a adaptação de domínio hierárquica, que aproveita explicitamente estruturas hierárquicas dos dados para realizar adaptações considerando o grau de importância de cada nível superior.

Para validar a proposta, foi realizado um estudo de caso focado na predição de epítopos de células B lineares (LBCE), uma tarefa crítica na imunoinformática devido à importância de epítopos em diagnósticos, vacinas e imunoterapias. Inicialmente, uma abordagem de adaptação de domínio de fonte única foi aplicada à tarefa de predição de epítopos, validando a capacidade de transferência filogenética. Em seguida, a solução foi generalizada por meio do método de adaptação de domínio hierárquica proposto, que ajusta dinamicamente a contribuição dos exemplos de treinamento com base na estrutura hierarquica dos dados.

Os resultados experimentais demonstraram ganhos de desempenho na tarefa de predição de epítopos lineares de células B. Na configuração de adaptação de domínio de fonte única (Single-Source Domain Adaptation), o método proposto **EpitopeTransfer** superou consistentemente três métodos estado da arte — *BepiPred 3.0*, *EpiDope* e *EpitopeVec* — além de duas baselines internas. A avaliação foi conduzida em um conjunto de 20 domínios-alvo, utilizando oito métricas distintas: AUC, F1-score, coeficiente de correlação de Matthews (MCC), acurácia balanceada (BACC), valor preditivo positivo (PPV), valor preditivo negativo (NPV), sensibilidade e especificidade.

O *EpitopeTransfer* obteve AUC média de 0,690 ± 0,029, F1-score de 0,592 ± 0,060 e MCC de 0,258 ± 0,052, demonstrando superioridade substancial em relação aos concorrentes. Além disso, atingiu sensibilidade de 0,697 ± 0,068 e especificidade de 0,549 ± 0,072, evidenciando sua capacidade de generalizar tanto para regiões epítopos quanto não epítopos.

Adicionalmente, ao aplicar a estratégia proposta de adaptação de domínio hierárquica (Hierarchical Domain Adaptation), observou-se desempenho consistentemente superior ao da baseline em 17 domínios-alvo distintos. O modelo generalizado alcançou AUC média de 0,698 ± 0,027, superando os 0,625 ± 0,033 da baseline. Também apresentou ganhos em F1-score (0,549 ± 0,053 vs. 0,454 ± 0,056) e MCC (0,249 ± 0,044 vs. 0,154 ± 0,039).

Palavras-chave: Adaptação de Domínio Hierárquica, Modelos de Linguagem Neural.

Contents

1 Introduction 1

1.1 Motivation . . . 1

1.2 Contributions . . . 2

1.3 Thesis Organization . . . 3

2 Language Models 4

2.1 Introduction . . . 4

2.2 The History of Language Models . . . 4

2.2.1 Statistical Language Models . . . 5

2.2.2 Neural Language Models . . . 9

2.2.3 Example of Self-Attention . . . 15

2.2.4 Pretrained Language Models . . . 17

2.2.5 Protein Language Models . . . 18

2.3 Conclusion . . . 20

3 Domain Adaptation 22

3.1 Introduction . . . 22

3.2 Techniques . . . 22

3.2.1 Discrepancy-Based Methods . . . 23

3.2.2 Adversarial Methods . . . 23

3.2.3 Reconstruction-Based Methods . . . 24

3.2.4 Normalization-Based Approaches . . . 25

3.2.5 Hybrid Methods . . . 25

3.3 Supervision Regimes in Domain Adaptation . . . 25

3.3.1 Supervised Domain Adaptation . . . 26

3.3.2 Semi-Supervised Domain Adaptation . . . 26

3.3.3 Unsupervised Domain Adaptation . . . 27

3.4 Multi-Source Domain Adaptation . . . 29

3.4.1 Problem Formulation . . . 29

3.4.2 Approaches for MSDA in NLP 30
3.5 Hierarchical Domain Adaptation 31
3.5.1 HDA as a special case of MSDA 31
3.5.2 Approaches to HDA 32
3.6 Conclusion 33

4 The Epitope Prediction Problem 34
4.1 Introduction 34
4.2 Immunological Context 34
4.2.1 Linear B-cell Epitopes 35
4.2.2 Importance for Diagnostics and Vaccines 35
4.3 Epitope prediction task 36
4.4 Biological Taxonomies 37
4.4.1 Leveraging Taxonomic Levels 38
4.5 Related Work on Epitope Prediction 38
4.5.1 Physicochemical properties-based methods 39
4.5.2 Machine learning-based methods 39
4.5.3 Deep Learning Methods 40
4.6 Discussion 41

5 Domain Adaptation with a Protein LLM 43
5.1 Estimated density 43
5.2 Single Domain Adaptation Modeling 49
5.3 Hierarchical Domain Adaptation Modeling 52
5.4 Conclusion 55

6 Results 56
6.1 Introduction 56
6.2 Experimental Setup 56
6.2.1 Data extraction and preparation 57
6.2.2 Selected Neural Network Architecture 58
6.2.3 Optimization Strategy 60
6.2.4 Evaluation Metrics 62
6.3 Single Domain Adaptation Modeling 62
6.3.1 Performance Results (ESM-1b) 62
6.3.2 Performance Results (ESM-2) 70
6.4 Hierarchical Domain Adaptation Modeling 71
6.4.1 Performance Results 71

6.5 SDA vs HDA 75
6.6 Conclusion 77

7 Discussion 78
7.1 Contributions 78
7.2 Limitations 79
7.3 Future Work 80

Appendices 82

References 213

A Hierarchical Domain Adaptation Method in Neural Language Models

With Application to Taxonomy-Aware Linear B-cell Epitope Prediction

Chapter 1

Introduction

1.1 Motivation

Transferring knowledge across domains is a central challenge in machine learning, particularly when the target domain offers limited labeled data [Ben-David et al., 2010, Pan and Yang, 2010]. Traditional domain adaptation methods often assume a single source domain. However, in many real-world scenarios, there are multiple source of domains, each contributing distinct and complementary information [Guo et al., 2020]. To address this setting, a technique known as multi-source domain adaptation (MSDA) leverages the diversity of multiple domains, enabling models to utilize a broader range of patterns and knowledge. By aggregating information from various sources, MSDA has shown significant improvements in tasks where a single source domain may fail to generalize to the target domain. Nevertheless, MSDA introduces additional challenges, such as managing diverse labeled sources and addressing the shift between source and target domains [Nguyen et al., 2021].

Although MSDA has demonstrated promise in various applications [Guo et al., 2018], a less investigated scenario emerges when the source domains follow a hierarchical organization. In such a hierarchy, higher-level domains encompass broader or more general information with abundant data, whereas lower-level domains contain more specific knowledge and fewer labeled instances. For example, in biological datasets arranged according to a phylogenetic tree, higher taxonomic ranks (e.g., *Phylum*, *Class*) aggregate substantial information about evolutionary relationships and tend to be data-rich. Conversely, lower taxonomic ranks (e.g., *Genus*, *Species*) capture more specialized features but typically have fewer labeled samples. This hierarchical structure reflects the evolutionary lineage of organisms, making it a candidate for knowledge transfer through hierarchical domain adaptation, where information from higher-level domains can be leveraged to improve learning in more specific, data-scarce lower-level domains.

This thesis is centered on *Hierarchical Domain Adaptation in Neural Language Models*, a specialized form of MSDA in which source domains are organized hierarchically. It addresses the key research question: **Given data that is hierarchically structured, how can knowledge be effectively transferred from higher levels, where data is abundant, to lower levels, where data is scarce?**

1.2 Contributions

This work advances the study of Hierarchical Domain Adaptation (HDA) in Neural Language Models (NLMs) by introducing a novel approach that captures and utilizes hierarchical relationships among domains. It specifically addresses the challenge of leveraging hierarchical relationships among domains to enhance knowledge transfer, particularly in scenarios where the target domain has limited labeled data.

The contributions of this work can be summarized as follows:

- **Hierarchical Domain Adaptation Method:** A method for Hierarchical Domain Adaptation in Neural Language Models is introduced. This method enables smoother adaptation across hierarchical levels, mitigating negative transfer effects and enhancing predictive performance in data-scarce lower levels. As part of this approach, a *Hierarchical Weighted Cross-Entropy Loss* is proposed, incorporating *hierarchical weights* to dynamically adjust the contribution of higher-level data. Additionally, the loss function applies weighting strategies to balance the exposure to positive and negative samples throughout the hierarchy, mitigating potential biases that could impact model generalization.

- **Application to Epitope Prediction:** The proposed method is applied to an epitope prediction task and outperforms three state-of-the-art baselines across eight evaluation metrics. Identifying epitopes is a crucial step in a broad range of medical and immunological applications, including vaccines [Hamley, 2022], therapeutic antibodies [Sun et al., 2024], and immunodiagnostics [Mucci et al., 2017].

The proposed method can be applied to a wide range of tasks, including but not limited to, *Phylogenetic Branching in Biological Datasets* [Campelo et al., 2024], *The Evolutionary Progression of Languages* [Gray and Atkinson, 2003], *Interconnected Thematic Structures in Scientific Literature* [Tang et al., 2008], *Offensive Language Identification* [Rosenthal et al., 2021], and *Fact Checking* [Thorne et al., 2018].

While the method is inherently designed for hierarchical datasets, it also accommodates cases where data lacks an explicit hierarchy but can be structured accordingly. This flexibility enables its application to tasks where hierarchical relationships are not initially present but can be leveraged to enhance learning.

1.3 Thesis Organization

The remainder of this thesis is structured as follows: Chapter 2 outlines the development of language models, from early statistical approaches to modern Transformers. Chapter 3 describes domain adaptation techniques, from traditional methods to hierarchical approaches. Chapter 4 provides the background for the chosen case study on B-cell epitope prediction, detailing the biological, computational, and taxonomic aspects relevant to the application of the proposed method. Chapter 5 presents a detailed description of the proposed method, providing an explanation of the approach developed for this research. Chapter 6 shows the results obtained by applying the proposed method in two distinct scenarios: single domain adaptation and hierarchical domain adaptation. Finally, Chapter 7 discusses the findings in relation to the research questions, limitations, and proposes directions for future work.

Chapter 2

Language Models

2.1 Introduction

This chapter explores the evolution of language models, from early n-gram models to modern transformer-based architectures. It examines the transition from statistical methods to neural network-based approaches, including RNNs, LSTMs, and GRUs, which improved sequential text processing. A significant breakthrough came with the introduction of self-attention mechanisms, laying the foundation for transformer models such as BERT, which are now among the most widely used in natural language processing.

2.2 The History of Language Models

Human languages are composed of discrete symbols such as words and characters [Manning and Schütze, 1999]. These symbols must be converted into continuous numerical representations for computers to process and understand their meanings. This conversion is essential because computers operate on numerical data. Without this transformation, the semantic richness of language cannot be captured [Bengio et al., 2000, Mikolov et al., 2013]. One of the primary challenges in this transformation process is the complexity and variability of human language. Language is infinitely flexible and constantly evolving, capable of generating an endless number of unique sentences, each with its own nuances and context. This makes it impossible for a computer to exhaustively calculate or store all possible linguistic combinations [Jurafsky and Martin, 2025, Manning and Schütze, 1999]. To address this challenge, a wide range of language models have been developed, evolving from early statistical methods to advanced neural approaches.

2.2.1 Statistical Language Models

Early Language Models: N-Grams

Statistical language modeling aims to learn the joint or conditional probability distribution over sequences of words [Bengio et al., 2000]. In traditional statistical language models, such as the n-gram model, the probability of a word is estimated based on its frequency following a specific context in the corpus. Thus, the probability of a sentence is decomposed into the product of conditional probabilities of each word given its preceding words [Jurafsky and Martin, 2025].

The concept of n-grams has its roots in Claude Shannon's work in 1948, where he explored the statistical structure of language through the joint probability of consecutive symbols in communication systems [Shannon, 1948]. While n-grams were not explicitly formalized in Shannon's work, his exploration laid the groundwork for probabilistic approaches to language modeling. The application of n-gram language models gained significant attention in the 1970s, particularly through the work of Frederick Jelinek and his team, who applied statistical models to speech recognition [Jelinek, 1976]. In the subsequent decades, numerous advancements were made to refine and expand n-gram modeling techniques, enabling their application across diverse domains, including natural language processing, machine translation, and information retrieval [Goodman, 2001].

The **Unigram model** is one of the simplest statistical language models, where the occurrence of each word in a sentence is considered independent of other words. This model calculates the probability of a sentence by multiplying the individual probabilities of each word, ignoring any contextual information. For example, consider the sentence "Brazil is beautiful". The Unigram model would estimate the probability of this sentence as:

$$P(\text{Brazil is beautiful}) = P(\text{Brazil}) \times P(\text{is}) \times P(\text{beautiful})$$

Here, the probability of each word is determined by its frequency in a large corpus. While this model is computationally simple and efficient, it fails to account for the relationships between words, which are crucial for understanding the structure of natural language.

The **N-Gram model** builds on the Unigram model by considering the relationship between words. Specifically, it assumes that the probability of a word depends on the $N-1$ words that precede it, a principle known as the Markov assumption [Jurafsky and Martin, 2025]. For example, in a **bigram model** ($N = 2$), the probability of the entire sentence "Brazil is beautiful" would be estimated as:

$$P(\text{Brazil is beautiful}) \approx P(\text{Brazil}) \times P(\text{is} \mid \text{Brazil}) \times P(\text{beautiful} \mid \text{is})$$

This model is more effective than the Unigram model at capturing the context and structure of language, making it a more powerful tool for natural language processing tasks. Thus, the generalization of the joint probability for a sequence of $n+1$ words can be expressed as the product of all the bigram conditional probabilities:

$$P(w_0 : w_n) \approx P(w_0) \cdot P(w_1 \mid w_0) \cdot \ldots \cdot P(w_n \mid w_{n-1}) = \prod_{i=1}^{n} P(w_i \mid w_{i-1})$$

However, a challenge arises when the model encounters words that it has not seen before, especially in new domains. This issue, known as the **Out of Vocabulary (OOV) problem**, requires a strategy to ensure that the model can still perform effectively [Jurafsky and Martin, 2025]. To address this, an open vocabulary approach is typically used, where a placeholder token, represented as `<UNK>`, is introduced to replace any unknown words. This approach helps the model manage words it has not encountered during training. Specifically, low-frequency words in the training corpus are replaced with the `<UNK>` token, based on the assumption that words that are rare in the training data are more likely to be unknown in future contexts. By incorporating `<UNK>` into the n-grams, the model can estimate probabilities for sequences that include unknown words, allowing it to better generalize to new domains where it may encounter words outside of its original vocabulary [Chen and Goodman, 1999].

Limitations. There are several limitations associated with building more powerful language models. First, creating a robust model requires a significantly larger training corpus to effectively model a wide range of n-grams. However, this often leads to the issue of the curse of dimensionality, where the model becomes increasingly complex and difficult to manage [Bengio et al., 2000]. Second, adapting these models to new domains can be challenging. Since the model relies heavily on the statistical patterns learned from the training data, it may struggle in domains where large training corpora are not available. In such cases, the model's performance tends to degrade [Bellegarda, 2004]. Additionally, simply increasing the size of the training corpus does not always result in better performance. Real-world test corpora frequently contain words and phrases that were not included in the original vocabulary, making it difficult for n-gram models to generalize across diverse test domains [Brants et al., 2007].

Hidden Markov Model

The Hidden Markov Model (HMM) is a probabilistic framework commonly used to model sequential data. It consists of two main components: a hidden Markov chain that represents an underlying sequence of states and a set of observation probability distributions associated with each state. At each step in the sequence, the model is assumed to be in one of these hidden states, which cannot be observed directly. Instead, an observable event or output is generated according to the probability distribution linked to the current hidden state. If the set of possible observations is finite, the HMM is referred to as discrete; if the observations can take any value, such as when generated by continuous probability distributions, it is called continuous [Rabiner, 1989].

The HMM's ability to model complex real-world phenomena through a sequence of hidden states and observable events has been well-established both theoretically and empirically [Rabiner, 1989]. When equipped with an adequate number of states and sufficient data, HMMs can capture the underlying probability distributions of complex processes, leading to simple yet powerful models. This strength is reflected in their widespread adoption as the foundation for developing automated speech recognition (ASR) systems, which are deployed in various practical applications [Rabiner, 1989]. Beyond speech recognition, HMMs have also proven their versatility in other fields, such as bioinformatics [Eddy, 1998], computer vision [Caelli and McCane, 2003], and other areas within natural language processing [Gao and Zhu, 2013].

Modeling Sequential Data

In the context of Natural Language Processing (NLP), HMMs play a important role in various tasks. They are applied to word prediction, part-of-speech tagging, and have been a foundational approach in the development of automated speech recognition systems [Rabiner, 1989]. These models are particularly effective in handling sequential data, where the underlying sequence of hidden states (such as parts of speech) can be inferred to predict the next word or phrase based on the previous context, making them powerful tools for various NLP applications [Jurafsky and Martin, 2025]. For instance, consider the sentence: *"Brazil won the match."* The goal is to determine the most probable sequence of parts of speech (POS) [1] for this sentence using HMM. A possible sequence of states could be:

- **Brazil** → Noun (NOUN)
- **won** → Verb (VERB)
- **the** → Article (ART)
- **match** → Noun (NOUN)

[1] Parts of speech (POS) are categories of words based on their grammatical roles in a sentence, such as nouns, verbs, adjectives, and adverbs. These categories help in syntactic and semantic analysis of language. For more information, see [Jurafsky and Martin, 2025].

In this context, the HMM is defined by two key components:

1. **Transition Probability** (A): The probability of transitioning from one state (part of speech) to another. For example, the probability that a Noun (NOUN) is followed by a Verb (VERB) can be expressed as:

   $$P(\text{Current State} \mid \text{Previous State})$$

   or more formally:

   $$A_{ij} = P(s_t = j \mid s_{t-1} = i)$$

   where:

   - s_t: Represents the state (part of speech) at time t. For example, this could be a NOUN, VERB, or other part of speech for the current word.
   - i: Represents the state (part of speech) at time $t-1$, i.e., the previous state.
   - j: Represents the state (part of speech) at time t, i.e., the current state.

2. **Emission Probability** (B): The probability of a specific word being generated given a state (part of speech). For example, the probability that the word "Brazil" is emitted given that the state is Noun (NOUN):

   $$B_j(o_t) = P(o_t \mid s_t = j)$$

   where o_t represents the observed word at time t.

Given a sequence of observed words $O = \{o_1, o_2, \ldots, o_T\}$ (e.g., "Brazil won the match"), the goal of the HMM is to find the most likely sequence of states $S = \{s_1, s_2, \ldots, s_T\}$ (e.g., NOUN, VERB, ART, NOUN) that explains the observed sequence. By modeling the relationship between words (observations) and their corresponding parts of speech (states), HMMs can predict the POS tags for new sentences, helping machines better understand human language [Manning and Schütze, 1999].

Optimization Techniques

In HMMs, the Baum-Welch (BW) algorithm is commonly used to estimate model parameters, particularly when dealing with incomplete or missing data [Rabiner, 1989]. The BW algorithm is a specific application of the expectation-maximization (EM) technique, which iteratively refines parameter estimates by alternating between two steps: the E-step, where the expected likelihood is calculated by treating hidden variables as if they were observed, and the M-step, where model parameters are updated to maximize this likelihood. This process continues until convergence to a stationary point of the likelihood function [Baum et al., 1970]. Originally designed for single sequences of discrete observations [Petrie, 1969], the BW algorithm has since been extended to handle multiple sequences and continuous observations [Rabiner, 1989].

To better illustrate this, consider again the task of predicting parts of speech (POS) in the sentence “Brazil won the match.” In this scenario, the observed sequence consists of the words in the sentence, while the hidden states correspond to the POS tags (NOUN, VERB, etc.). The Baum-Welch algorithm would be used to estimate the transition probabilities (such as the likelihood of a NOUN being followed by a VERB) and the emission probabilities (such as the likelihood that “Brazil” is a NOUN).

After estimating the model parameters using the Baum-Welch algorithm, the Viterbi algorithm [Viterbi, 1967] is applied to perform inference and identify the most likely sequence of hidden states (e.g., POS tags) for a given sequence of observations. The Viterbi algorithm, an efficient dynamic programming technique, calculates the optimal path through the states, ensuring that the sequence of POS tags best fits the observed words [Viterbi, 1967]. For instance, given the words “Brazil won the match”, the Viterbi algorithm would determine that the most likely sequence of states is NOUN, VERB, ART, NOUN, based on the estimated model parameters.

Limitations. Despite its usefulness, the HMM has several limitations that can impact its performance. First, the commonly used HMM formulation is based on the first-order Markov assumption, which states that the probability of being in a particular state at time t depends only on the state at time $t-1$. However, in textual data, dependencies often extend over multiple states, meaning this assumption does not always hold true [Rabiner, 1989]. Second, HMM assumes that observations are conditionally independent given the hidden states. In practice, this is rarely the case in real-world textual data, where observations are often correlated, leading to potential inaccuracies in the model’s predictions [Jurafsky and Martin, 2025]. Lastly, selecting the model’s topology is often done through trial and error. Although some general guidelines exist, there is no formal method to determine the optimal architecture for a given task. Additionally, deciding on the appropriate number of states and transitions for a model remains a significant challenge [Dimri et al., 2024].

2.2.2 Neural Language Models

Traditional language modeling techniques, such as n-gram models and HMMs, have been pivotal in the early development of NLP. While these models have achieved considerable success, they face certain challenges, such as difficulties in capturing long-range dependencies and handling rare word sequences [Chen and Goodman, 1999, Jurafsky and Martin, 2025].

The early 2000s marked a turning point in overcoming these limitations with the advent of neural language models, pioneered by [Bengio et al., 2000]. Unlike n-gram models that estimate the probability of a word w_i based on its preceding $n-1$ words, this new approach leveraged neural networks to represent each word as a continuous d-dimensional vector $z_i \in \mathbb{R}^d$. This shift was inspired by the concept of *distributed representations*, introduced by [Rumelhart et al., 1986], which suggests that concepts are represented by patterns of activation across multiple units rather than by single symbols. This idea later influenced the development of word embeddings. Although computationally intensive at the time, this approach demonstrated superior generalization for rare words and long-range patterns compared to n-grams. Practical adoption only became widespread later with implementations such as Word2Vec [Mikolov et al., 2013] and architectural advances in recurrent and transformer networks.

Recurrent Neural Networks

Early neural language models, such as the one proposed by [Bengio et al., 2000], estimate the probability of a word w_i given its preceding n-gram context using a feed-forward neural network f parameterized by θ. This is achieved by mapping the high-dimensional vectors of the preceding words to a continuous hidden representation. The network computes a hidden vector $h = f_\theta(z_{i-n+1}, \ldots, z_{i-1})$, where $h \in \mathbb{R}^d$. This hidden vector is then projected onto a score vector $s \in \mathbb{R}^{|V|}$ over the vocabulary. The probability of each word is obtained using the softmax function:

$$\hat{P}(w_i = v_i \mid w_{i-n+1}, \ldots, w_{i-1}; \theta) = \frac{e^{s_i}}{\sum_j e^{s_j}}. \tag{2.1}$$

This formulation converts the scores s_i into probabilities, where e^{s_i} reflects the models confidence in word v_i, and the denominator ensures normalization over the entire vocabulary.

Although effective, these models are limited by their fixed-size context window, which constrains their ability to capture dependencies across variable-length sequences. To address this limitation, Recurrent Neural Networks (RNNs) emerged as a promising architecture due to their ability to capture and model sequential data [Elman, 1990]. Unlike feed-forward neural networks[2], RNNs are specifically designed to process sequences by

[2] Feed-forward neural networks are a class of neural networks where information flows in one direction, from input to output, without loops or cycles.

maintaining a hidden state that evolves over time, encoding information from previous steps.

For example, consider the sentence “I live in Brazil”. As the RNN processes each word sequentially, it updates its hidden state to reflect the accumulated context. By the time it reaches Brazil, the hidden state contains information about the preceding words “I live in”, which helps the model make predictions about subsequent words or understand the context in which “Brazil” appears.

While this probabilistic formulation allows the model to generate coherent sequences, RNNs also face significant challenges, particularly with long-term dependencies. As the length of the sequence increases, RNNs struggle to retain important information from earlier in the sequence due to issues such as the vanishing gradient problem [Bengio et al., 1994]. This problem occurs when gradients used in backpropagation become exceedingly small, effectively preventing the network from learning long-range dependencies.

To address the limitations of traditional RNNs in learning long-term dependencies, the Long Short-Term Memory (LSTM) network was introduced by [Hochreiter and Schmidhuber, 1997]. LSTMs enhance the recurrent architecture by incorporating a memory cell capable of maintaining information over extended time intervals. This memory cell is regulated by three gates - input, forget, and output gates - which control the flow of information into, within, and out of the cell.

- **Input Gate**: Regulates how much of the new input is added to the cell state.
- **Forget Gate**: Determines how much of the previous cell state is discarded.
- **Output Gate**: Controls how much of the cell state is exposed to the hidden state and passed to the next time step.

This gating mechanism enables LSTMs to retain relevant information across longer sequences. For instance, in the sentence “I live in Brazil”, the model can remember that Brazil refers to a key entity even when predicting words that occur much later in the sequence. The hidden state h_t in an LSTM is computed based on the current input x_t, the previous hidden state h_{t-1}, and the internal cell state c_t. It is typically defined as:

$$h_t = o_t \odot \tanh(c_t), \tag{2.2}$$

where o_t is the output gate, c_t is the updated cell state, and $\odot$ denotes element-wise multiplication [Hochreiter and Schmidhuber, 1997].

Following the development of LSTM, Gated Recurrent Units (GRUs) were introduced as a simpler alternative to LSTMs [Cho et al., 2014]. GRUs simplify the LSTM architecture by combining the forget and input gates into a single update gate and by unifying the cell state and hidden state into a single vector. They often achieve performance comparable to LSTMs in many tasks [Chung et al., 2014]

- **Update Gate**: Controls how much of the previous state is retained and how much of the new input is incorporated.
- **Reset Gate**: Controls how much of the past information should be forgotten.

This architectural simplification results in a reduced number of parameters, making GRUs more computationally efficient than LSTMs. Despite their simpler structure, GRUs remain effective at capturing long-term dependencies, which makes them particularly suitable for scenarios with limited computational resources. The hidden state h_t in a GRU is typically computed as:

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t, \tag{2.3}$$

where $\tilde{h}_t$ is the candidate hidden state.

Although LSTM and GRU architectures have advanced the capabilities of RNNs, they still have some important limitations:

- **Training time**: The added complexity of LSTMs can substantially increase training duration, primarily due to the additional parameters involved. Although GRUs typically train faster than LSTMs, both demand more training time than basic RNNs [Chung et al., 2014].
- **Memory consumption**: The gating operations in LSTMs lead to a larger parameter space, thus consuming more memory than simple RNNs. GRUs, despite merging certain gates, still exhibit higher memory usage compared to standard RNN architectures [Chung et al., 2014].
- **Difficulty in capturing very long-term dependencies**: Although LSTMs and GRUs mitigate issues related to long sequences better than basic RNNs, they can still struggle with extremely long-term dependencies due to challenges such as vanishing and exploding gradients [Bengio et al., 1994].

Convolutional Neural Networks

Convolutional neural networks (CNNs) were originally developed for image processing tasks [Lecun et al., 1998], which excel at capturing spatial hierarchies within visual data. Their remarkable success in image recognition has inspired their adaptation for NLP tasks. In NLP, CNNs operate on text by applying convolutional filters to sequences of word embeddings, enabling the model to effectively identify local patterns and contextual relationships in textual data.

CNNs have been applied to various NLP tasks, including semantic parsing, search query retrieval, and sentence modeling [Collobert et al., 2011]. By using convolutional filters to sequences of word vectors, CNNs can capture local patterns in text data, making them well-suited for identifying features that are crucial for these tasks. Their ability to focus on n-grams of varying lengths allows them to detect important phrases and word combinations, contributing to their success in NLP. However, CNNs face certain limitations when applied to text. Unlike images, which have consistent spatial structures, text data often lacks such regularity, making it difficult for CNNs to capture long-range dependencies between words [Kim, 2014]. Moreover, CNNs may struggle with understanding the sequential nature of text, which is essential for tasks that require context from distant words [Tang et al., 2018]. These challenges highlight the need for careful adaptation when using CNNs in the less structured domain of language.

Attention

Attention mechanisms were introduced into RNNs [Bahdanau et al., 2014] and were later applied to CNNs in various NLP tasks [Yin et al., 2016, Zhao and Wu, 2016] to overcome some of the limitations these models face, particularly in NLP tasks. While CNNs excel at capturing local patterns within text data, they often struggle with modeling long-range dependencies due to the sequential nature of language [Tang et al., 2018]. Similarly, RNNs, despite being designed to handle sequences, can have difficulty retaining information over long sequences, leading to issues with context preservation [Bengio et al., 1994]. Attention mechanisms mitigate these challenges by enabling the network to selectively focus on the most relevant portions of the input during processing, improving performance on tasks that require understanding of complex dependencies [Vaswani et al., 2017].

In CNNs, the integration of attention mechanisms has led to the development of attention-based architectures. These architectures enhance feature extraction by enabling the model to focus on critical regions of the input data, which can improve generalization and performance in NLP tasks [Yin et al., 2016]. In RNNs, attention helps capture long-distance dependencies by dynamically weighting different parts of the input sequence during processing. This flexibility allows the model to retrieve relevant contextual information, improving performance in tasks such as machine translation. [Bahdanau et al., 2014, Luong et al., 2015].

Despite these advantages, attention mechanisms in CNNs and RNNs still have limitations. In CNNs, the added computation required to calculate attention weights can increase training time and memory usage, especially in large-scale applications [Xu et al., 2025]. In RNNs, while attention mitigates some long-range dependency issues, it does not completely resolve the vanishing gradient problem [Bengio et al., 1994].

Overall, the incorporation of attention mechanisms has improved the effectiveness of CNNs and RNNs in natural language processing. However, further research is required to address the remaining challenges associated with attention mechanisms.

Transformer

The Transformer architecture replaces recurrent and convolutional layers entirely with self-attention mechanisms. Unlike RNNs, which process inputs sequentially, or CNNs, which focus on local patterns, the Transformer uses multi-head self-attention to globally model relationships between all sequence elements. This enables parallel processing of sequences and eliminates positional biases, addressing long-range dependency limitations [Vaswani et al., 2017].

The Transformer architecture consists of several key components that work together to process and understand input data, primarily through the use of multi-head self-attention and feed-forward neural networks. The architecture is built around the following components [Vaswani et al., 2017]:

- **Input embedding and positional encoding:** The input tokens (words or subwords) are first converted into fixed-size continuous vector representations, called embeddings. Since the Transformer does not inherently consider the order of words (unlike RNNs), positional encodings are added to these embeddings to inject information about the relative positions of words in a sequence. This allows the model to understand the order in which words appear.
- **Multi-head self-attention mechanism:** The self-attention mechanism is the core innovation of the Transformer. It allows the model to focus on different parts of the input sequence when encoding a particular word. The "multi-head" aspect means that the model runs multiple attention mechanisms in parallel, each focusing on different parts of the sequence. The results are then concatenated and linearly transformed to create the final output for each position. This process allows the model to capture different types of relationships between words simultaneously.
- **Scaled dot-product attention:** This is the mathematical mechanism behind the self-attention operation and is applied separately in each attention head. For each token in the input sequence, three vectors are generated through learned linear transformations: Query (Q), Key (K), and Value (V). Attention scores are computed by taking the dot product between the Query and Key vectors, scaled by

$\sqrt{d_k}$, and passed through a softmax function to produce attention weights. These weights indicate how much focus each token should give to the others. Finally, each tokens new representation is computed as a weighted sum of all Value vectors, where the weights - determined by the attention mechanism - assign greater importance to the most relevant tokens.

- **Feed-forward neural networks (FFN):** Following the self-attention mechanism, each output passes through a position-wise feed-forward neural network. This network comprises two linear transformations separated by a ReLU activation function, enabling the model to introduce non-linearity and enhance the learned representations.
- **Residual connections and layer normalization:** To help with training deep networks, the Transformer uses residual connections around each sub-layer (self-attention and feed-forward layers) followed by layer normalization. This allows gradients to flow through the network more effectively, preventing the vanishing gradient problem and stabilizing training.
- **Encoder and decoder stacks:** The Transformer architecture consists of two main stacks: an encoder stack and a decoder stack. The encoder stack is composed of multiple identical layers, each containing a multi-head self-attention mechanism and a feed-forward network. The decoder stack also has multiple identical layers but includes an additional "encoder-decoder attention" layer that helps the decoder focus on relevant parts of the input sequence when generating the output.

2.2.3 Example of Self-Attention

To understand how self-attention works, consider the sentence: *"Brazil is the largest country in South America, and its Amazon rainforest is a global treasure."* When reading this, humans understand that *"Brazil"* is the subject and that *"Amazon rainforest"* is an important feature associated with it, even though these words are separated by several tokens.

The **self-attention mechanism** operates within a single sequence and allows every word to relate to every other word in that same sequence. For example, when processing the word *"treasure"*, the model not only considers *"Amazon rainforest"* (its immediate context) but also *"Brazil"* (the subject) and *"largest country"* (a descriptive phrase). This ensures that the model captures all dependencies in the sequence, regardless of their distance.

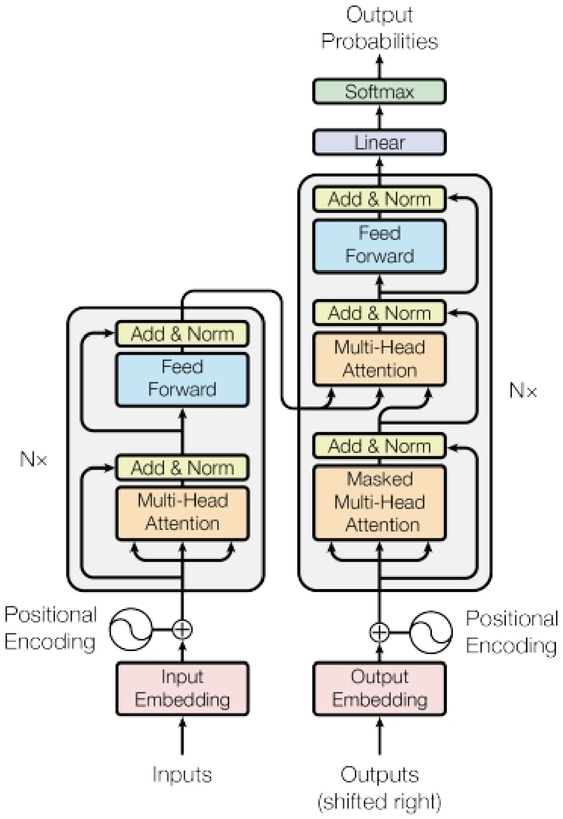


Figure 2.1: Original Transformer architecture [Vaswani et al., 2017]

Self-attention is computed by comparing each word (query) with every other word (key), assigning weights to the relevant information (value). Formally, this is calculated as [Vaswani et al., 2017]:

$$\text{SelfAttention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V,$$

where:

- Q (Query) corresponds to the word currently being processed.
- K (Key) represents the words being compared to the query.
- V (Value) carries the information to be integrated for each word.
- d_k is the dimension of the keys, used to scale the dot product for numerical stability.

By enabling each word to influence the interpretation of every other word, self-attention can capture both local and long-distance dependencies more effectively than traditional recurrent or convolutional approaches. Unlike RNNs, which process sequences step-by-step and rely on hidden states to carry information, or CNNs, which focus primarily on local receptive fields, the Transformer processes the entire sequence in parallel. It also integrates positional embeddings to encode the ordering of tokens, ensuring that even though words are handled concurrently, their relative positions in the text are preserved.

2.2.4 Pretrained Language Models

Pretrained language models are a significant advancement in NLP, leveraging the foundational strengths of the Transformer architecture to achieve impressive performance across a wide variety of language tasks. These models are trained on large corpora of text data in an unsupervised or self-supervised manner, learning general language representations that can be fine-tuned for specific downstream tasks [Devlin et al., 2019].

The key idea behind pretrained language models is to build a robust understanding of language by learning from vast amounts of text data. This pretraining phase typically involves objectives such as language modeling, where the model predicts the next word in a sentence, or masked language modeling, where certain words in the input are randomly masked, and the model must predict them (bidirectional). By doing this, the model captures rich contextual representations that generalize well to various applications, such as sentiment analysis, named entity recognition, and machine translation [Qiu et al., 2020].

Popular pretrained language models include:

- **BERT (Bidirectional Encoder Representations from Transformers):** Proposed by [Devlin et al., 2019], BERT uses a bidirectional training approach that considers both left and right contexts, enhancing its understanding of word meaning. It is pretrained with two tasks: Masked Language Modeling (MLM), where random tokens are masked and predicted, and Next Sentence Prediction (NSP), which trains the model to predict whether one sentence follows another in context.

- **GPT (Generative Pre-trained Transformer):** Introduced by [Radford and Narasimhan, 2018], GPT uses an autoregressive approach where the model generates text by predicting the next word in a sequence based on the preceding context. This method has demonstrated the effectiveness of generative models pre-trained on large datasets and fine-tuned for specific tasks.

- **RoBERTa (A Robustly Optimized BERT Pretraining Approach):** An extension of BERT, RoBERTa, proposed by [Liu et al., 2020b], introduces modifications to the pretraining process to improve performance. These updates include removing the Next Sentence Prediction (NSP) objective, training on significantly larger datasets, using dynamic masking during pretraining (instead of static masking), increasing batch sizes, and training for longer durations. These optimizations allow RoBERTa to leverage the Transformer architecture more effectively, leading to superior performance on a wide range of NLP benchmarks compared to the original BERT model.

- **T5 (Text-to-Text Transfer Transformer):** Developed by [Raffel et al., 2020], T5 treats every NLP task as a text-to-text problem, using the same model, objective, and training procedure for different tasks. This unified approach allows T5 to achieve high performance across a wide range of NLP tasks, from translation to summarization and question answering.

These pretrained language models have advanced NLP by learning transferable representations from large-scale text, allowing them to perform well on many tasks even with limited task-specific labeled data. The Transformer architecture in these models enables a deep understanding of language semantics and context, leading to state-of-the-art results across various language tasks [Qiu et al., 2020]. Recent works have also explored incorporating layout information to further enhance document understanding [de Lucena Drumond et al., 2023].

2.2.5 Protein Language Models

The paradigm of pretrained language models in NLP has rapidly expanded into the biological domain, with protein sequences now being treated as sentences. Similar to how large unlabeled text corpora enable the learning of rich linguistic representations, the abundance of unlabeled protein sequences provides an opportunity to learn biological representations without explicit supervision. Indeed, large-scale self-supervised models have proven capable of extracting meaningful biochemical and structural information from raw sequences alone. The Evolutionary Scale Modeling (ESM) model [Rives et al., 2021] was one of the first Transformer models for proteins. It uses a deep encoder architecture (34-layer Transformer, similar to BERT) and is pretrained with a masked language modeling (MLM) objective on an evolutionary-scale dataset of protein sequences. By training on 250 million diverse protein sequences, [Rives et al., 2021] showed that *information emerges in the learned representations* about fundamental protein properties (secondary structure, contacts, etc.) even without any aligned sequences or labels. These models have rapidly gained attention due to their ability to generate versatile protein sequence embeddings that can be applied to a wide range of downstream tasks, such as structure prediction, often reducing dependency on evolutionary alignments or manually created features [Heinzinger et al., 2019].

In parallel to the development of ESM, the ProtTrans project [Elnaggar et al., 2021] also explored the use of pretrained language models for protein sequences by adapting

NLP architectures such as BERT, in which amino acids are treated as individual tokens. The ProtTrans framework encompasses a family of models, including ProtBERT, ProtT5, and ProtAlbert, among others. These models were pretrained on extremely large protein corporacomprising up to 393 billion amino acids from datasets such as BFD and UniRef. This large-scale pretraining enabled the models to capture rich biophysical properties of proteins, leading to improved performance across a variety of downstream tasks. Among these, ProtBERT, an encoder-only Transformer model following the original BERT architecture and trained with the masked language modeling (MLM) objective, demonstrated significant gains in tasks such as secondary structure prediction.

Another model developed within the ProtTrans project is ProtT5 [Elnaggar et al., 2021], which leverages the T5 sequence-to-sequence architecture for protein modeling. Unlike ProtBERT, which uses an encoder-only Transformer, ProtT5 adopts a full encoder-decoder Transformer architecture, following the original design of the T5 model. It is trained using BERTs masked language modeling objective, in which individual amino acids are randomly masked in the input, and the model is trained to reconstruct these masked tokens based on their context.

A distinct approach to protein language modeling is proposed by ProteinBERT [Brandes et al., 2022], which incorporates biological knowledge directly into the pretraining process. Its architecture combines a Transformer encoder with convolutional layers to capture both global and local sequence patterns, tailored to protein-specific characteristics. ProteinBERT adopts a multi-task training strategy that unites masked language modeling with the prediction of gene ontology (GO) annotations, enabling the model to learn both structural and functional representations. [Brandes et al., 2022] report that ProteinBERT achieves competitive performance across a range of benchmarks, sometimes surpassing larger models trained with substantially greater computational resources.

Ankh [Elnaggar et al., 2023] is a more recent protein language model that shifts the focus from model scale to optimization and efficiency. It adopts an encoder-decoder Transformer architecture and incorporates several protein-specific design refinements based on extensive empirical evaluation, including variations in masking strategies, model depth, and embedding dimensions. The central goal of Ankh is to achieve state-of-the-art performance with fewer parameters, improving accessibility and reducing computational costs. [Elnaggar et al., 2023] report that, despite its smaller size and reduced embedding dimension, Ankh matches or surpasses the performance of earlier large-scale models across a range of structure and function prediction benchmarks.

Model	Architecture	Year
ESM	Encoder (Transformer)	2021
ProtBERT	Encoder (Transformer)	2021
ProtT5	Encoder-Decoder (Transformer)	2021
ProteinBERT	Encoder (Transformer + Conv)	2022
Ankh	Encoder-Decoder (Transformer)	2023

Table 2.1: Architectural comparison of selected protein language models and their respective architecture and year.

2.3 Conclusion

The progression of language modeling has been traced from early *n*-gram and Hidden Markov Models, which captured local context but faced limitations with unseen data, to RNNs and CNNs, which improved sequential modeling but struggled with long-range dependencies [Bengio et al., 1994, Chen and Goodman, 1999, Kim, 2014, Rabiner, 1989]. The introduction of attention-based Transformers marked a pivotal advancement, enabling parallel sequence processing and more effective context modeling [Vaswani et al., 2017]. This development facilitated the rise of pretrained language models (PLMs) such as BERT, GPT, and RoBERTa, which leverage large-scale corpora to learn transferable representations, improving performance across various NLP tasks [Devlin et al., 2019, Liu et al., 2020b, Radford and Narasimhan, 2018]. Recently, this paradigm has been extended to biology through protein language models, which treat protein sequences such as natural language sentences.

While PLMs have revolutionized NLP by capturing deep contextual relationships and enabling efficient fine-tuning for diverse applications, they also introduce new challenges that must be addressed:

- **Computational cost**: Transformer models exhibit poor scalability with sequence length, requiring substantial hardware resources [Vaswani et al., 2017].
- **Data constraints and bias**: Large corpora collected from the web, while enabling broad generalization, often encode societal biases that are difficult to mitigate [Bender et al., 2021].
- **Interpretability**: While attention mechanisms are often used to explain model decisions, their weight distributions may not always accurately reflect the true importance of input components [Jain and Wallace, 2019, Serrano and Smith, 2019].

A particularly relevant research direction is the adaptation of PLMs to specialized domains, where labeled data may be scarce. A common approach to addressing this challenge is transfer learning, which leverages knowledge from pre-trained models to improve performance on new tasks. However, when new domains experience significant distributional changes compared to the original data — a phenomenon known as *domain shift* [Pan and Yang, 2010] — it becomes necessary to go beyond simply using pre-trained models. In this context, the field of *domain adaptation* emerges as a key research area explored in the following chapter.

Chapter 3

Domain Adaptation

3.1 Introduction

This chapter presents key concepts and techniques for addressing domain shift in real scenarios. As a subfield of transfer learning, domain adaptation leverages knowledge from one domain to improve learning in another, especially when their data distributions differ. It begins with an overview of traditional domain adaptation methods, emphasizing strategies for settings where domain data exhibit distributional changes while remaining relatively stable.

The discussion then progresses to supervision regimes in domain adaptation: supervised, semi-supervised and unsupervised domain adaptation. The chapter also explores multi-source domain adaptation approaches, which harness data from multiple source domains to improve overall performance. Finally, it examines hierarchical domain adaptation, where domains are organized within a hierarchical structure.

3.2 Techniques

Domain Adaptation (DA) methods focus on improving the performance of a model on a target domain by adapting it from one source domain, which have different data distributions from the target domain [Ramponi and Plank, 2020]. These methods operate under the assumption that both the source and unlabelled target data are available during training. The primary goal of these methods is to minimize the mismatch between the source and target distributions, improving the model's ability to generalize effectively to the target domain [Csurka, 2017].

A domain is characterized by a specific data distribution, represented by a joint probability distribution P_{XY} over the input space X and the label space Y. In this context, the terms "domain" and "distribution" are used interchangeably. Let $S =$

$((x_1^s, \ldots, x_n^s), (y_1^s, \ldots, y_n^s))$ denote the source domain, where x_i^s represents input features and y_i^s represents corresponding labels. Similarly, let $T = ((x_1^t, \ldots, x_n^t), (y_1^t, \ldots, y_n^t))$ represent the target domain, with x_i^t as input features and y_i^t as corresponding labels. The source and target domains are associated with probability distributions P_s and P_t, respectively, where $P_s \neq P_t$, indicating that the data distributions differ between the two domains [Ben-David et al., 2010]. The objective of DA is to construct a model that effectively leverages the information from the source domain S to accurately predict the labels y_t in the target domain T.

3.2.1 Discrepancy-Based Methods

Discrepancy-based methods focus on reducing the statistical divergence between source and target domains [Ben-David et al., 2010]. Common measures include the Maximum Mean Discrepancy (MMD), which quantifies the distance between distributions in a reproducing kernel Hilbert space. For instance, the Deep Domain Confusion (DDC) method [Tzeng et al., 2014] incorporates an MMD-based regularization term into a deep network's loss to align feature distributions across domains.

An illustrative example in NLP involves adapting a sentiment classifier trained on Brazilian Portuguese news articles to Brazilian social media content. News text tends to be formal and structured, whereas social media text contains colloquialisms and abbreviations. By minimizing MMD, discrepancy-based methods (e.g., DDC) help the model bridge these textual style gaps.

The Joint Adaptation Network (JAN) [Long et al., 2017] extends this idea by jointly aligning marginal and conditional distributions. This approach is particularly beneficial in cases where class-conditional distributions differ notably across domains. For named entity recognition (NER) tasks, for instance, the same entity (e.g., "Brazil") may appear in distinct syntactic or semantic contexts in legal documents versus news articles. JAN mitigates such differences through the concurrent alignment of feature and label distributions.

3.2.2 Adversarial Methods

Adversarial methods are inspired by the principles of Generative Adversarial Networks (GANs) [Goodfellow et al., 2020], leveraging adversarial training to learn domain-invariant features. An example is the Domain-Adversarial Neural Network (DANN) [Ganin et al., 2016], which introduces a gradient reversal layer to establish a competition between a feature extractor and a domain discriminator. The feature extractor is trained to create domain-invariant representations, while the domain discriminator attempts to differentiate between the source and target domains. The gradient reversal layer acts as a bridge, flipping the gradient direction from the domain discriminator during backpropagation. This process encourages the feature extractor to generate representations that make it harder for the domain discriminator to distinguish between domains, providing better generalization to the target domain.

A potential NLP application involves adapting a text classification model trained on formal Brazilian Portuguese to regional dialects across the country. DANN enables the extraction of linguistic features that remain invariant to dialectal variations, allowing a single classifier to handle inputs from diverse regions (e.g., São Paulo and Bahia).

Another approach, Adversarial Discriminative Domain Adaptation (ADDA) [Tzeng et al., 2017], refines this concept by employing separate feature extractors for source and target data, along with a shared discriminator. In multilingual settings, such as adapting a chatbot from English to Brazilian Portuguese, ADDA trains feature extractors independently for each language while a discriminator seeks to differentiate the learned features, pushing both languages toward a shared representation space.

3.2.3 Reconstruction-Based Methods

Reconstruction-based methods utilize autoencoders or generative architectures to learn representations common to both domains. The Deep Reconstruction-Classification Network (DRCN) [Ghifary et al., 2016] simultaneously minimizes reconstruction and classification losses via a shared encoder. This method is advantageous when the domains share structural similarities but differ in specific aspects.

In the field of language translation, for example, a system originally trained on Spanish may be adapted to Portuguese by jointly reconstructing the original sentences and classifying them (e.g., by topic or sentiment). Despite being distinct languages, Spanish and Portuguese share similarities in grammar and vocabulary due to their shared Latin roots. The model retains these shared linguistic structures while learning to adjust for lexical and stylistic differences between the two languages.

Reconstruction-based approaches are also well-suited for Automatic Speech Recognition (ASR) tasks. For instance, a model trained on American English can be adapted to recognize British English by learning to reconstruct audio signals that capture shared phonetic traits, while also accounting for differences in pronunciation, such as the tendency for 'r' to be silent in British accents. This strategy enables the model to generalize across accent variations within the English language.

3.2.4 Normalization-Based Approaches

Normalization-based approaches address domain shifts by recalibrating feature statistics. Adaptive Batch Normalization (AdaBN) [Li et al., 2017] replaces source-domain statistics (mean and variance) with those from the target domain in batch normalization layers, improving generalization when data distributions diverge substantially. For instance, AdaBN can help adapt NLP models from formal to informal domains - such as from formal texts to social media language in Brazilian Portuguese - by updating normalization statistics in the layers that process token embeddings.

Instance Normalization [Ulyanov et al., 2016] is another approach that is considered valuable in handling stylistic differences, as commonly seen in style transfer tasks. It normalizes the activations of each individual example by subtracting its own mean and dividing by its own standard deviation, independently of the rest of the batch. It can be adapted to NLP scenarios where text style varies significantly. By applying instance-level normalization, the model preserves semantic content while reducing stylistic variance.

3.2.5 Hybrid Methods

Hybrid methods combine elements from the aforementioned strategies to leverage their strengths while mitigating limitations. A representative example is the Adversarial Discriminative Domain Adaptation (ADDA) [Tzeng et al., 2017], which combines discriminative modeling with adversarial training to align feature representations across domains.

This type of integration holds potential for cross-lingual natural language processing, where the goal is to learn shared representations across languages with distinct structures. By leveraging adversarial objectives to align distributions while preserving discriminative capacity, such methods can facilitate knowledge transfer between high- and low-resource languages, even in settings with limited annotated data [Chen et al., 2018].

3.3 Supervision Regimes in Domain Adaptation

While the previous section categorized domain adaptation methods based on their underlying adaptation techniques, this section provides a complementary classification according to the level of supervision available in the target domain. These adaptation methods can be applied under different supervision regimes.

3.3.1 Supervised Domain Adaptation

Supervised Domain Adaptation (SDA) assumes access to labeled data from both the source and target domains. This approach is considered one of the simplest forms of DA, as the availability of labeled data in both domains provides a clear path for adaptation by allowing the model to learn domain-specific differences [Ben-David et al., 2010]. Early work in this field includes methods that modify feature representations to create shared and domain-specific features, which are then used to train a supervised classifier on both domains. For instance, [Daumé III, 2007] proposed a feature augmentation technique where input features are split into general and domain-specific components, simplifying the adaptation process. More recent approaches, such as the Contrastive Domain Adaptation Framework (CCSA) [Motiian et al., 2017], leverage siamese neural networks to learn a common embedding space. These networks align the two domains by minimizing the distance between instances of the same class while preserving inter-class separation.

These advancements demonstrate the evolution of SDA from foundational approaches centered on feature transformations to contemporary embedding-based methods that utilize contrastive learning. By progressively building on these innovations, SDA techniques have become increasingly effective in addressing the complex challenges of domain shifts.

Mathematical Formulation: Given a labeled dataset $S = (S_S, S_T)$, where $S_S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ and $S_T = \{(x_i^t, y_i^t)\}_{i=1}^{n_t}$ correspond to source and target domains respectively, the goal of supervised domain adaptation is to learn a model f that minimizes a weighted average loss:

$$\hat{\epsilon}_\alpha(f) = \alpha \cdot \hat{\epsilon}_T(f) + (1 - \alpha) \cdot \hat{\epsilon}_S(f),$$

where $\hat{\epsilon}_T(f)$ and $\hat{\epsilon}_S(f)$ are empirical losses over target and source data, and $\alpha \in [0, 1]$ adjusts their relative importance. This weighting is particularly relevant when $n_t \ll n_s$, as commonly occurs in domain adaptation scenarios [Ben-David et al., 2010].

3.3.2 Semi-Supervised Domain Adaptation

Semi-Supervised Domain Adaptation (SSDA) addresses scenarios where labeled data is abundant in the source domain but scarce in the target domain. This approach combines limited labeled target data with abundant labeled source data and unlabeled target data to improve cross-domain generalization. A key strategy in SSDA is pseudo-labeling, where the model generates labels for unlabeled target samples and iteratively refines them using techniques such as consistency regularization [Tarvainen and Valpola, 2017] or contrastive alignment [Kang et al., 2022]. For instance, the Contrastive Adaptation Network (CAN) aligns source and target domains by minimizing intra-class discrepancies and maximizing inter-class separation [Kang et al., 2022].

Mathematical Formulation: Following the semi-supervised learning mathematical formulation inspired by [Yang et al., 2023], consider a labeled source domain $S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$, a small labeled target subset $T_L = \{(x_i^t, y_i^t)\}_{i=1}^{m_t}$, and an unlabeled target subset $T_U = \{x_i^t\}_{i=1}^{n_t}$. The objective of Semi-Supervised Domain Adaptation is to minimize the following objective function:

$$\min_f \sum_{(x,y)\in S\cup T_L} \mathcal{L}_s(f(x), y) + \alpha \sum_{x\in T_U} \mathcal{L}_u(f(x)) + \beta \sum_{x\in S\cup T_L\cup T_U} R(x),$$

where:

- $\mathcal{L}_s$ is the supervised loss (e.g., cross-entropy),
- $\mathcal{L}_u$ is the unsupervised loss applied to unlabeled target samples (e.g., entropy minimization),
- $R(x)$ is a regularization term (e.g., consistency regularization),
- α and β are trade-off hyperparameters controlling the influence of the unsupervised and regularization terms, respectively.

3.3.3 Unsupervised Domain Adaptation

Unsupervised Domain Adaptation (UDA) addresses learning scenarios where labeled data is available only in the source domain, while the target domain contains only unlabeled data. The objective is to learn a model that performs well on the target domain, despite the domain shift.

Mathematical Formulation: Given a labeled dataset $S_S = \{(x_i^s, y_i^s)\}_{i=1}^{n_s}$ from the source domain and an unlabeled dataset $S_T = \{x_i^t\}_{i=1}^{n_t}$ from the target domain, the goal is to learn a model f that minimizes the target error $\epsilon_T(f)$. Although target labels are unavailable during training, the domain adaptation theory introduced by [Ben-David et al., 2010] provides an upper bound for the target error:

$$\epsilon_T(f) \leq \epsilon_S(f) + \frac{1}{2}\hat{d}_{\mathcal{H}\Delta\mathcal{H}} + \lambda + \varepsilon,$$

where $\epsilon_S(f)$ is the source domain error, $d_{\mathcal{H}\Delta\mathcal{H}}(D_S, D_T)$ measures the divergence between source and target distributions, $\lambda = \min_{h\in\mathcal{H}} (\epsilon_S(h) + \epsilon_T(h))$ represents the lowest possible combined error that any hypothesis can achieve on both domains, and ε is a small generalization term that accounts for the discrepancy between the empirical and true divergence due to the use of finite unlabeled samples. λ is a theoretical constant that reflects the intrinsic difficulty of the adaptation problem: although it cannot be minimized directly due to the absence of target labels, it serves as a reference to assess the feasibility of adaptation. This bound shows that minimizing the source error and reducing the domain divergence are key strategies to achieve low target error in the absence of target labels.

Example: The task of offensive language identification [Rosenthal et al., 2021] aims to classify text posts as offensive or non-offensive. In this scenario, the source domain comprises posts from formal online forum with extensive labeled data, characterized by standardized language usage, formal expressions, and minimal slang. In contrast, the target domain consists of posts from a social media platform, which are typically informal, containing slang, abbreviations, emojis, and varied linguistic expressions.

This problem is analyzed across three distinct scenarios reflecting the availability of labeled data in the target domain:

- **SDA:** In this case, there is abundant labeled data from both the forum (source) and the social media platform (target). Using SDA methods, the model explicitly learns domain-specific differences. It leverages labeled examples from social media to adapt to informal language and diverse linguistic patterns, reducing classification errors caused by stylistic differences.

- **SSDA:** Here, abundant labeled data is available in the forum, but only a small set of labeled examples exists for social media. The remaining social media data remains unlabeled. SSDA techniques utilize the extensive labeled source data and the limited labeled target data to initially train the model. Methods such as pseudo-labeling could be employed on the unlabeled social media posts to refine predictions, enhancing the model's capability to classify offensive language within the informal context of social media.

- **UDA:** In the most challenging scenario, only labeled data from the forum is available, while all social media posts are unlabeled. UDA techniques aim to minimize the domain discrepancy between the forum and social media datasets without relying on explicit labels from the target domain. Approaches such as adversarial training, embedding alignment, or distribution matching could be applied to align the feature representations of formal forum texts and informal social media texts, enabling the model to identify offensive language in social media contexts, even in the absence of labeled social media examples.

This example highlights the incremental challenges posed by decreasing target domain labeling, demonstrating how domain adaptation methods progressively address and overcome these challenges.

3.4 Multi-Source Domain Adaptation

In many real-world applications, data may come from several different sources, each with its own distinct characteristics and distribution [Sun et al., 2015]. Multi-source domain adaptation (MSDA) aims to leverage multiple labeled source domains to improve performance on a target domain that is either unlabeled or sparsely labeled. Unlike traditional source domain adaptation, which assumes a single source domain, MSDA must integrate knowledge from diverse domains, each of which may exhibit different types of domain shifts [Mansour et al., 2008].

3.4.1 Problem Formulation

The MSDA framework assumes access to M source domains. Each source domain D_i is associated with a trained hypothesis h_i, which performs well on its own domain - that is, it makes small prediction errors under the distribution D_i.

The goal is to construct a new hypothesis h that performs well on a different domain, referred to as the target domain, whose data distribution D_T may be different from the source distributions. Even without assuming any specific relationship between D_T and the D_i, [Mansour et al., 2008] show that it is possible to combine the source hypotheses $h_1, \ldots, h_M$ in a way that ensures good performance on the target domain. Formally, if there exists a target function f such that each source hypothesis has low expected loss on its respective domain:

$$\mathcal{L}(D_i, h_i, f) \leq \epsilon \quad \text{for all } i = 1, \ldots, M,$$

then there exists a combined hypothesis h whose expected loss on the target domain is bounded by:

$$\mathcal{L}(D_T, h, f) \leq 3\epsilon + \delta,$$

for any small value $\delta > 0$. This result indicates that, under minimal assumptions, knowledge from multiple sources can be transferred to an arbitrary target domain.

For example, in offensive language identification, labeled data may be available from multiple sources such as social media platforms, online forums, and moderated comment sections. Each domain differs in vocabulary, style, and formality, and a separate classifier can be trained for each source. Now consider a target domain consisting of transcribed emergency call reports, where the language is more fragmented and urgent. This target domain may have little or no labeled data and exhibits characteristics different from the sources. According to [Mansour et al., 2008], it is possible to combine source models to derive a hypothesis that performs well on the target domain, even when the target is not strictly related to the sources. However, the effectiveness of this transfer depends on the divergence between the source and target distributions.

3.4.2 Approaches for MSDA in NLP

MSDA expands single-source domain adaptation by integrating multiple source domains to enhance generalization to a target domain [Sun et al., 2015]. This extension is particularly advantageous in NLP tasks, where data from various genres, registers, or thematic areas can be combined to create more robust training sets. Unlike single-source domain adaptation, MSDA addresses the challenge of learning from heterogeneous source distributions, providing robustness to domain shifts in the target domain.

Several strategies have been proposed for MSDA in NLP, each aiming to align the data distributions between source and target domains or to generate intermediate representations that bridge the gap between them. [Sun et al., 2015] present a systematic survey of MSDA methods in deep learning, highlighting strategies such as latent space transformation and intermediate domain generation. These techniques aim to align feature distributions across multiple source domains and a target domain, reducing discrepancies and enhancing model generalization. In NLP, this enables the adaptation of models trained on diverse corpora (e.g., news articles, social media texts, scientific literature) to more specialized domains (e.g., user reviews, technical reports).

A common MSDA approach involves aligning the feature spaces across multiple domains by minimizing discrepancies. For instance, when adapting a sentiment analysis model trained on Amazon product reviews, Yelp restaurant reviews, and IMDB movie reviews to a target domain such as X (formerly Twitter) data, measures like Maximum Mean Discrepancy (MMD) [Tolstikhin et al., 2016] or Wasserstein distance [Arjovsky et al., 2017] are often employed. These measures reduce distributional differences between text embeddings from various domains. Aligning representations across domains enhances the model's ability to generalize to the target domain [Xu et al., 2018].

Beyond feature alignment, several advanced approaches leverage domain relationships explicitly. A notable example is the Mixture of Experts framework proposed by [Guo et al., 2018], which captures relationships between multiple source domains and the target domain through a point-to-set metric. For instance, the Kitchen domain, which includes reviews on pans, cookbooks, and electronic devices, cannot be perfectly aligned with any single source domain such as Cookware, Books, or Electronics. Aggregating complementary information from multiple sources allows better approximation of the target distribution.

Another strategy is the DistanceNet-Bandits approach introduced by [Guo et al., 2020], which integrates reinforcement learning with MSDA. This technique employs a multi-armed bandit algorithm to dynamically select the most informative source domains, estimating their relevance to the target domain. In text classification scenarios involving product reviews, DistanceNet-Bandits has demonstrated improved performance by prioritizing the source domains most aligned with the target category, achieving more robust and computationally efficient adaptation.

Additionally, recognizing that domain shift may not be uniform, [Li et al., 2021] propose Dynamic Transfer, which treats domain adaptation as a dynamic, instance-specific phenomenon. Instead of assuming fixed domain alignments, Dynamic Transfer employs *Dynamic Residual Transfer (DRT)*, where residual matrices adapt the model parameters at the instance level. This method allows for fine-grained adaptation without requiring explicit domain labels, resulting in effective domain adaptation.

Finally, also addressing scenarios with limited labeled data, [Ren et al., 2022] introduce the Pseudo Target Domain Adaptation method. This technique constructs pseudo target domains - formed by combining pseudo-labeled target samples with labeled examples from a single source domain - enabling the model to better approximate the target distribution.

3.5 Hierarchical Domain Adaptation

This thesis considers Hierarchical Domain Adaptation (HDA) as a specialized form of MSDA that takes advantage of the hierarchical structure inherent in the data. Unlike MSDA techniques, which generally treat data from all source domains uniformly, HDA exploits hierarchical relationships among domains to align feature representations across different levels of the hierarchy. By incorporating hierarchical information, this approach enables finer-grained adaptations and more nuanced alignment, improving the model's performance on the target domain [Raj et al., 2014].

3.5.1 HDA as a special case of MSDA

In traditional MSDA, data from multiple source domains $\mathcal{S}_1, \mathcal{S}_2, \ldots, \mathcal{S}_M$ is used to train a model that generalizes to a target domain $\mathcal{T}$. For this thesis, HDA is considered to extend this concept by leveraging the hierarchical organization of the source and target domains. To illustrate, consider the evolutionary progression of languages. Languages evolve from shared ancestors and diverge into different dialects over time. In this scenario, the source domains correspond to various stages of linguistic evolution, structured hierarchically from a root language (e.g., Proto-Indo-European) down to its descendant languages (e.g., Latin, Greek, Sanskrit) and further down to modern languages and dialects (e.g., Italian, Spanish, Hindi). HDA seeks to align language models at each stage of this hierarchy, enabling knowledge transfer from ancestral languages to modern descendants. By leveraging this structure, HDA captures the gradual linguistic shifts that occur over time, enhancing model performance on target languages that lack sufficient labeled data.

This hierarchical organization enables finer-grained domain alignment, as intermediate domains (e.g., descendant languages) serve as bridges between broad ancestral domains and specific target domains. Such intermediate adaptations allow for smoother transitions and improved generalization, addressing challenges associated with significant domain shifts.

3.5.2 Approaches to HDA

HDA comprises strategies that use hierarchical relationships to enable adaptation across multiple domains. These approaches can be broadly categorized into two types: (1) methods that utilize hierarchy within the model architecture, and (2) methods that leverage hierarchical structures within the data.

Methods employing hierarchical structures in the model architecture align representations at multiple granularity levels, regardless of whether the datasets have explicit hierarchical organizations. For example, [Wen et al., 2022] proposed hierarchical alignment of local feature patterns, organizing features across multiple internal levels to improve generalization, but without hierarchical structure in domains.

In contrast, approaches that utilize hierarchical data structures assume that domains are organized into predefined hierarchical structures, such as taxonomic or phylogenetic trees. [Raj et al., 2014] presented a subspace-based method that adapts across hierarchical levels of semantic categories, facilitating transfer between closely related domains in the hierarchy. [Chronopoulou et al., 2022] proposed an adaptation framework for pretrained language models, arranging textual domains into a hierarchical tree structure to enable selective adaptation through adapters, resulting in improved performance and computational efficiency for related domains. [Finkel and Manning, 2009] introduced Hierarchical Bayesian Domain Adaptation, employing a hierarchical Bayesian prior to connect parameters across related domains, enabling information sharing between domains organized hierarchically.

In summary, while model-based hierarchical approaches capture various internal representational levels, methods that leverage data hierarchies may benefit from prior structural knowledge of domains, which can support more effective adaptations, especially in low-resource scenarios. The focus of this thesis lies on the methods that exploit hierarchical relationships present within the data.

3.6 Conclusion

This chapter reviewed the foundations DA, from traditional supervised, semi-supervised, and unsupervised approaches to recent advances involving MSDA and HDA methods. While traditional DA techniques focus on reducing distributional shifts between a single source and target domain, MSDA utilizes multiple sources to improve generalization, and HDA further incorporates hierarchical relationships between domains to enable finer-grained adaptation.

Each paradigm presents unique strengths and challenges: DA provides solid theoretical foundations but struggles with large domain shifts, MSDA enhances generalization but may suffer from negative transfer and increased complexity, and HDA enables granular adaptation across nested domains but often faces the challenge of negative transfer when the hierarchical relationships between domains are imperfect or when knowledge transfer is not properly balanced across levels.

This thesis addresses these limitations by proposing a *Hierarchical Domain Adaptation Method for Neural Language Models* that introduces a hierarchical weighted loss, explicitly balancing the contribution of higher- and lower-level domains. This approach is designed to enable smoother knowledge transfer throughout the hierarchy, reducing the risk of negative transfer and improving performance, particularly in low-resource scenarios.

Chapter 4

The Epitope Prediction Problem

4.1 Introduction

This chapter provides the theoretical foundation necessary for understanding epitope prediction, as the proposed method in this thesis is applied to improve linear B-cell epitope prediction. It introduces the core concepts, explores associated challenges, and highlights key applications of epitope prediction, encompassing both traditional and advanced computational approaches. In addition, it emphasizes the importance of taxonomic levels, illustrating how the phylogenetic tree can enable the transfer of knowledge from abundant taxons to closely related species with limited data. The phylogenetic structure inherently exhibits a hierarchical organization of data, making it an ideal setting for applying the hierarchical domain adaptation approach proposed in this thesis.

4.2 Immunological Context

An *antigenic determinant*, or B-cell epitope (BCE), is the specific region on an antigen that interacts with the receptor of a B-cell, eliciting an immune response. When a host organism encounters a pathogen such as a virus or bacterium, the immune system's B-cells[1] bind to antigens[2] through their B-cell receptors[3] and produce antibodies. In protein antigens, epitopes can be contiguous amino acid sequences or non-adjacent regions brought into proximity by protein folding [Ponomarenko and Van Regenmortel, 2009, Sanchez-Trincado et al., 2017].

[1] B cells are white blood cells that produce antibodies to combat foreign substances such as pathogens.

[2] An antigen is any substance capable of inducing an immune response [Merriam-Webster, 2023].

[3] B-cell receptors (BCRs) are specialized membrane-bound proteins found on the surface of B cells, a type of lymphocyte.

4.2.1 Linear B-cell Epitopes

Linear B-cell epitopes (LBCEs) refer to continuous segments of amino acids within a protein that can be recognized by the immune system. Identifying LBCEs has significant implications for immunodiagnostics, vaccine design, and therapeutic antibody development. Experimental identification of LBCEs, however, is typically costly, labor-intensive, and time-consuming [Ashford et al., 2021]. As a result, researchers have sought to develop computational methods that utilize protein sequence data to predict these epitopes efficiently and at scale.

LBCEs are especially attractive targets in diagnostic development because they are straightforward to synthesize and their recognition is preserved even after loss of native structure, which is advantageous for many antibody-based assays [Forsström et al., 2015]. Nevertheless, the scarcity of training data in many organisms — particularly novel or understudied pathogens — can negatively affect predictive performance. The method proposed in this thesis addresses this challenge by leveraging transfer learning at higher hierarchical levels, where data are more abundant, and adapting to lower levels, where data become scarcer. This structured approach enables more effective learning in cases where little labeled data are available in the target domain.

4.2.2 Importance for Diagnostics and Vaccines

Recent studies demonstrate the benefits of accurate epitope prediction in diagnostics [Jiang et al., 2023]. By identifying highly antigenic regions that induce strong immune responses, diagnostic tests can achieve better sensitivity and specificity, reducing false positives and improving early disease detection [Campelo et al., 2024]. These benefits are especially valuable in resource-limited settings, where cost-effectiveness and rapid development cycles are crucial.

In the context of vaccines, LBCE prediction supports the identification of immunodominant regions that incite robust and targeted immune responses. Traditional vaccines often rely on whole pathogens (inactivated or attenuated) or large subunits that may pose safety risks, especially to immunocompromised individuals [Zhang and Ulery, 2018]. By contrast, synthetic peptide-based vaccines derived from LBCEs are non-infectious, simpler to manufacture, and can be combined to generate multivalent protection [Ponomarenko and Van Regenmortel, 2009].

4.3 Epitope prediction task

The humoral immune response in mammals relies on antibodies, which are essential for recognizing and binding to antigens [Janeway, 2012]. B-cell epitopes (BCEs) represent specific regions of an antigenic molecule that interact with antibodies [Sanchez-Trincado et al., 2017]. These epitopes may originate either from contiguous sequences of amino acid residues in the primary protein structure or from non-contiguous regions brought together by protein folding. The former are commonly known as *linear B-cell epitopes*, while the latter are referred to as *conformational* BCEs. Identifying epitopes is a critical step in a wide array of medical and immunological applications, including vaccine development [Hamley, 2022], therapeutic antibody design [Sun et al., 2024], and immunodiagnostics [Mucci et al., 2017].

Deep learning techniques are becoming increasingly common for modelling and analyzing protein data [Bahai et al., 2021, Collatz et al., 2020], with transfer learning emerging as a notable approach [Fenoy et al., 2022, Liu et al., 2024, Schmirler et al., 2024]. Several methods involve using latent space vector representations of amino acid residues that are extracted from large, pre-trained protein language models. These representations have the ability to encode biological properties of proteins in a context-dependent manner [Elnaggar et al., 2021, Rives et al., 2021], making them a compelling option for the development of new models capable of capturing the immunogenic properties of peptides. As a particular example, the Evolutionary Scale Models (ESM) for protein representation [Lin et al., 2023, Rives et al., 2021] employ self-supervised learning and are based on the Transformer architecture, which has proven to be a powerful general-purpose framework for representation learning and generative modeling, outperforming recurrent and convolutional architectures in natural language domains [Vaswani et al., 2017]. These models provide powerful representations of protein properties that encode useful information for a variety of downstream modelling tasks, including the prediction of B-cell epitopes [Clifford et al., 2022].

Most methods for predicting LBCEs, including new approaches that use large protein language models [Clifford et al., 2022], are trained using datasets containing labeled peptide sequences from a phylogenetically diverse range of organisms. The main goal of epitope prediction models developed using such heterogeneous datasets is to provide general-purpose models. There is, however, evidence that pathogen- or taxon-specific models can result in improved performance in the usual scenario where predictions are made with a specific target pathogen [Ashford et al., 2021, Campelo et al., 2024]. The hierarchical domain adaptation method proposed in this thesis utilizes taxon-specific data to improve the performance of LBCE predictions.

4.4 Biological Taxonomies

Taxonomy is a hierarchical system used to organize and categorize living organisms according to their evolutionary relationships, morphological traits, and genetic characteristics. This framework encompasses the major ranks of **domain**, **kingdom**, **phylum**, **class**, **order**, **family**, **genus**, and **species**, moving from the most general (domain) to the most specific (species) - Figure 4.1. Such classifications enable researchers to systematically contextualize the morphology, genetics and ecological niche of an organism [Woese et al., 1990].

However, viruses pose a unique challenge to taxonomy. Since they are not strictly considered *living* organisms, they follow a distinct classification system established by the International Committee on Taxonomy of Viruses (ICTV). The ICTV taxonomy organizes viruses into hierarchical ranks such as **realm**, **kingdom**, **phylum**, **class**, **order**, **family**, **genus**, and **species**, similar to the classification of cellular organisms but specifically adapted to the unique features of viruses. These adaptations consider aspects like the type of genomic material, replication mechanisms, host range, and structural components. This system provides a unified framework for classifying and understanding viral diversity [Gorbalenya et al., 2020].

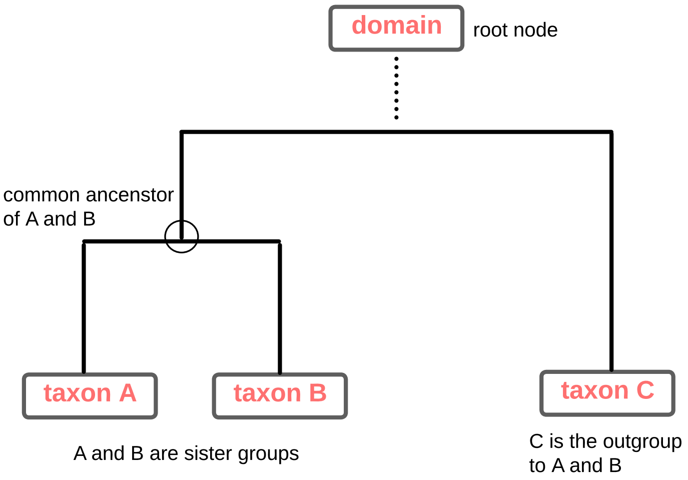


Figure 4.1: The diagram represents the progression from general categories, such as domain, kingdom, phylum, class, order, family, and genus, to the most specific level, species. In this hierarchy, a **taxon** can represent any category at any level, from the broadest (e.g., domain) to the most specific (e.g., species). Taxa A and B are sister groups, sharing a common ancestor, while taxon C serves as the outgroup to A and B. This diagram was created by the author, inspired by [Collins et al., 2020].

4.4.1 Leveraging Taxonomic Levels

Taxonomic information can improve the identification of linear B-cell epitopes by leveraging high-level organism classification, which helps distinguish epitopes from non-epitopes across diverse species [da Silva et al., 2023]. These evolutionary signals not only reveal conserved epitopes but also enable researchers to identify antibodies that can neutralize a wide range of emerging viral variants [Ishimaru et al., 2024]. Indeed, comparative analyses frequently detect cross-neutralizing epitopes[4] shared within the same family or genus, reinforcing the notion that certain antigenic features remain highly conserved over evolutionary time [Huang et al., 2023].

Using evolutionary patterns driven by phylogenetics, researchers can prioritize candidate epitopes under selective constraints, reducing experimental workloads and improving the identification of immune targets [Lacerda et al., 2010]. Furthermore, domain adaptation techniques in machine learning can incorporate these taxonomic signals to prioritize features that are conserved across lineages, mitigating overfitting and boosting predictive performance on novel pathogens. However, relatively few LBCE predictors integrate taxonomic or phylogenetic insights into their pipelines. The hierarchical domain adaptation strategy outlined in this thesis aims to address this gap by:

- Weighting training data based on evolutionary proximity;
- Enhancing generalization by transferring immunologically relevant features from better-represented organisms.

4.5 Related Work on Epitope Prediction

The concept that the statistical patterns present in protein sequences contain crucial information about their biological function and structure is grounded in scientific research, as demonstrated by previous studies [Altschuh et al., 1987, Yanofsky et al., 1964]. During evolution, sequences that correlate with the most favorable fitness [5].

The evolutionary process typically favors fitness-related outcomes selected from a wide range of possible random perturbations. The unobservable factors that govern the contribution of a protein to fitness - such as its stability, structure, and function - are

[4] Cross-neutralizing epitopes are antigenic regions that are conserved across different species or strains, allowing antibodies generated against one pathogen to neutralize others within the same evolutionary group.

[5] The biological term fitness describes an organism's ability to survive and reproduce in its environment based on its genetic features. It serves as a measure of the genetic contribution of the organism to the next generation[Orr, 2009]

indirectly captured in the distribution of naturally occurring sequences [Göbel et al., 1994].

In machine learning, natural language understanding leverages the distributional hypothesis, which states that word meaning can be inferred from contextual usage. Self-supervised learning has recently become a central approach, using unlabeled data to predict elements such as the next word in a sentence or masked words in context, without manual annotation [Bengio et al., 2000, Devlin et al., 2019]. This allows models to benefit from large datasets. Recent work shows that, when combined with large data and powerful architectures, self-supervised methods achieve state-of-the-art performance in tasks like question answering, semantic reasoning, and deep learning for protein modeling [Devlin et al., 2019, Rives et al., 2021].

4.5.1 Physicochemical properties-based methods

In the early days of epitope prediction, researchers focused on evaluating individual physiochemical properties of amino acids to identify potential epitopes. They examined properties such as flexibility [Karplus and Schulz, 1985], surface accessibility [Emini et al., 1985], hydrophobicity [Levitt, 1976], and antigenicity [Kolaskar and Tongaonkar, 1990]. Researchers developed algorithms that utilize sliding windows along the protein sequence to calculate average amino acid propensity scales. Regions of the protein that scored above a certain cut-off on these scales were identified as potential linear B-cell epitopes. However, it was later determined that relying solely on 484 propensity scales is not reliable enough for accurately detecting BCEs [Blythe and Flower, 2005]. To address the limitations of using individual physiochemical properties and propensity scales, more advanced epitope prediction methods have been developed. These methods employ a variety of approaches, including sequence-based algorithms, structural modeling, and machine learning techniques.

4.5.2 Machine learning-based methods

Machine learning (ML) methods have emerged as a powerful tool for predicting linear B-cell epitopes in proteins. These approaches move beyond traditional single-feature propensity scales by leveraging machine learning algorithms that can integrate a wide range of sequence-derived and physicochemical features to improve prediction accuracy [Yang and Yu, 2009]. Examples of popular tools utilizing machine learning methods for B-cell epitope prediction include BepiPred [Larsen et al., 2006], ABCPred [Saha and Raghava, 2006], LBTope [Singh et al., 2013], APCPred [Shen et al., 2015], iBCE-El

[Manavalan et al., 2018], BepiPred 2.0 [Jespersen et al., 2017], DLBEpitope [Liu et al., 2020a], EpiDope [Collatz et al., 2020], and EpitopeVec [Bahai et al., 2021].

Although machine learning methods have improved B-cell epitope prediction compared to traditional approaches, significant challenges remain. These methods still struggle with generalization and often require large amounts of high-quality data [Jespersen et al., 2017, Manavalan et al., 2018].

4.5.3 Deep Learning Methods

Deep learning techniques are increasingly utilized in protein analysis, and transfer learning is emerging as an effective approach. These methods use the latent vector representations of amino acid residues derived from large, pre-trained protein language models [Chowdhury et al., 2022, Elnaggar et al., 2021]. These representations encode structural, functional, and physicochemical properties of proteins in a context-dependent manner, making them a robust foundation for developing models that characterize the immunogenic properties of amino acid residues [Rives et al., 2021].

Deep Learning for Epitope Prediction

Deep learning has demonstrated significant potential in advancing epitope prediction accuracy. Among the methods evaluated, BepiPred 3.0 [Clifford et al., 2022] emerges as a notable advancement. By leveraging the pre-trained ESM-2 protein language model [Lin et al., 2023], BepiPred 3.0 enhances the prediction of both linear and conformational epitopes. The method achieves this by improving the annotation of epitope residues and integrating rich input features, including context-dependent embeddings from ESM-2 that encode structural and functional information for each amino acid. Furthermore, BepiPred 3.0 incorporates predicted surface accessibility scores, which help to more effectively discriminate between epitope and non-epitope residues.

Similarly, other methods such as EpiDope [Collatz et al., 2020] and EpitopeVec [Bahai et al., 2021] utilize deep learning to tackle the challenges of epitope prediction. EpiDope employs a deep neural network architecture with context-sensitive and non-context-sensitive embeddings to improve epitope predictions. Meanwhile, EpitopeVec leverages protein language model-based representations alongside residue properties and modified antigenicity scales, demonstrating superior performance compared to traditional tools [Jespersen et al., 2017].

These advancements underscore the potential of deep learning in epitope prediction, particularly when combined with pre-trained protein language models.

Transfer Learning in Protein Data

Transfer learning has become a key technique in addressing challenges associated with sparse labeled datasets in protein research. For example, [Bugnon et al., 2023] highlight the potential of transfer learning to annotate the vast protein universe, where only a fraction of protein sequences in UniProtKB are functionally characterized. By employing self-supervised learning on large unannotated datasets followed by fine-tuning on smaller labeled datasets, they demonstrated a significant reduction in prediction errors for protein family classification.

Other studies, such as [Heinzinger et al., 2019], explore SeqVec, a method for representing protein sequences as continuous vectors using ELMo [Peters et al., 2018]. This approach captures biophysical properties from unlabeled data, enabling effective protein prediction tasks. Similarly, [Shashkova et al., 2022] developed SEMA, a model fine-tuned on ESM-1v and ESM-IF1 protein language models to predict antibody-antigen interactions and identify epitopes with high accuracy. These studies demonstrate the power of transfer learning in extracting meaningful patterns from protein sequences, enhancing predictions even in data-scarce scenarios.

In summary, transfer learning represents a pivotal step forward in protein analysis, combining the ability to utilize large, pre-trained models with fine-tuning capabilities for specific tasks. This approach enables the development of more robust and generalizable tools for tasks such as structure prediction, detection of remote homologs, and protein engineering [Rao et al., 2019].

4.6 Discussion

This chapter provided an overview of LBCE prediction, detailing fundamental immunological principles and exploring the computational challenges to epitope identification. LBCEs have proven crucial for diagnostic applications and vaccine development, offering advantages in cost-efficiency and safety compared to traditional vaccine approaches.

This chapter also emphasized that phylogenetic relationships offer valuable signals for computational modeling in epitope prediction. By leveraging taxonomic hierarchies and evolutionary information, it is possible to transfer knowledge gained from well-studied pathogens to newly emerging or understudied organisms. Incorporating these evolutionary information into computational models has been shown to enhance prediction accuracy and help overcome the limitations imposed by scarce data [da Silva et al., 2023, Lacerda et al., 2010]. However, not all phylogenetic signals are equally relevant; determining which taxonomic levels provide meaningful information and how to use them remains a challenge.

While state-of-the-art epitope prediction methods such as BepiPred 3.0 leverage large pre-trained protein language models, their reliance on broadly diverse training sets can limit predictive accuracy. Recent evidence indicates that taxon-specific approaches can outperform generalized models [Ashford et al., 2021].

In response to these limitations, this thesis introduces a hierarchical domain adaptation approach based on taxon-specific protein language models. This strategy is supported by evidence that transfer learning from large protein datasets improves model performance across various tasks [Rao et al., 2019]. The proposed method is discussed in the next chapter.

Chapter 5

Domain Adaptation with a Protein LLM

This chapter presents the method proposed in this thesis: *A Hierarchical Domain Adaptation Method in Neural Language Models*. It begins by examining the dataset's neighborhood structure through t-SNE visualizations, which reveal the distribution and overlap of positive and negative samples from distinct pathogens, highlighting the potential benefits of taxon-specific modeling. Next, a single-domain adaptation approach is introduced to provide a foundation for exploring phylogeny-aware transfer learning strategies. Finally, the chapter defines and formalizes the proposed HDA method, which generalizes single-domain adaptation to leverage hierarchical domain structures.

5.1 Estimated density

To qualitatively investigate the neighborhood structure of the datasets, a t-SNE projection [van der Maaten and Hinton, 2008] of the whole data was used and later stratified by pathogen group. The analysis aimed to determine whether positive/negative data from distinct pathogens clustered around distinct regions of the feature space. Insights gathered from this projection could help explain the enhanced performance of taxon-specific models over generalist approaches, without however addressing the underlying biological mechanisms.

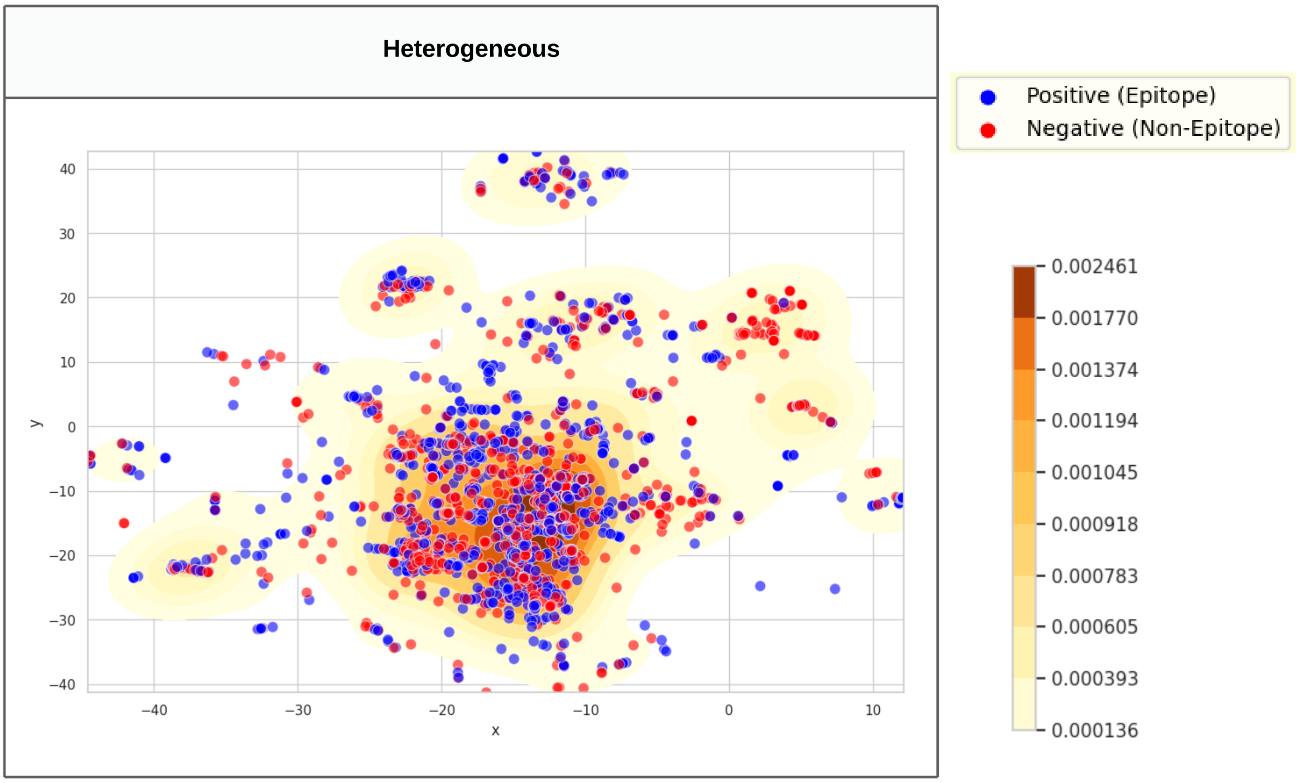


Figure 5.1: The projection was computed using a fraction of the entire dataset. For t-SNE training, 720 samples were employed, and the resulting model was applied to project 2,930 samples (as shown in the figure above) from a larger dataset consisting of 42,990 samples in total. It's worth noting the heterogeneity within the data, consisting of observations from numerous distinct organisms, as positive and negative points appear to be distributed with a certain uniformity across the projected space.

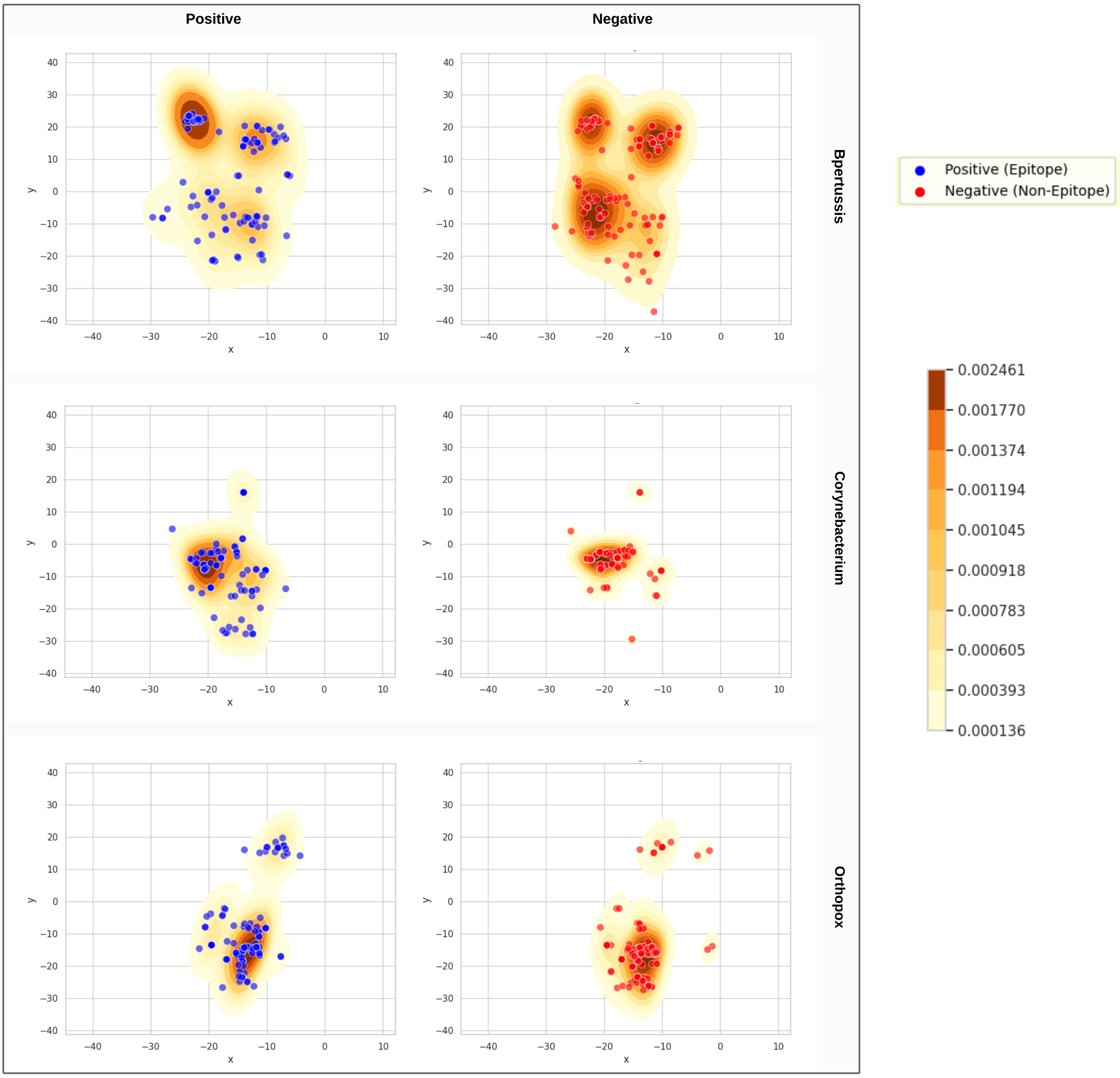


Figure 5.2: The t-SNE projection is presented, stratified by *B. pertussis*, *Corynebacterium and Orthopoxvirus*. Observe the well-defined clusters of high-density positive and negative observations, occupying distinct segments within the feature space. This visual representation illustrates the propensity for epitopes (positive observations) from various pathogens to consistently manifest in separate regions of the feature space. Importantly, regions with a high density of positive examples for one pathogen can also have a high density of negative examples for another pathogen. For instance, the portion around (-20, -5) of the negative *B. pertussis* examples overlaps a high-density region of positive *Corynebacterium* points in the same region. This type of data characteristic may make taxon-specific models better able to learn which regions of the feature space are more strongly associated with positive/negative examples for specific pathogens. Generalist models, on the other hand, are trained on data from multiple pathogens, which can make it more difficult for them to learn the specific signatures of each pathogen.

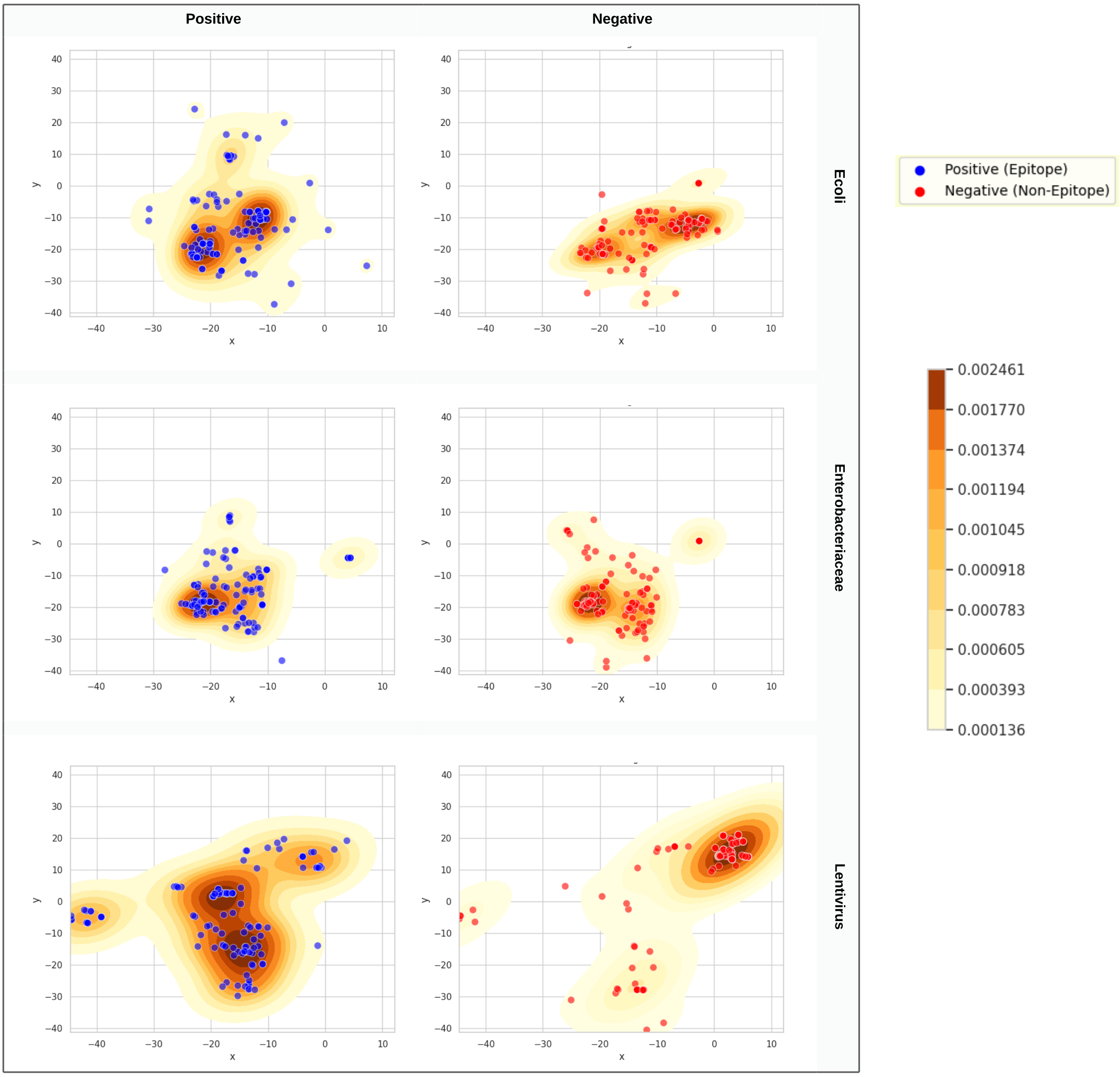


Figure 5.3: t-SNE-projected data from *E. coli, Enterobacteriaceae and Lentivirus.* As another example of why taxon-specific modelling may be preferrable, the negative examples of *Enterobacteriaceae* in the region of (-22, -20) align with a high-density cluster of positive *E. coli* data points in the same region.

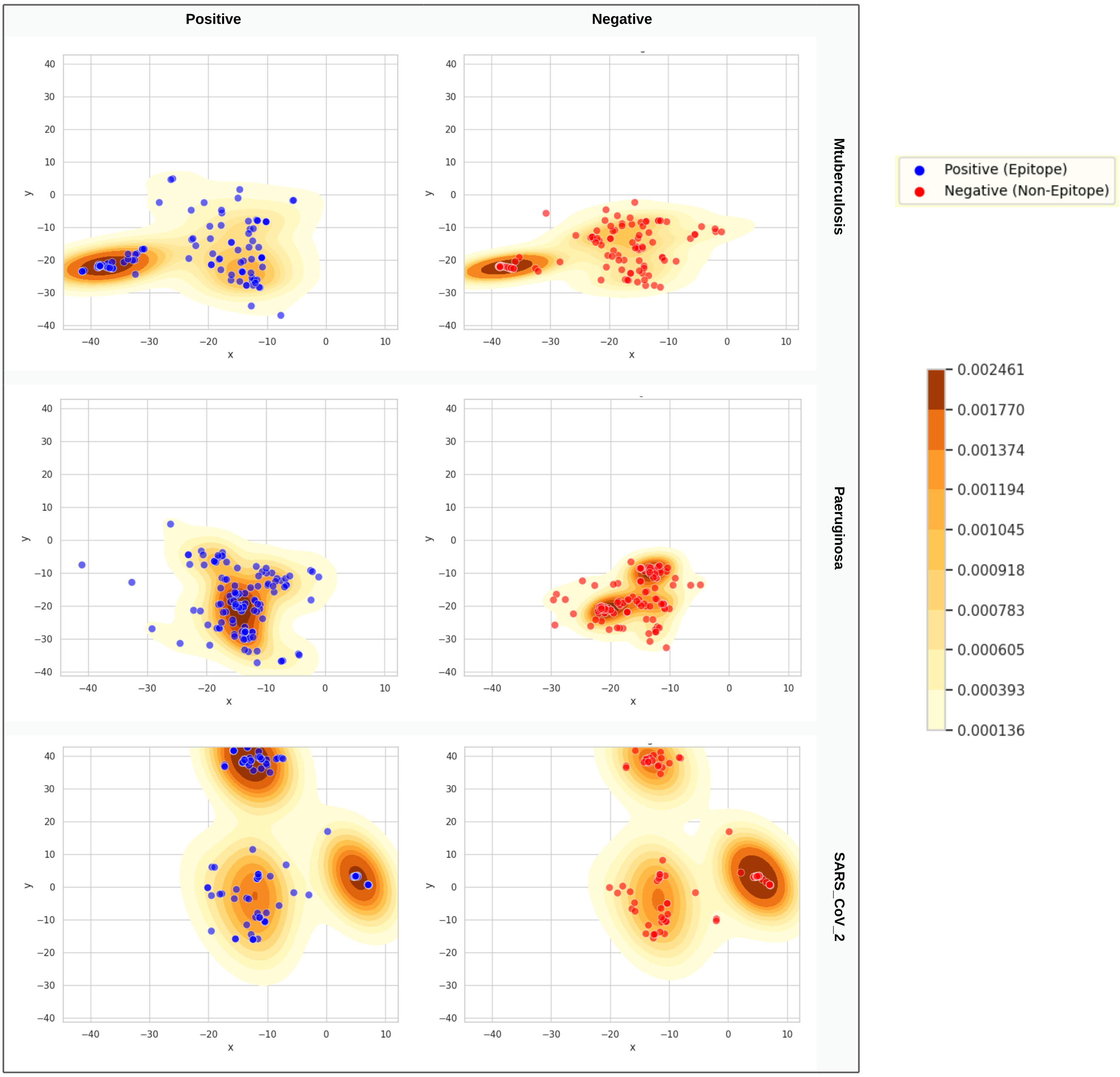


Figure 5.4: t-SNE-projected data from *M. tuberculosis, P. aeruginosa and SARS-Cov-2.* In the projection, the negative examples of *M. tuberculosis*, portion (-15, -20), align with a high-density cluster of positive *P. aeruginosa* data points in the same region. Additionally, note that the negative *P. aeruginosa* samples within the range (-15, -20) roughly coincide with a populated cluster of positive *M. tuberculosis* data points within the same region.

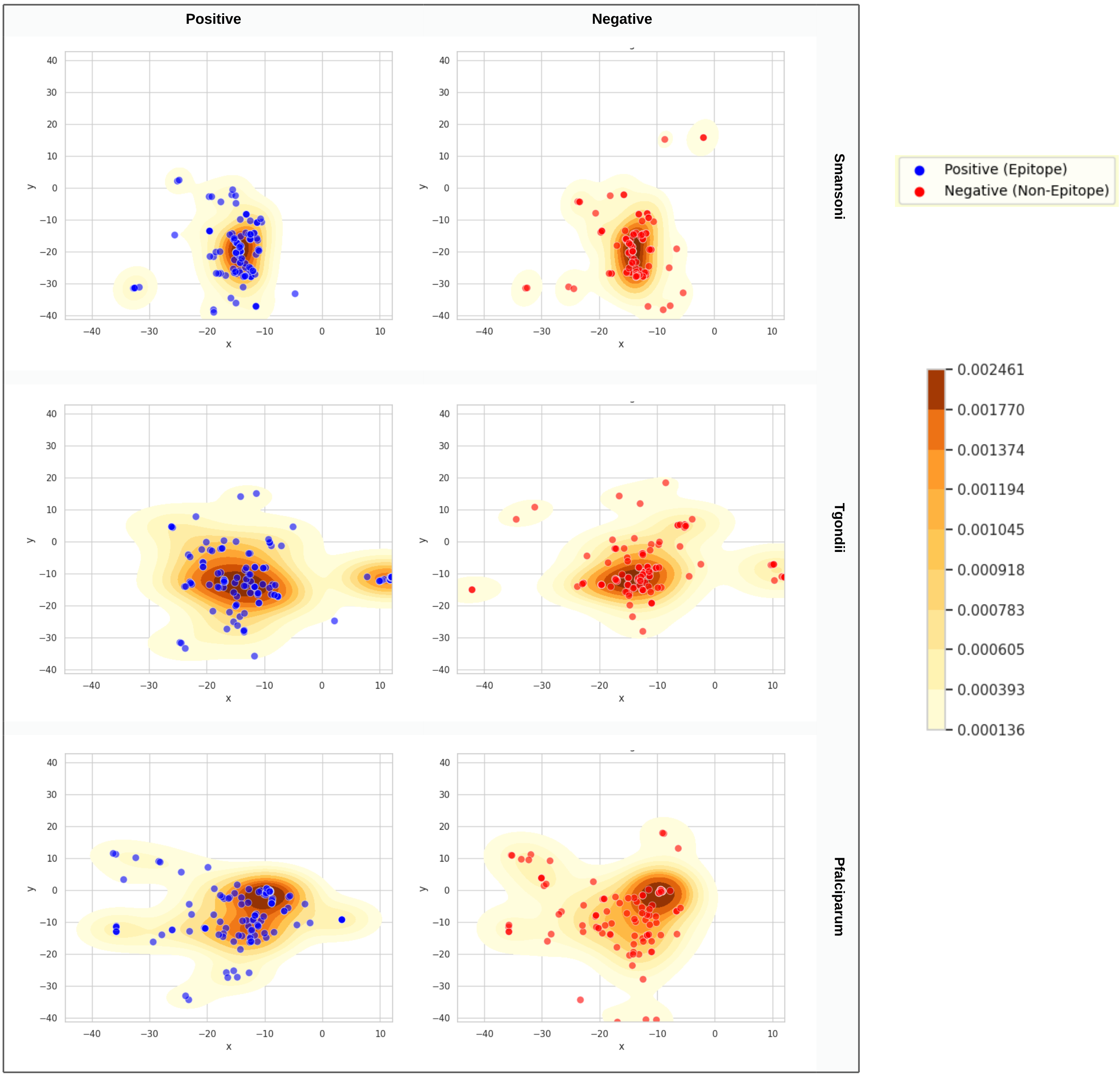


Figure 5.5: t-SNE-projected data from *S. mansoni, T. gondii and P. falciparum.* In this projection, the negative examples of *S. mansoni*, portion (-15, -15), align with a high-density cluster of positive *T. gondii* data points in the same area.

Although each figure presents only qualitative comparisons for three pathogens, this analysis, based on consistent coordinates, enables clearer identification of overlapping positive and negative regions among different pathogen groups. This evaluation provides support for the adoption of taxon-specific models.

5.2 Single Domain Adaptation Modeling

Before formalizing the hierarchical domain adaptation method, a preliminary framework is established using a single domain adaptation approach, referred to as *EpitopeTransfer* in this study. This initial work focused on evaluating whether meaningful information could be transferred from higher-level to lower-level taxa. To achieve this, a higher hierarchical level with a substantial amount of data was selected as the single source domain for these experiments. The framework for this analysis consisted of four main components (see Figure 5.6) and involved modeling tasks as follows:

1. *Data extraction and preparation*: labelled peptides (i.e., peptides known to be either epitope-containing regions or non-epitopes) are extracted from the IEDB database, and their corresponding source proteins are retrieved either from the NCBI Protein database or UniprotKB, based on the protein ID provided in the IEDB entry. Protein sequences are then clustered based on normalized alignment scores, using single-linkage hierarchical clustering. A similarity threshold of 30% is applied, such that only sequences sharing at least 30% similarity are grouped into the same cluster. The resulting clusters serve as allocation units for splitting data during model development, training, and testing.

2. *Embedder development*: fine tuning of the general-purpose ESM protein language model to the task of LBCE prediction, as detailed in the Appendix A. The ESM model is re-trained using the entries from the higher-level dataset, resulting in a protein embedder fine tuned for that particular phylogenetic group. Through this fine tuning step, the model structure learned as part of the ESM model's primary development [Rives et al., 2021] is augmented with knowledge about the representation of LBCEs from that particular higher level taxon. In this work, for single domain adaptation experiments, all experiments were first conducted using the 650-million-parameter version of ESM-1b as the base model, and the entire procedure was then repeated using ESM-2 as the base model; however, adapting the framework to employ more recent or larger embedder versions is a straightforward task.

3. *Feature calculation*: the fine-tuned embedder is deployed to extract features for the sequences from the target (lower level) taxon. In the feature generation process, the full lower-level protein sequences are fed to the tuned embedder rather than just the labelled peptides, which enables the model to capture richer contextual information. This results in an enhanced feature representation for each residue position. The output of this feature calculation model is then reduced so that only

the labeled peptide regions are selected for later classifier training and performance assessment.

4. *Predictive model training*: finally, the data generated in the previous step are used to fit and optimize the hyperparameters of a Random Forest classifier [Breiman, 2001], resulting in a bespoke LBCE prediction model for the target taxon.

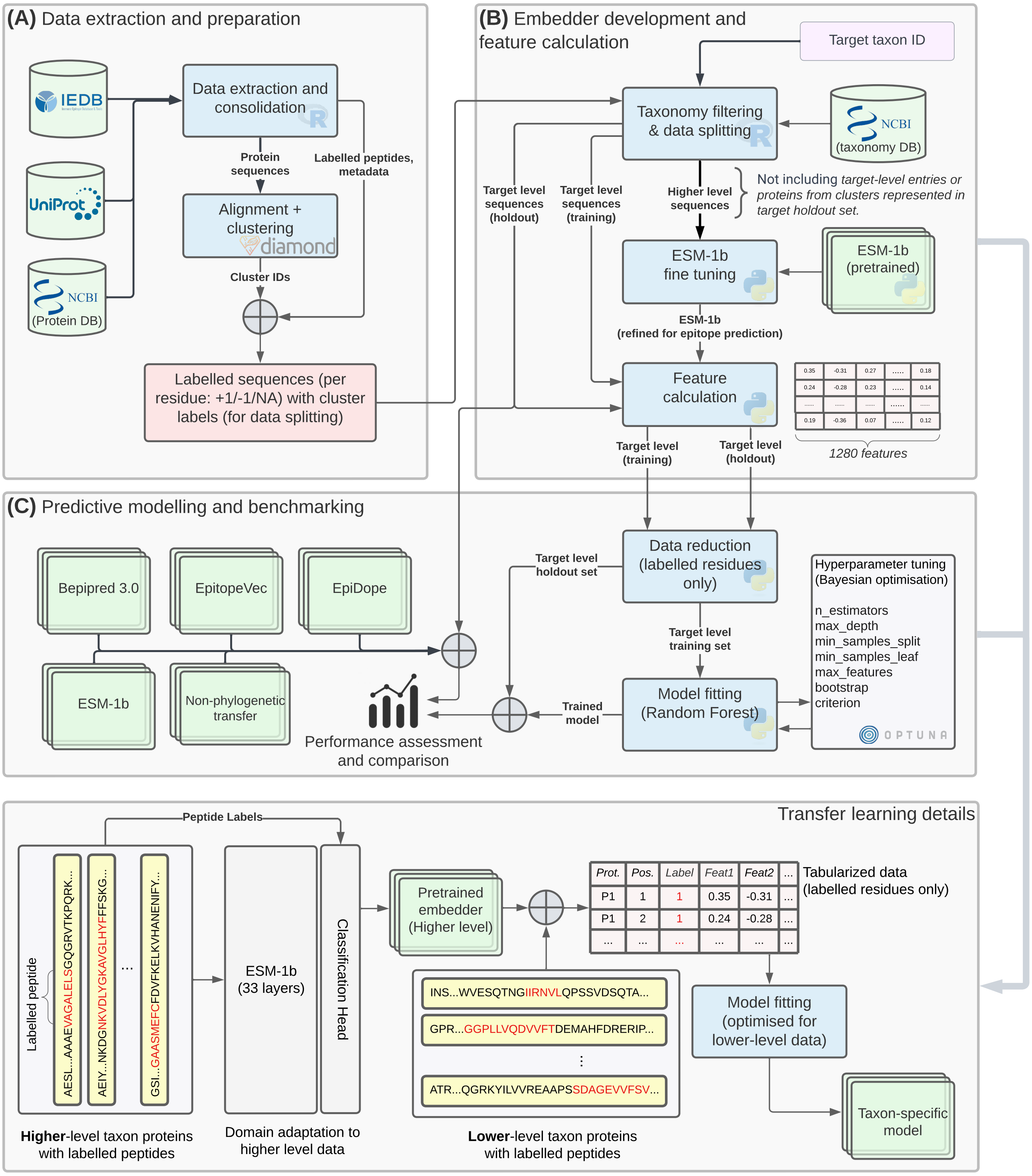

(A) Data extraction and preparation
IEDB
UniProt
NCBI
(Protein DB)
Data extraction and consolidation
Protein sequences
Labelled peptides, metadata
Alignment + clustering
diamond
Cluster IDs
Labelled sequences (per residue: +1/-1/NA) with cluster labels (for data splitting)
(B) Embedder development and feature calculation
Target taxon ID
Taxonomy filtering & data splitting
NCBI
(taxonomy DB)
Target level sequences (holdout)
Target level sequences (training)
Higher level sequences
Not including target-level entries or proteins from clusters represented in target holdout set.
ESM-1b fine tuning
ESM-1b (pretrained)
ESM-1b (refined for epitope prediction)
Feature calculation
0.35
-0.31
0.27
.....
0.18
0.24
-0.28
0.23
.....
0.14
0.19
-0.36
0.07
.....
0.12
1280 features
Target level (training)
Target level (holdout)
(C) Predictive modelling and benchmarking
Bepipred 3.0
EpitopeVec
EpiDope
ESM-1b
Non-phylogenetic transfer
Performance assessment and comparison
Target level holdout set
Data reduction (labelled residues only)
Target level training set
Model fitting (Random Forest)
Trained model
Hyperparameter tuning (Bayesian optimisation)
n_estimators
max_depth
min_samples_split
min_samples_leaf
max_features
bootstrap
criterion
OPTUNA
Transfer learning details
Peptide Labels
Labelled peptide
AESL...AAAEVAGALELSGQGRVTKPQRK...
AEIY...NKDGNKVDLYGKAVGLHYFFFSKG...
...
GSI...GAASMEFCFDVFKELKVHANENIFY...
ESM-1b (33 layers)
Classification Head
Higher-level taxon proteins with labelled peptides
Domain adaptation to higher level data
Pretrained embedder (Higher level)
INS...WVESQTNGIIRNVLQPSSVDSQTA...
GPR...GGPLLVQDVVFTDEMAHFDRERIP...
⋮
ATR...QGRKYILVVREAAPSSDAGEVVFSV...
Lower-level taxon proteins with labelled peptides
Prot.
Pos.
Label
Feat1
Feat2
...
P1
1
1
0.35
-0.31
...
P1
2
1
0.24
-0.28
...
...
Tabularized data (labelled residues only)
Model fitting (optimised for lower-level data)
Taxon-specific model

Figure 5.6: Overview of the EpitopeTransfer framework for building taxon-specific Linear B-Cell Epitope predictors. Figure extracted from the article "EpitopeTransfer: a Phylogeny-aware transfer learning framework for taxon-specific linear B-cell epitope prediction".

5.3 Hierarchical Domain Adaptation Modeling

This section extends the single domain adaptation methods described previously to a hierarchical domain adaptation (HDA) approach. Unlike the earlier strategy that considers only a single source domain, this generalization incorporates information from all higher-level domains. Central to this extension is the introduction of a *Hierarchical Weighted Cross-Entropy Loss* designed to dynamically adjust the influence of higher-level data while simultaneously balancing exposure to positive and negative samples. By tailoring the importance of each level, the method aims to mitigate negative transfer and address class imbalance within hierarchical structures. In doing so, it fills critical gaps in existing HDA approaches, offering a more flexible and adaptable methodology. To support the formalization of the problem in the **Total Cost subsection**, the following key terms are defined:

1. **Data Domains**

   - Source Domain (D_S): Includes data from varying levels of the hierarchy, represented by:

     – X_S: Feature space of the source domain.

     – $P(X_S)$: Marginal probability distribution of the features in X_S.

     – Y_S: Labels in the source domain.

     – H_{indexS}: Hierarchical index of the source domain, is defined within the interval: $H_{indexS} \in [0, n]$.

     Formalized as $D_S = \{X_S, P(X_S), Y_S, H_{indexS}\}$.

   - Target Domain (D_T): Consists of data from a level of the hierarchy, including:

     – X_T: Feature space of the target domain.

     – $P(X_T)$: Marginal probability distribution in the target domain.

     – Y_T: Labels in the target domain for supervised tasks.

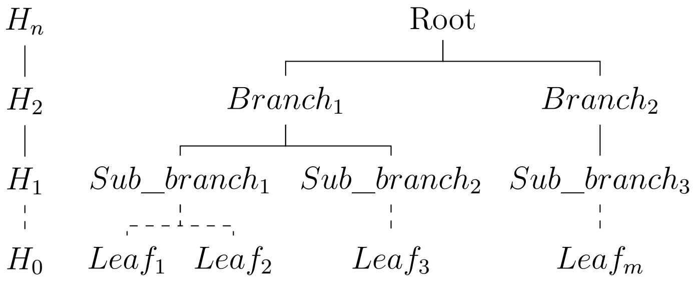
     – H_{indexT}: Hierarchical index of the target domain, is defined within the interval: $H_{indexT} \in [0, n]$.

     Formalized as $D_T = \{X_T, P(X_T), Y_T, H_{indexT}\}$.

2. **Mapping Function:** The mapping function, denoted as f_{map}, is defined to translate the raw data from both the source (D_S) and target (D_T) domains into a computational representation.

3. **Hierarchical Proximity:** The hierarchical index (H_{index}) quantifies the level of each domain within the overall data hierarchy. This index is determined based on

the hierarchical organization of the data, with higher values representing higher levels in the hierarchy and vice versa. The proximity between D_S and D_T in the hierarchy is determined by the difference in their hierarchical indices, $\Delta H = |H_{indexS} - H_{indexT}|$, where $H_{indexS} > H_{indexT}$. A smaller value of ΔH indicates closer proximity in the hierarchy.



This figure illustrates a general hierarchical data structure. As an example, $Branch_1$ might represent the Source Domain (D_S), while $Leaf_3$, within Sub_branch_2, could exemplify the Target Domain (D_T). To prevent data leakage, data related to $Leaf_3$ is excluded from $Branch_1$.

4. **Hierarchical Weights:** Hierarchical weights (W_H) are level-specific weights assigned to each hierarchical level during the training process. These weights are used to define the contribution of each hierarchical level to the total loss. By assigning higher weights to specific levels, the model can emphasize the importance of certain levels during optimization. The hierarchical weight set is represented as:

$$W_H = \{w_{h_0}, w_{h_1}, \ldots, w_{h_n}\},$$

where $w_{h_0}, w_{h_1}, \ldots, w_{h_n}$ are the weights assigned to the hierarchical levels h_0, h_1, $\ldots, h_n$, respectively.

5. **Weighted Cross-Entropy Loss:** The *Weighted Cross-Entropy Loss* is formulated to manage hierarchical data and address class imbalance at each level of the hierarchy.

    - The hierarchical weight w_0 is fixed at 1.0, while $w_l \in [0, 1]$ for all levels $l \in \{1, \ldots, n\}$.
    - The weights are normalized such that their sum across all levels is given by $\sum_{j=0}^{n} w_j$.

- Class imbalance at each level l is addressed using a class-weight vector α_l, computed based on inverse frequency for each class. The vector is then normalized to ensure that $\sum_{\text{classes}} \alpha_{l,c} = 1$.

The loss at a specific level l is defined as:

$$\mathcal{L}_l = \frac{w_l}{\sum_{j=0}^{n} w_j} \cdot \mathrm{CE}\Big(\hat{y}_l, y_l; \alpha_l\Big),$$

where:

- w_l represents the hierarchical weight for level l,
- $\hat{y}_l$ denotes the predicted logits for level l,
- y_l corresponds to the ground-truth labels for level l, and
- $\mathrm{CE}(\cdot)$ is the Cross-Entropy Loss function that incorporates class weights.

Total Cost

The total cost, denoted as Weighted Cross Entropy Loss (WCEL), is defined as follows:

$$\mathrm{WCEL} = \min_{\theta} \Bigg[\underbrace{\sum_{(x_s,\, y_s) \,\in\, (X_S,\, Y_S)} \sum_{l=0}^{n} \frac{w_l}{\sum_{j=0}^{n} w_j} \, \mathrm{CE}\Big(f_{\theta,l}\Big(f_{\mathrm{map}}(x_s,\ H_{indexS})\Big),\ y_s;\ \alpha_l\Big)}_{\text{Source-Domain Loss}} + \underbrace{\sum_{(x_t,\, y_t) \,\in\, (X_T,\, Y_T)} \sum_{l=0}^{n} \mathrm{CE}\Big(f_{\theta,l}\Big(f_{\mathrm{map}}(x_t,\ H_{indexT})\Big),\ y_t;\ \alpha_l\Big)}_{\text{Target-Domain Loss}} \Bigg].$$

1. **Source-Domain Loss:**

$$\sum_{(x_s,\, y_s) \,\in\, (X_S,\, Y_S)} \sum_{l=0}^{n} \frac{w_l}{\sum_{j=0}^{n} w_j} \, \mathrm{CE}\Big(f_{\theta,l}\Big(f_{\mathrm{map}}(x_s,\ H_{indexS})\Big),\ y_s;\ \alpha_l\Big)$$

   For the source domain (X_S, Y_S), a separate Cross-Entropy Loss (CE) is calculated for each hierarchical level $l \in \{0, 1, \ldots, n\}$. The total source-domain loss is a weighted average of these losses, where the hierarchical weights (w_l) determine the relative importance of each level. The class weights (α_l) are included in the Cross-Entropy Loss to address class imbalance within each level.

2. **Target-Domain Loss:**

$$\sum_{(x_t,\, y_t) \,\in\, (X_T,\, Y_T)} \sum_{l=0}^{n} \mathrm{CE}\Big(f_{\theta,l}\Big(f_{\mathrm{map}}(x_t,\ H_{indexT})\Big),\ y_t;\ \alpha_l\Big)$$

In the target domain (X_T, Y_T), a separate Cross-Entropy Loss is also calculated for each hierarchical level l. The hierarchical weight w_l is fixed at 1.0 in the target domain to reflect its critical importance during training. Class weights (α_l) are used to mitigate class imbalance at this level.

3. **Optimization Parameters (θ):** The parameters θ represent the trainable components of the neural language model, such as weights and biases across all layers. The objective is to optimize θ to minimize the total weighted loss across all hierarchical levels for both the source and target domains.

5.4 Conclusion

This chapter introduced a novel HDA method designed to address challenges posed by data arranged across multiple hierarchical levels. In particular, the method extends single domain adaptation approaches by incorporating level-specific weights (W_H), and class-weight vectors (α_l) that jointly manage the variable importance of each level and class imbalance in the hierarchy. By formalizing the Weighted Cross-Entropy Loss with separate source- and target-domain components, the method provides a flexible mechanism for balancing the influence of both higher and target domains during training. The decoupling of domain losses ensures that each hierarchical level contributes proportionally to the training process, minimizing negative transfer and noise, and preserving the structure of the hierarchy during joint optimization.

In contrast to single-domain adaptation techniques that often assume a flat domain structure, the proposed approach considers hierarchical proximity, enabling more fine-grained control over adaptation when the target domain has limited labeled data. Altogether, these features make the proposed HDA strategy particularly well-suited for complex, multi-level NLP tasks, where source domains are organized hierarchically. To demonstrate the effectiveness of this approach, the method is applied to the task of epitope prediction in the following chapter.

Chapter 6

Results

6.1 Introduction

This chapter aims to evaluate the proposed hierarchical domain adaptation method in a case study. The chosen task is epitope prediction, which provides a suitable scenario due to its hierarchical organization of data within a phylogenetic structure. This inherent organization allows the exploration of hierarchical domain adaptation strategies that leverage relationships between taxonomic levels.

Initially, a Single Domain Adaptation (SDA) approach is applied to the epitope prediction task to establish a starting point. Following this, the solution is generalized through the proposed Hierarchical Domain Adaptation (HDA) method, which adjusts the relative importance of training examples based on the hierarchical structure. This process demonstrates the ability of the method to transfer knowledge from broader domains, such as higher taxonomic levels, to more specific domains, such as species or genus.

The chapter also provides a detailed description of the datasets, emphasizing their hierarchical and phylogenetic organization, along with the methods employed for data preparation and clustering. Additionally, the performance of the proposed method is rigorously evaluated through statistical significance analysis, comparing it to baseline methods across eight metrics. Finally, the results are discussed, highlighting their implications and contributions.

6.2 Experimental Setup

This section describes the experimental setup adopted to evaluate the proposed domain adaptation approaches. It includes details on data extraction and preparation, the architecture of the neural network models, and the evaluation metrics used for performance assessment.

6.2.1 Data extraction and preparation

Data extraction, filtering, and consolidation were performed using the *epitopes* R package [Campelo and Ashford, 2022]. Epitope data was retrieved from the complete XML export of the Immune Epitopes Database, IEDB [Vita et al., 2018]. All entries classified as LBCEs from organisms within the taxa Viruses (NCBI:txid10239), Bacteria (NCBI:txid2), and Eukaryota (NCBI:txi2759) were extracted from the IEDB export.

The proteins associated with each entry were retrieved from either the NCBI protein database [Coordinators, 2015] or UniprotKB [Consortium, 2022], based on the protein IDs provided in the metadata of each IEDB record. Overlapping peptides of the same class were merged into a single entry to prevent partial data duplication, while residues with conflicting labels were labelled as the mode of the labels for each residue, and removed in case of ties. Positive-labelled peptides of length greater than 30 were removed to prevent long "Epitope-containing regions" from adding excessive noise to the training data, and labelled peptides shorter than 5 residues were also treated as noise and removed from the datasets.

Entries were clustered based on source protein dissimilarity, calculated using DIAMOND [Buchfink et al., 2021], using agglomerative clustering with single linkage and a 30% similarity threshold. Clusters were considered as the basic splitting unit when isolating the final test sets, as well as for cross validation and hyperparameter tuning, to minimize data leakage due to similarity/homology.

To investigate the transfer learning approach proposed in this work, twenty pairs of datasets were instantiated for a diverse range of pathogens, including bacterial, viral, and eukaryotic pathogens. These datasets are detailed in Table 6.1.

Table 6.1: Twenty pairs of datasets for a diverse range of pathogens

Higher Level Taxon	**Peptides**	**Lower Level Taxon**	**Peptides**
Bacteria			
Pseudomonadota (phylum)	276- / 551+	B. pertussis (species)	34- / 61+
		P. aeruginosa (species)	12- / 12+
		E. coli (species)	22- / 94+
		Enterobacteriaceae (family)	46- / 153+
Terrabacteria (clade)	977- / 888+	M. tuberculosis (species)	267- / 322+
		Corynebacterium (genus)	12- / 13+
Bacillota (phylum)	538- / 424+	C. difficile (species)	43- / 31+
Chlamydia (genus)	299- / 293+	C. trachomatis (species)	79- / 144+
Virus			

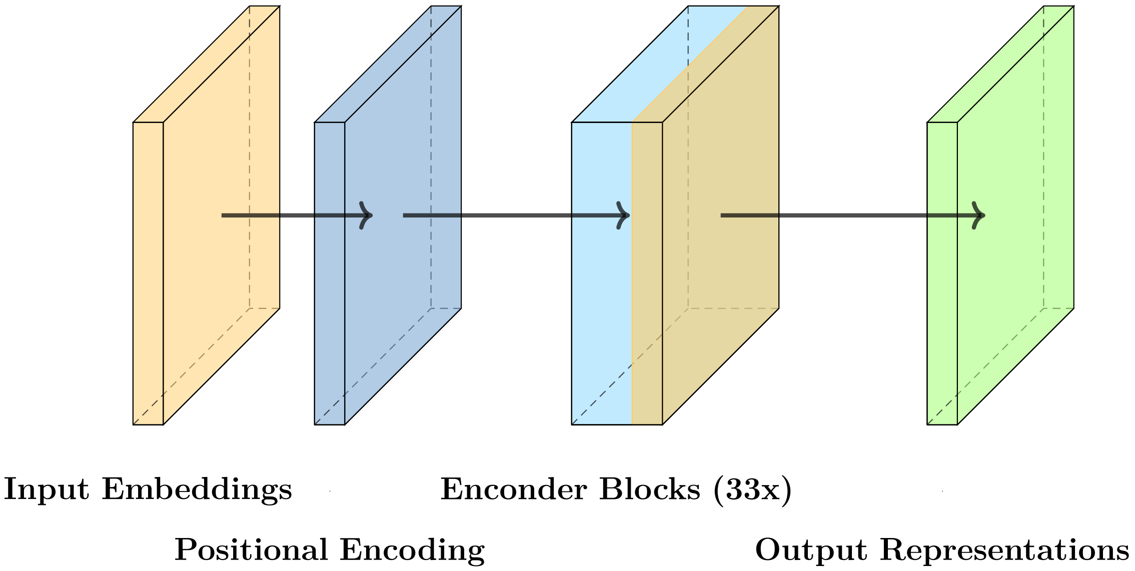
Higher Level Taxon	Peptides	Lower Level Taxon	Peptides
Bamfordvirae (kingdom)	39- / 134+	Orthopoxvirus (genus)	14- / 20+
Pararnavirae (kingdom)	188- / 410+	Lentivirus (genus)	12- / 99+
Orthornavirae (kingdom)	8356- / 4145+	SARS-CoV-2 (genus)	726- / 244+
Negarnaviricota (phylum)	426- / 594+	Influenza A (species)	89- / 246+
		Measles morbilivirus (species)	26- / 37+
		Filoviridae (family)	138- / 74+
		Mononegavirales (order)	240- / 260+
Duplodnaviria (realm)	716- / 770+	Human Gamma. 4 (species)	466- / 354+
Eukaryota			
Platyhelminthes (phylum)	147- / 132+	S. mansoni (species)	243- / 173+
Apicomplexa (phylum)	357- / 1184+	T. gondii (species)	60- / 82+
Sar (clade)	357- / 1186+	P. falciparum (species)	206- / 921+
Protostomia (clade)	837- / 722+	O. volvulus (species)	246- / 133+

6.2.2 Selected Neural Network Architecture

The neural network models adopted in this work were selected based on a balance between representation quality and computational efficiency. Although different architectures were used for SDA and HDA modeling, it is important to emphasize that the proposed modelling framework is architecture-agnostic. That is, it does not depend on the specific internal structure of any particular model, as long as it is capable of generating meaningful contextual embeddings from protein sequences. This flexibility ensures that the methodology remains adaptable to future developments in protein language modeling and applicable across different domains and tasks.

For SDA modeling, the model selected was ESM-1b [Rives et al., 2021], a large-scale protein language model based on a 33-layer encoder-only Transformer architecture, comprising 650 million parameters. This model was pretrained on 250 million protein sequences and produces 1280-dimensional contextual embeddings for each residue. Its performance, open-source nature, and ease of use make it suitable for SDA experiments designed to ensure reproducibility. Additionally, all experiments were also executed using ESM-2 [Lin et al., 2023], an evolution of ESM-1b that introduces improvements in architecture, training parameters, computational resources, and data. The ESM-2 model employed has the same architecture and size as ESM-1b, comprising 33 layers, 650 million parameters, and 1280-dimensional embeddings.

In contrast, for HDA modeling, only the lightest architecture from the ESM-2 family [Lin et al., 2023] was selected. This model is based on a 6-layer encoder-only Transformer architecture, comprises 8 million parameters, and generates residue-level embeddings with a dimensionality of 320. The choice of a smaller model is motivated by the multi-level nature of hierarchical adaptation, which involves fine-tuning across several domain levels. In this setting, a lightweight model enables faster training and evaluation, while maintaining satisfactory performance and ensuring better scalability in resource-limited or structurally complex scenarios.



The schematic representation of the ESM-1b architecture is shown above, illustrating its structural components. The model takes as input a sequence of amino acids, which is first converted into high-dimensional (1280-dimensional) embeddings. Positional encodings are then added to incorporate sequence order information. The resulting representations are passed through 33 Transformer encoder blocks, each composed of multi-head self-attention and position-wise feed-forward layers. These blocks iteratively refine the representations, capturing both local and long-range dependencies between amino acids. The final output layer generates contextualized 1280-dimensional embeddings for each residue, enabling a wide range of downstream tasks. Compared to ESM-1b, ESM-2 maintains the same architecture but replaces the absolute positional encoding with Rotary Position Embedding [Su et al., 2024], allowing for better extrapolation beyond the training context window. Additionally, dropout layers used in both the hidden and attention components of ESM-1b were completely removed in ESM-2, increasing the model's effective capacity.

6.2.3 Optimization Strategy

This section presents the optimization strategies adopted for both the SDA and HDA scenarios. In the SDA case, optimization focuses on tuning the hyperparameters of a Random Forest classifier, while in the HDA approach, it involves fine-tuning the parameters of a transformer-based protein language model. The following subsections detail the techniques, search spaces, and decisions used in each context.

Single Domain Adaptation

Hyperparameter optimization in the SDA setup was conducted using Bayesian optimization with the Optuna framework [Akiba et al., 2019]. The default sampler, Tree-structured Parzen Estimator (TPE) [Bergstra et al., 2011], was used due to its efficiency in exploring complex search spaces. The following hyperparameters of the Random Forest classifier were optimized:

- `n_estimators`: integer sampled from the interval $[100, 500]$. This parameter defines the number of decision trees in the ensemble.
- `max_depth`: categorical choice among `[None, 10, 20, 30]`. This parameter controls the maximum depth of each decision tree.
- `min_samples_split`: integer sampled from $[2, 10]$. This parameter specifies the minimum number of samples required to split an internal node.
- `min_samples_leaf`: integer sampled from $[1, 10]$. This parameter specifies the minimum number of samples required to form a leaf node.
- `max_features`: categorical choice among `['sqrt', 'log2', None]`. This parameter determines the number of features to consider when searching for the best split at each node.
- `bootstrap`: categorical choice between `True` and `False`. If set to `True`, the model uses bootstrap sampling when building trees.
- `criterion`: categorical choice between `'gini'` and `'entropy'`. This parameter defines the function used to measure the quality of a split.

The objective of the optimization process was to identify the set of hyperparameters that maximizes the overall Matthews Correlation Coefficient (MCC), computed as the average across all cross-validation folds on the validation set. The trial that achieved the highest MCC was selected as the final configuration. After selecting the optimal hyperparameters, a global decision threshold was calculated by aggregating the predictions across the validation folds and selecting the threshold that maximized the MCC. With the optimal hyperparameters and threshold selected, a final model was trained on all folds, excluding the held-out test fold. Predictions on the test set were then made using the previously determined global threshold.

Hierarchical Domain Adaptation

For the HDA scenario, the optimization strategy focused on fine-tuning the smallest variant of the ESM-2 model (6 layers and 8 million parameters) [Lin et al., 2023]. A key hyperparameter in this setting was the number of trainable layers.

To determine a suitable value for this parameter, a preliminary exploratory phase was carried out using three representative datasets: *B. pertussis*, *E. coli*, and *M. tuberculosis*. In this phase, the number of trials was fixed at five, and the optimization focused solely on identifying the optimal number of trainable layers. The process began with a single trainable layer, progressively unfreezing additional layers. The selection was guided by identifying a balance point between underfitting and overfitting. This procedure revealed that setting four trainable layers provided consistent behavior across all three taxa.

Based on the preliminary exploratory phase, the number of trainable layers was fixed at four and the number of optimization trials was limited to five for the 17 remaining taxa, as increasing trials tended to promote overfitting rather than improve generalization. This phase also revealed a broader challenge: while the number of trainable layers could be adequately defined, the optimal extent of optimization - particularly the number of trials - varies across taxa, highlighting the need for adaptive strategies to set this parameter more effectively in future work.

The following hyperparameters were optimized:

- `learning_rate`: log-uniformly sampled from the interval $[10^{-6}, 5 \cdot 10^{-4}]$. This parameter controls the step size in the weight update during training.
- `weight_decay`: sampled from $[10^{-6}, 10^{-3}]$. This coefficient is used for L2 regularization to prevent overfitting by penalizing large weights in the model.
- `dropout_rate`: sampled from $[0.0, 0.2]$. This parameter defines the dropout probability, which randomly disables units during training to improve generalization.
- `num_train_epochs`: integer sampled from the interval $[5, 10]$. This parameter defines the number of training epochs, i.e., the number of full passes through the training dataset.
- `level_weight_i`: for each level i in the hierarchical loss, a weight was assigned to balance its contribution to the total loss. The weight for level 0 was fixed at 1.0,

while weights for levels $i = 1 \ldots L$ were sampled from $[0.0, 1.0]$. These values adjust the influence of higher hierarchical levels during model training.

6.2.4 Evaluation Metrics

To assess the performance of the proposed method, a comprehensive set of evaluation metrics was employed. The metrics include Area Under the ROC Curve (AUC), F1 Score, Matthews Correlation Coefficient (MCC), Balanced Accuracy (BACC), Positive Predictive Value (PPV or Precision), Negative Predictive Value (NPV), Sensitivity (Recall), and Specificity. Each of these indicators offers complementary insights into different aspects of model performance, capturing the ability to correctly identify epitopes, minimize false positive predictions, and ensure reliable behavior under class imbalance.

Model selection and threshold optimization were guided by the MCC, which served as the primary performance metric. The classification threshold was defined as the value that maximized the MCC on the validation set.

A detailed description and mathematical formulation of all evaluation metrics are provided in Appendix A

6.3 Single Domain Adaptation Modeling

This section introduces the SDA strategy for LBCE prediction, initially applied to validate the effectiveness of phylogeny-based knowledge transfer. The proposed method, EpitopeTransfer, incorporates a phylogeny-aware fine-tuning at the higher level that leverages evolutionary relationships among pathogens to improve performance at the target level. The method is evaluated across a diverse set of taxa and compared against both internal baselines and state-of-the-art predictors. Two base models serve as the foundation for the experiments: ESM-1b and ESM-2.

6.3.1 Performance Results (ESM-1b)

The performance evaluation is organized into four parts: (1) the impact of phylogeny-aware transfer learning, (2) a comparison with state-of-the-art LBCE prediction methods, and (3) an analysis of specific target taxa that benefit the most from the SDA strategy.

1. Phylogeny-aware transfer learning

This first analysis of results investigated the impact of phylogeny-aware transfer learning on the performance of taxon-specific LBCE predictors. To achieve this, EpitopeTransfer was compared with baseline models that followed the same procedure but omitted the phylogeny-aware fine-tuning step (*ESM-1b baseline*).

Based on this comparison, EpitopeTransfer demonstrated significant performance gains over the *ESM-1b baseline*, as shown in the plot (Figure 6.1) and detailed in the first row of Table 6.2. Specifically, the median of the paired differences (*Medians of diff.*) was 0.082 (0.025, 0.143), with $p_{\text{adj}} = 0.01698$, indicating a statistically relevant improvement. This highlights the effectiveness of the phylogeny-aware fine-tuning step in enhancing the MCC metric, allowing EpitopeTransfer to better classify examples when compared to the baseline approach.

Table 6.2: Comparison Results for MCC. P-values correspond to Wilcoxon Signed-Ranks tests on the median of paired differences. Adjusted p-values were calculated using the Benjamini-Hochberg stepwise correction for *all vs. one* tests. Adjusted p-values under 0.05 correspond to statistically significant results at the (FDR-corrected) 95% confidence level.

Metric	Pair	Medians of diff.	p-value	FDR	Sign.
MCC	EpitopeTrans. vs BepiPred 3	0.204 (0.105, 0.390)	0.00823	0.01698	Yes
MCC	EpitopeTrans. vs EpiDope	0.134 (0.023, 0.254)	0.02041	0.02041	Yes
MCC	EpitopeTrans. vs EpitopeVec	0.150 (0.027, 0.296)	0.02041	0.02041	Yes
MCC	EpitopeTrans. vs ESM-1b	0.082 (0.025, 0.143)	0.00618	0.01698	Yes
MCC	EpitopeTrans. vs NPTransfer	0.065 (0.019, 0.154)	0.01019	0.01698	Yes

A core hypothesis in the development of EpitopeTransfer is that LBCE data from phylogenetically closer pathogens provides more relevant information for training taxon-specific predictors than data from distantly related organisms. This hypothesis was examined by comparing EpitopeTransfer with non-phylogenetic transfer learning baselines (NPTransfer). These baselines use the same methodology as EpitopeTransfer but perform embedder fine-tuning on epitopes exclusively from higher-level taxa that do not include the target taxon and are composed of distantly related pathogens (see Appendix B).

The results clearly indicate performance gains from incorporating a phylogeny-aware data filtering strategy for fine-tuning (see the second row of Table 6.2). A statistically significant improvement in MCC was observed for EpitopeTransfer compared to NPTransfer (Δmedians = 0.065; 95% CI: 0.019, 0.154; $p = 0.01019$; FDR = 0.01698). This improvement demonstrates the effectiveness of leveraging phylogenetic relationships to refine the feature embedder on data derived from pathogens that are phylogenetically closer to the target taxon. These results, along with the observed improvement over the ESM-1b baseline, show that EpitopeTransfers superior performance comes from fine-tuning the feature embedder using data related to pathogens phylogenetically close to the target taxon. This phylogeny-aware approach enables the model to capture evolutionary re-

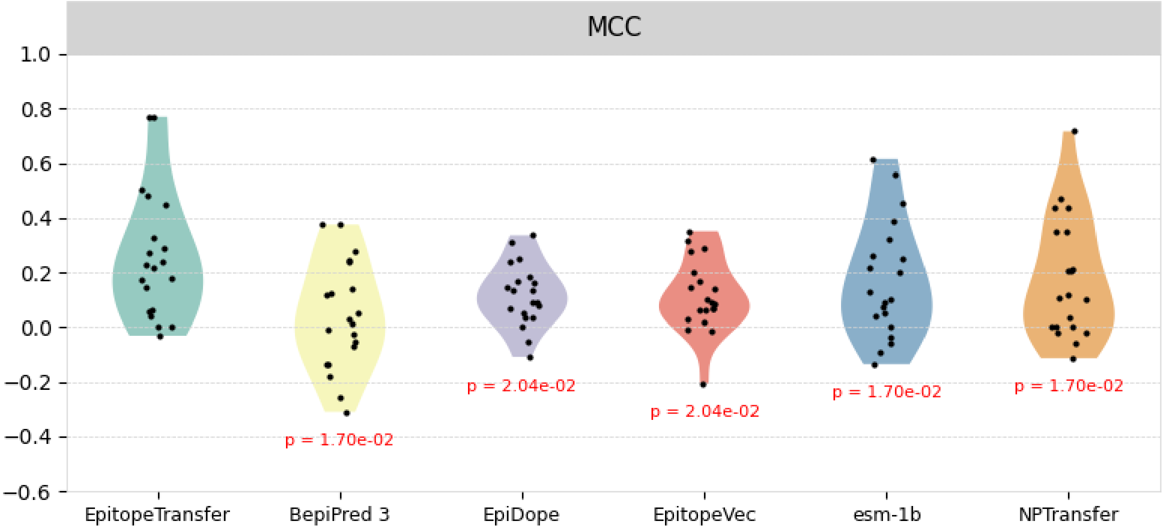


Figure 6.1: Comparison of methods in terms of MCC. The violin plots represent the distribution of MCC values for each method across multiple datasets. P-values correspond to Wilcoxon Signed-Ranks tests performed on the median of paired differences between EpitopeTransfer and each baseline. Adjusted p-values were calculated using the Benjamini-Hochberg stepwise correction for all vs. one comparisons. Adjusted p-values below 0.05 indicate statistically significant results at the (FDR-corrected) 95% confidence level.

lationships that appear to contribute to accurate predictions, going beyond the general refinement/fine-tuning of the protein language model for epitope prediction.

2. Comparison with state-of-the-art approaches

The performance of EpitopeTransfer was evaluated against three state-of-the-art LBCE prediction methods: BepiPred 3.0 [Clifford et al., 2022], EpiDope [Collatz et al., 2020], and EpitopeVec [Bahai et al., 2021]. These methods employ deep-learning techniques for epitope prediction but do not explicitly utilize taxon-specific or phylogeny-aware strategies. The objective of these comparisons was to assess whether the components of the EpitopeTransfer modeling pipeline are sufficient to produce taxon-specific predictors with performance comparable to or exceeding the leading prediction methods.

Table 6.2 presents the results, indicating statistically significant differences in median MCC scores between EpitopeTransfer and the external baselines. EpitopeTransfer demonstrated significant improvements over BepiPred 3.0 (Δmedians = 0.204; 95% CI: 0.105, 0.390; $p = 0.00823$; FDR = 0.01698), EpiDope (Δmedians = 0.134; 95% CI: 0.023, 0.254; $p = 0.02041$; FDR = 0.02041), and EpitopeVec (Δmedians = 0.150; 95% CI: 0.027, 0.296; $p = 0.02041$; FDR = 0.02041). These effect sizes were larger than those observed in comparisons against ESM-1b and NPTransfer. Further details, including all eight metrics, are available in Appendix D.

As presented in Table 6.4, which summarizes the average performance of methods across 20 datasets, EpitopeTransfer achieves an average MCC of 0.258, outperforming all three generalist LBCE predictors. Among the external baselines used, EpiDope obtains the highest average MCC (0.118), while BepiPred 3.0 and EpitopeVec reach 0.041 and 0.112, respectively. These results indicate that the phylogeny-aware fine-tuning strategy of EpitopeTransfer leads to more robust and accurate taxon-specific predictor.

3. EpitopeTransfer provides high-performance LBCE predictors for some pathogens

Some of the EpitopeTransfer models constructed in this study exhibit particularly high predictive performance, making them attractive for researchers interested in specific pathogens. One prominent example is the model for the Filoviridae family, which includes highly virulent viruses such as Ebola and Marburg [Kuhn et al., 2019]. According to Table 6.3, this model achieves an MCC of 0.766, representing substantial improvements over BepiPred 3.0 (0.143), EpiDope (0.240), and EpitopeVec (0.202). This large margin suggests that EpitopeTransfer is a robust LBCE predictor for pathogens within this family.

Other EpitopeTransfer models also demonstrate notably strong performance. For instance, the *Plasmodium falciparum* model achieves an MCC of 0.505, showing improvements over BepiPred 3.0 (0.119), EpiDope (0.088), and EpitopeVec (0.018). The model developed for genus Lentivirus, which covers HIV, exhibits an MCC of 0.770, outperforming BepiPred 3.0 (0.376), EpiDope (0.033), EpitopeVec (0.067). Similarly, the model trained for the Enterobacteriaceae family achieves an MCC of 0.479, surpassing BepiPred 3.0 (0.054), EpiDope (0.144), and EpitopeVec(0.061).

On the other hand, EpitopeTransfer did not yield strong results for some pathogens. For *Sars-cov-2*, its MCC of 0.043 is lower than the 0.169 achieved by EpiDope, making EpiDope the better model for this virus. Likewise, for *M. tuberculosis*, EpitopeTransfer obtained an MCC of -0.031, whereas ESM-1b emerged as the strongest approach with an MCC of 0.039. In these cases, the EpitopeTransfer models are unlikely to provide improvements over existing predictors. Nevertheless, substantial MCC improvements were observed in most EpitopeTransfer models, with only a few exceptions showing limited gains. Complete results for the remaining seven metrics are provided in Appendices C.

Table 6.3: Comparison of methods for MCC

Method	Dataset	Value
EpitopeTransfer	B. pertussis	0.000
BepiPred 3.0	B. pertussis	-0.008
EpiDope	B. pertussis	-0.108
EpitopeVec	B. pertussis	0.351
ESM-1b	B. pertussis	-0.090
NPTransfer	B. pertussis	0.000
EpitopeTransfer	C. difficile	0.173
BepiPred 3.0	C. difficile	-0.027
EpiDope	C. difficile	0.000
EpitopeVec	C. difficile	0.279
ESM-1b	C. difficile	0.052
NPTransfer	C. difficile	0.000
EpitopeTransfer	Corynebacterium	0.492
BepiPred 3.0	Corynebacterium	0.247
EpiDope	Corynebacterium	0.309
EpitopeVec	Corynebacterium	0.315
ESM-1b	Corynebacterium	0.294
NPTransfer	Corynebacterium	0.000
EpitopeTransfer	C. trachomatis	0.447
BepiPred 3.0	C. trachomatis	-0.055
EpiDope	C. trachomatis	0.137
EpitopeVec	C. trachomatis	0.286
ESM-1b	C. trachomatis	0.385
NPTransfer	C. trachomatis	0.440
EpitopeTransfer	E. coli	0.325
BepiPred 3.0	E. coli	-0.136
EpiDope	E. coli	0.248
EpitopeVec	E. coli	0.031
ESM-1b	E. coli	0.217
NPTransfer	E. coli	0.349
EpitopeTransfer	Enterobacteriaceae	0.479
BepiPred 3.0	Enterobacteriaceae	0.054

Method	Dataset	Value
EpiDope	Enterobacteriaceae	0.144
EpitopeVec	Enterobacteriaceae	0.061
ESM-1b	Enterobacteriaceae	-0.035
NPTransfer	Enterobacteriaceae	-0.113
EpitopeTransfer	Filoviridae	0.766
BepiPred 3.0	Filoviridae	0.143
EpiDope	Filoviridae	0.240
EpitopeVec	Filoviridae	0.202
ESM-1b	Filoviridae	0.558
NPTransfer	Filoviridae	0.716
EpitopeTransfer	Human gammaherpesvirus 4	0.241
BepiPred 3.0	Human gammaherpesvirus 4	-0.180
EpiDope	Human gammaherpesvirus 4	0.068
EpitopeVec	Human gammaherpesvirus 4	0.070
ESM-1b	Human gammaherpesvirus 4	0.249
NPTransfer	Human gammaherpesvirus 4	0.209
EpitopeTransfer	Influenza A	0.176
BepiPred 3.0	Influenza A	0.239
EpiDope	Influenza A	0.054
EpitopeVec	Influenza A	0.139
ESM-1b	Influenza A	-0.058
NPTransfer	Influenza A	0.099
EpitopeTransfer	Lentivirus	0.770
BepiPred 3.0	Lentivirus	0.376
EpiDope	Lentivirus	0.033
EpitopeVec	Lentivirus	0.067
ESM-1b	Lentivirus	0.615
NPTransfer	Lentivirus	0.469
EpitopeTransfer	M. tuberculosis	-0.031
BepiPred 3.0	M. tuberculosis	0.029
EpiDope	M. tuberculosis	0.033
EpitopeVec	M. tuberculosis	-0.008
ESM-1b	M. tuberculosis	0.039

Method	Dataset	Value
NPTransfer	M. tuberculosis	-0.056
EpitopeTransfer	Measles morbilivirus	0.386
BepiPred 3.0	Measles morbilivirus	-0.310
EpiDope	Measles morbilivirus	0.079
EpitopeVec	Measles morbilivirus	0.091
ESM-1b	Measles morbilivirus	0.160
NPTransfer	Measles morbilivirus	0.194
EpitopeTransfer	Mononegavirales	0.286
BepiPred 3.0	Mononegavirales	-0.136
EpiDope	Mononegavirales	0.336
EpitopeVec	Mononegavirales	0.170
ESM-1b	Mononegavirales	0.321
NPTransfer	Mononegavirales	0.209
EpitopeTransfer	Orthopox	0.366
BepiPred 3.0	Orthopox	0.375
EpiDope	Orthopox	0.163
EpitopeVec	Orthopox	-0.206
ESM-1b	Orthopox	0.246
NPTransfer	Orthopox	0.181
EpitopeTransfer	Ovolvulus	0.272
BepiPred 3.0	Ovolvulus	0.277
EpiDope	Ovolvulus	-0.055
EpitopeVec	Ovolvulus	0.064
ESM-1b	Ovolvulus	0.202
NPTransfer	Ovolvulus	0.210
EpitopeTransfer	P. aeruginosa	0.147
BepiPred 3.0	P. aeruginosa	-0.258
EpiDope	P. aeruginosa	0.137
EpitopeVec	P. aeruginosa	0.145
ESM-1b	P. aeruginosa	0.263
NPTransfer	P. aeruginosa	0.350
EpitopeTransfer	P. falciparum	0.505
BepiPred 3.0	P. falciparum	0.119

Method	Dataset	Value
EpiDope	P. falciparum	0.088
EpitopeVec	P. falciparum	0.018
ESM-1b	P. falciparum	0.453
NPTransfer	P. falciparum	0.437
EpitopeTransfer	S. mansoni	0.056
BepiPred 3.0	S. mansoni	0.126
EpiDope	S. mansoni	0.185
EpitopeVec	S. mansoni	-0.016
ESM-1b	S. mansoni	0.093
NPTransfer	S. mansoni	0.120
EpitopeTransfer	Sars-cov-2	0.043
BepiPred 3.0	Sars-cov-2	0.011
EpiDope	Sars-cov-2	0.169
EpitopeVec	Sars-cov-2	0.101
ESM-1b	Sars-cov-2	0.072
NPTransfer	Sars-cov-2	0.037
EpitopeTransfer	T. gondii	0.218
BepiPred 3.0	T. gondii	-0.070
EpiDope	T. gondii	0.092
EpitopeVec	T. gondii	0.084
ESM-1b	T. gondii	0.128
NPTransfer	T. gondii	0.108

Metric	EpitopeTrans	BepiPred 3.0	EpiDope	EpitopeVec	ESM-1b	NPTransfer
AUC	0.690 (±0.029)	0.503 (±0.035)	0.634 (±0.032)	0.602 (±0.027)	0.656 (±0.030)	0.642 (±0.032)
F1	0.592 (±0.060)	0.363 (±0.045)	0.276 (±0.029)	0.509 (±0.044)	0.542 (±0.060)	0.529 (±0.061)
MCC	0.258 (±0.052)	0.041 (±0.044)	0.118 (±0.025)	0.112 (±0.029)	0.172 (±0.047)	0.177 (±0.049)
B. ACC	0.623 (±0.028)	0.527 (±0.021)	0.548 (±0.011)	0.566 (±0.019)	0.581 (±0.023)	0.585 (±0.025)
PPV	0.549 (±0.056)	0.462 (±0.066)	0.581 (±0.065)	0.496 (±0.055)	0.529 (±0.058)	0.522 (±0.062)
NPV	0.724 (±0.057)	0.555 (±0.057)	0.571 (±0.054)	0.604 (±0.050)	0.638 (±0.060)	0.584 (±0.066)
Sensit.	0.697 (±0.068)	0.393 (±0.062)	0.226 (±0.037)	0.610 (±0.037)	0.641 (±0.073)	0.656 (±0.073)
Specif.	0.549 (±0.072)	0.660 (±0.061)	0.869 (±0.030)	0.522 (±0.023)	0.521 (±0.083)	0.513 (±0.079)

Table 6.4: Summary of average test set performance (*mean ±standard error*) for EpitopeTransfer (proposed method) and five baseline methods across 20 selected datasets. Each row corresponds to a performance evaluation metric, and the values indicate the mean performance of each method over all datasets.

6.3.2 Performance Results (ESM-2)

All experiments described in the previous subsection for ESM-1b were repeated using the ESM-2 as the base model, and the results are presented in Appendices E and F. However, despite the architectural and training improvements introduced in ESM-2, a significant performance gain was observed only in specificity, as illustrated in Figure 6.5. In contrast, ESM-1b demonstrated statistically significant superiority in both F1 score and sensitivity. All other metrics showed no significant improvement.

Metric	Pair	Medians of diff.	p-value	Significant
AUC	ET_ESM2 vs ET_ESM1b	-0.003 (-0.029, 0.024)	0.86949	No
BACC	ET_ESM2 vs ET_ESM1b	0.007 (-0.033, 0.039)	0.70118	No
F1	ET_ESM2 vs ET_ESM1b	-0.051 (-0.107, -0.004)	0.03277	Yes
MCC	ET_ESM2 vs ET_ESM1b	0.023 (-0.046, 0.080)	0.57060	No
NPV	ET_ESM2 vs ET_ESM1b	0.006 (-0.026, 0.069)	0.84082	No
PPV	ET_ESM2 vs ET_ESM1b	0.008 (-0.034, 0.044)	0.64766	No
Sensitivity	ET_ESM2 vs ET_ESM1b	-0.109 (-0.224, -0.014)	0.02395	Yes
Specificity	ET_ESM2 vs ET_ESM1b	0.084 (0.019, 0.264)	0.01718	Yes

Table 6.5: Summary of paired statistical comparisons for all performance metrics between EpitopeTransfer (ET) models utilizing ESM-2 and ESM-1b as base models. Statistically significant differences ($p < 0.05$) are indicated in the “Significant” column.

6.4 Hierarchical Domain Adaptation Modeling

This section presents the results obtained for the hierarchical generalization of the SDA strategy described previously. While the SDA approach focuses on knowledge transfer from a single phylogenetically related source domain, the HDA modeling extends this paradigm by incorporating data from distinct taxonomic levels with potentially different weights. In this approach, the levels of the source domain are weighted according to their hierarchical relevance, reflecting their relative importance to the prediction task. The proposed strategy enables the model to simultaneously leverage broader information from higher-level taxa and specific patterns from lower-level groups. Although evolutionary proximity may provide a useful assumption, the method allows the contribution of each level to be flexibly adjusted based on its actual relevance to the prediction task, enabling a more effective adaptation process.

6.4.1 Performance Results

The performance evaluation for the hierarchical domain adaptation modeling is presented in two parts: (1) a comparative analysis between the proposed hierarchical strategy and a baseline model, and (2) a statistical significance analysis of the observed results.

The baseline model was developed as a reference to assess the impact of incorporating hierarchical information in LBCE prediction. In this setup, all training data from different levels were combined and used uniformly, without considering their hierarchical relationships. No level-specific weighting was applied, and the model was trained in a flat manner, treating all examples equally regardless of their domain level. This represents a standard supervised learning approach without hierarchical domain adaptation.

1. Hierarchical domain adaptation vs Baseline

The majority of HDA models developed in this study outperformed the baseline model, indicating the effectiveness of incorporating hierarchical information to enhance predictive performance in LBCE prediction. Notably, the model for the Filoviridae family (Table 6.6) achieved an MCC of 0.595, an improvement of +0.227 over the baseline (MCC = 0.368). Similarly, the model for *C. trachomatis* reached an MCC of 0.469, exceeding the baseline (MCC = 0.240) by +0.229, and the Mononegavirales model obtained an MCC of 0.483, surpassing its baseline (MCC = 0.339) by +0.144. The Orthopoxvirus model also benefited considerably from hierarchical adaptation, with an MCC of 0.372, which is +0.316 higher than its baseline (MCC = 0.056).

In contrast, for some pathogens, improvements were minor or even negative. For *S. mansoni*, the hierarchical model resulted in an MCC of −0.003, underperforming relative to the baseline (MCC = 0.157). Similarly, the model for *T. gondii* showed a lower MCC (0.217) compared to its baseline (MCC = 0.326). Nevertheless, in most evaluated taxa, the hierarchical domain adaptation strategy consistently enhanced predictive performance, underscoring its robustness and generalizability for LBCE prediction across diverse pathogens.

Table 6.6: Comparison of methods using MCC across 17 selected datasets

Method	Dataset	Value
EpitopeTransfer	C. difficile	0.063
Baseline	C. difficile	-0.038
EpitopeTransfer	C. trachomatis	0.469
Baseline	C. trachomatis	0.240
EpitopeTransfer	Corynebacterium	0.181
Baseline	Corynebacterium	0.099
EpitopeTransfer	Enterobacteriaceae	0.235
Baseline	Enterobacteriaceae	0.156
EpitopeTransfer	Firoviridae	0.595
Baseline	Firoviridae	0.368
EpitopeTransfer	Human gammaherpesvirus 4	0.200
Baseline	Human gammaherpesvirus 4	0.178
EpitopeTransfer	Influenza A	0.384
Baseline	Influenza A	0.317
EpitopeTransfer	Lentivirus	0.256
Baseline	Lentivirus	0.227
EpitopeTransfer	Measles morbilivirus	0.100
Baseline	Measles morbilivirus	-0.105
EpitopeTransfer	Mononegavirales	0.483
Baseline	Mononegavirales	0.339
EpitopeTransfer	Orthopoxvirus	0.372
Baseline	Orthopoxvirus	0.056
EpitopeTransfer	Ovolvulus	0.167
Baseline	Ovolvulus	0.043
EpitopeTransfer	P. aeruginosa	-0.060
Baseline	P. aeruginosa	-0.163
EpitopeTransfer	P. falciparum	0.422

Method	Dataset	Value
Baseline	P. falciparum	0.343
EpitopeTransfer	S. mansoni	-0.003
Baseline	S. mansoni	0.157
EpitopeTransfer	Sars-cov-2	0.143
Baseline	Sars-cov-2	0.077
EpitopeTransfer	T. gondii	0.217
Baseline	T. gondii	0.326

Table 6.7: Summary of average test set performance (*mean ± standard error*) for EpitopeTransfer (HDA) and the baseline method across 17 selected datasets. Each row corresponds to a performance evaluation metric, and the values indicate the mean performance of each method over all datasets.

Metric	EpitopeTransfer	Baseline
AUC	0.698 (±0.027)	0.625 (±0.033)
F1	0.549 (±0.053)	0.454 (±0.056)
MCC	0.249 (±0.044)	0.154 (±0.039)
Balanced Accuracy	0.630 (±0.027)	0.581 (±0.020)
PPV	0.541 (±0.060)	0.545 (±0.072)
NPV	0.707 (±0.065)	0.620 (±0.058)
Sensitivity	0.664 (±0.072)	0.508 (±0.063)
Specificity	0.596 (±0.076)	0.654 (±0.072)

2. Statistical Significance Analysis

To evaluate the effectiveness of the HDA strategy, a statistical significance analysis compared the performance of HDA models with baseline models without hierarchical weighting. The comparison was conducted across several evaluation metrics: AUC, Balanced Accuracy, F1 Score, MCC, NPV, PPV, Sensitivity, and Specificity.

Table 6.8 summarizes the pairwise comparisons between HDA and baseline models. For each metric, the table reports the median of paired differences (with 95% confidence intervals), the associated p-values from Wilcoxon signed-rank tests, and indicates whether the differences are statistically significant at the 0.05 level.

Metrics with significant positive gains. The hierarchical domain adaptation strategy led to statistically significant improvements in multiple key metrics. Notably, MCC exhibited a median improvement of +0.091, with a p-value of 0.00934, indicating a consistent enhancement in predictive capability. Similar improvements were observed for AUC (+0.075; $p = 0.00934$), Balanced Accuracy (+0.049; $p = 0.01500$), F1 Score (+0.091; $p = 0.00934$), NPV (+0.075; $p = 0.00567$), and Sensitivity (+0.147; $p = 0.03479$). These findings suggest that the hierarchical weighting scheme improves the overall discriminative performance of the model and enhances its ability to correctly identify positive and negative examples.

Metrics with non-significant differences. Some metrics did not exhibit statistically significant differences. PPV showed a slight increase (+0.020), but this gain was not significant ($p = 0.45857$). Similarly, Specificity showed a small decrease (-0.021), which was also not statistically significant ($p = 0.73679$), indicating that hierarchical adaptation did not meaningfully affect specificity.

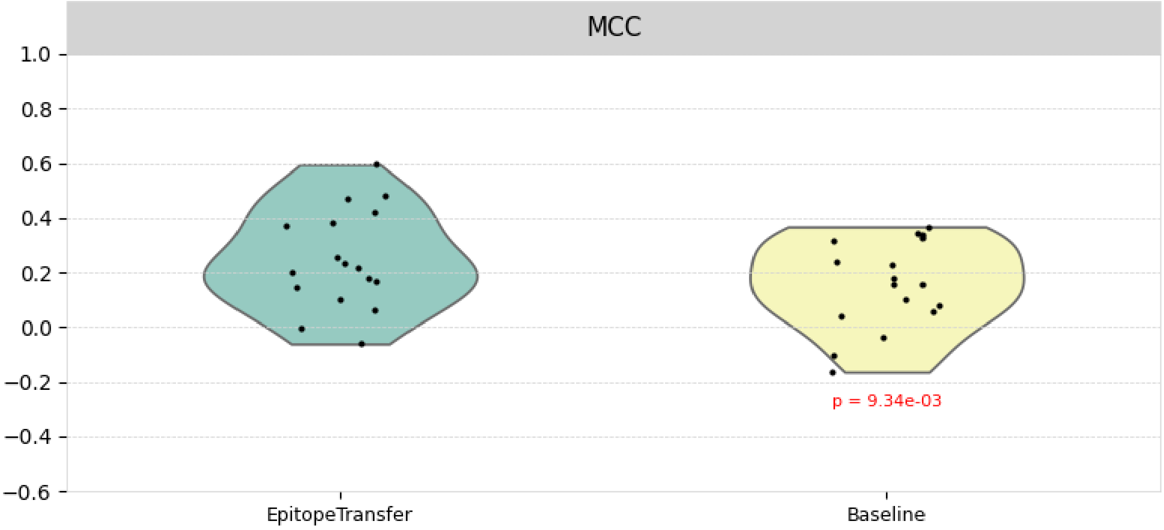


Figure 6.2: Comparison of EpitopeTransfer against the baseline method in terms of MCC. Violin plots represent the distribution of MCC values for each method across multiple datasets. A Wilcoxon Signed-Ranks test was performed on the median of paired differences, yielding a p-value of 9.34×10^{-3}. The p-value below 0.05 indicates a statistically significant difference at the 95% confidence level.

Table 6.8: Summary of comparison results between EpitopeTransfer (HDA) and the Baseline across all metrics

Metric	Pair	Medians of Diff.	p-value	Signif.
AUC	EpitopeTransfer vs Baseline	0.075 (0.017, 0.132)	0.00934	Yes
BACC	EpitopeTransfer vs Baseline	0.049 (0.024, 0.084)	0.01500	Yes
F1	EpitopeTransfer vs Baseline	0.091 (0.044, 0.131)	0.00934	Yes
MCC	EpitopeTransfer vs Baseline	0.091 (0.048, 0.154)	0.00934	Yes
NPV	EpitopeTransfer vs Baseline	0.075 (0.020, 0.148)	0.00567	Yes
PPV	EpitopeTransfer vs Baseline	0.020 (-0.032, 0.059)	0.45857	No
Sensit.	EpitopeTransfer vs Baseline	0.147 (0.032, 0.267)	0.03479	Yes
Specif.	EpitopeTransfer vs Baseline	-0.021 (-0.212, 0.110)	0.73679	No

6.5 SDA vs HDA

This section presents a comparative analysis of the proposed EpitopeTransfer method under two domain adaptation strategies: SDA and HDA.

A direct comparison between SDA and HDA as presented in this chapter must be interpreted with caution, as the two strategies were evaluated under different experimental conditions. These include substantial variations in model capacity, data availability, and the extent of optimization:

- **SDA**: leveraged a large ESM-2 model with 650 million parameters, full access to labeled data at the higher level, and a hyperparameter search with 100 trials.
- **HDA**: relied on a small ESM-2 model with 8 million parameters, 150 peptides per level, and a significantly smaller optimization of 5 trials.

As shown in Table 6.9, the performance of HDA was comparable to that of SDA across the 17 target taxa, with some variation observed for individual datasets. Table 6.10 presents the results of the Wilcoxon paired test, indicating that there was no statistically significant difference between HDA and SDA across all evaluated metrics (p-values $>$ 0.05).

Table 6.9: Performance estimates of EpitopeTransfer under Hierarchical (HDA) and Single-level Domain Adaptation (SDA), based on MCC scores for 17 target taxa.

Dataset	**HDA (Hierarchical)**	**SDA (Single)**
C. difficile	0.063	0.137
C. trachomatis	0.469	0.568
Corynebacterium	0.181	0.282
Enterobacteriaceae	0.235	0.427
Filoviridae	0.595	0.610
Human gammaherpesvirus 4	0.200	0.275
Influenza A	0.384	0.218
Lentivirus	0.256	0.350
Measles morbilivirus	0.100	0.068
Mononegavirales	0.483	0.428
Orthopoxvirus	0.372	0.168
O. volvulus	0.167	0.095
P. aeruginosa	-0.060	0.249
P. falciparum	0.422	0.410
S. mansoni	-0.003	0.069
SARS-CoV-2	0.143	0.018
T. gondii	0.217	0.312

Table 6.10: Wilcoxon paired test between HDA and SDA for all metrics.

Metric	Pair	Medians of diff. (95% CI)	p-value	Significant
AUC	SDA vs HDA	-0.005 (-0.040, 0.038)	0.84980	No
F1	SDA vs HDA	0.033 (-0.049, 0.086)	0.40376	No
MCC	SDA vs HDA	0.030 (-0.045, 0.090)	0.35589	No
BACC	SDA vs HDA	0.004 (-0.040, 0.044)	0.83126	No
PPV	SDA vs HDA	0.035 (-0.046, 0.083)	0.37782	No
NPV	SDA vs HDA	0.012 (-0.036, 0.096)	0.58612	No
Sensit.	SDA vs HDA	-0.012 (-0.186, 0.141)	0.74666	No
Specif.	SDA vs HDA	0.048 (-0.134, 0.212)	0.64413	No

In summary, even though SDA employed a more resource-intensive configuration, HDA still achieved similar performance under more modest conditions. A more rigorous comparison under the same conditions (model capacity, data availability, and the extent of optimization) will be developed in future work.

6.6 Conclusion

This chapter presented a case study demonstrating the application of the proposed single and hierarchical domain adaptation method to the task of LBCE prediction. The results show that incorporating phylogenetic structure into domain adaptation yields consistent performance improvements in many scenarios, both under the single and hierarchical configurations.

While the single domain adaptation approach already demonstrated substantial gains over three external and two internal baselines, the hierarchical modeling further generalized this strategy by enabling knowledge transfer across several taxonomic levels. The proposed method showed particular advantages for pathogens where structured phylogenetic relationships offer complementary information to enhance prediction.

In conclusion, the comparative analysis between SDA and HDA revealed that, while SDA benefited from a more powerful setup, HDA was still able to deliver competitive results despite operating under significantly more constrained conditions. These findings reinforce the potential of hierarchical domain adaptation as a viable strategy in low-resource scenarios. However, a more rigorous and fair comparison - using equivalent model sizes, data volumes, and optimization efforts - is necessary to fully assess its advantages and limitations.

Chapter 7

Discussion

This thesis addressed the challenge of transferring knowledge in neural language models across domains that are hierarchically structured. A central research question guided this work: **Given data that is hierarchically structured, how can knowledge be effectively transferred from higher levels, where data is abundant, to lower levels, where data is scarce?** To answer this question, the thesis proposes a domain adaptation method that models hierarchical relationships and progressively adapts across taxonomic levels.

7.1 Contributions

This thesis proposed a method for hierarchical domain adaptation in neural language models, with application on the task of linear B-cell epitope prediction.

Initially, the method was evaluated under a SDA strategy across 20 distinct datasets and compared to five state-of-the-art LBCE predictors. In this setting, the method consistently outperformed the baselines in terms of AUC, MCC, Balanced Accuracy, F1 Score, NPV, PPV and Sensitivity. Statistical significance analysis confirmed that the improvements were not due to random variations, but represented significant gains. The SDA approach yielded the best performance for several pathogens. For instance, the model for the Filoviridae family achieved an MCC of 0.766, representing substantial improvements over BepiPred 3.0 (0.143), EpiDope (0.240), and EpitopeVec (0.202). The *Plasmodium falciparum* model achieves an MCC of 0.505, showing improvements over BepiPred 3.0 (0.119), EpiDope (0.088), and EpitopeVec (0.018), while the genus Lentivirus model exhibits an MCC of 0.770, outperforming BepiPred 3.0 (0.376), EpiDope (0.033), EpitopeVec (0.067). The Enterobacteriaceae family model achieves an MCC of 0.479, surpassing BepiPred 3.0 (0.054), EpiDope (0.144), and EpitopeVec(0.061)

Following this, the generalization of the method was tested under a HDA strategy applied to 17 target domains. A baseline for comparison was constructed by removing the hierarchical weights, enabling an evaluation of the added value brought by the hierarchical modeling. The results again favored the proposed method, with HDA demonstrating superior performance in the vast majority of cases. Among the evaluated datasets, the model for the Filoviridae family achieved an MCC of 0.595, improving over the baseline of 0.368. The *C. trachomatis* model reached an MCC of 0.469, compared to the baseline of 0.240, and the Mononegavirales model obtained an MCC of 0.483, an improvement over the baseline of 0.339. The Orthopoxvirus model also benefited significantly, achieving an MCC of 0.372 versus a baseline of 0.056.

The results support a positive answer to the central question posed: **it is indeed possible to effectively transfer knowledge from higher-level, data-rich domains to lower-level, data-scarce domains when hierarchical relationships are explicitly modeled**.

This thesis proposed a HDA method tailored for neural language models, with the following key contributions:

- A hierarchical training strategy that progressively adapts models across taxonomic levels, enabling smoother knowledge transfer from higher-level to lower-level domains.
- The introduction of the *Hierarchical Weighted Cross-Entropy Loss*, which incorporates level-specific weights to dynamically identify and prioritize the most relevant level for the target domain. Additionally, the loss function incorporates weighting strategies to balance positive and negative samples across the hierarchy, reducing potential biases and improving model generalization.
- Successful application of the proposed method to the task of linear B-cell epitope prediction, demonstrating its effectiveness in a real-world problem. This task is particularly relevant for the development of vaccines, therapeutic antibodies, and immunodiagnostic tools.

7.2 Limitations

This section details the main limitations identified in this study. The three categories concern the application of the approaches in the case study of epitope prediction: (1) limitations of the SDA modeling approach, (2) limitations of the HDA modeling approach, and (3) limitations derived from the comparative analysis between both strategies.

(1) Single Domain Adaptation Modeling

Although the SDA strategy demonstrated high performance across several taxa, there were exceptions where the proposed *EpitopeTransfer* method underperformed in comparison to existing models. For example, in the case of *SARS-CoV-2*, *EpitopeTransfer* achieved a MCC of only 0.043, considerably lower than the 0.169 obtained by Epidope. Likewise, for *M. tuberculosis*, the model recorded an MCC of -0.031, while ESM-1b achieved 0.039. These findings suggest that, although SDA-based *EpitopeTransfer* generally outperformed other approaches, it may not be suitable for all pathogens. In particular, its use may not be recommended for cases such as *SARS-CoV-2* and *M. tuberculosis*, where it underperformed compared to baseline models.

(2) Hierarchical Domain Adaptation Modeling

The HDA strategy also presented limitations. For instance, in the case of *S. mansoni*, the model achieved a MCC of -0.003, which was lower than the 0.157 obtained by the baseline. Similarly, for *T. gondii*, the MCC was 0.217, compared to 0.326 for the baseline. In certain cases, such as *S. mansoni* and *T. gondii*, a decline in performance was observed compared to baseline models, indicating that HDA might be less suitable in such contexts.

(3) Comparison between SDA and HDA

When compared to HDA, the SDA strategy did not demonstrate statistically significant differences in performance across any evaluated metric, as shown in Table 6.10. According to the Wilcoxon paired test, neither approach outperformed the other in terms of AUC, F1, MCC, BACC, PPV, NPV, sensitivity, or specificity.

It is important to note that, despite differences in experimental configuration — such as the limitation to 150 labeled peptides per hierarchical level for HDA and a more restricted hyperparameter search (5 trials for HDA versus 100 for SDA) — SDA did not achieve superior results; in fact, HDA achieved comparable performance even under less favorable conditions. Therefore, a rigorous evaluation under the same experimental setup is necessary to determine whether HDA can truly outperform SDA.

7.3 Future Work

From a methodological perspective, some potential research directions emerge from this thesis:

- Improve optimization by refining the stopping criteria. The regret-based method proposed by [Makarova et al., 2022] was initially adopted but was found to be insufficient to prevent overfitting in the epitope prediction task. This reveals a key challenge in applying the fully neural network-based HDA approach: determining the appropriate extent of optimization required to maximize performance while avoiding overfitting, as the optimal stopping point may vary considerably across different taxa.

- Extending the method to unsupervised hierarchical domain adaptation scenarios. Although the proposed approach was designed for settings with limited labeled samples in the target domain, an interesting direction for future research is to explore its applicability in fully unsupervised contexts, where no target labels are available. This would allow the method to operate without supervision in the target domain, expanding its usability in real-world scenarios where annotated data is unavailable.

- Incorporating interpretability mechanisms into the HDA framework to identify which regions of the input sequence most influenced the prediction at each position. In the context of epitope prediction, for example, this would enable a better understanding of the underlying biological mechanisms that lead the model to classify a given residue as an epitope or not.

Appendices

Appendix A

Embedder Development Details

The *Embedder Development* process has two main components: ESM-1b fine-tuning and feature calculation. A key aspect to fine-tuning step is the use of a sliding window technique to compute amino acid level data.

Sliding window

The sliding window involves capturing contextual sequences, considering the window size along the entire protein length. This process ensures that each amino acid is analyzed within its specific context, comprising neighboring sequences. In this work, a window size of 1024 is utilized, aligning with the maximum capacity permitted by the ESM-1b model.

Mathematically, the sliding window is defined as follows: Let $S = [a_1, a_2, ..., a_n]$ be the amino acid sequence, where n is the length of the sequence. Let w be the window size, and let p be the current position in the sequence, starting from p_{start} to p_{end}, where $1 \leq p \leq n$. Define $L(p, w)$ and $R(p, w)$ as functions that return the amino acids to the left and right of p, respectively, considering the window size w. These functions can be defined as follows:

$$L(p, w) = \begin{cases} S[1 : p] & \text{if } p \leq w, \\ S[p - w : p] & \text{otherwise.} \end{cases}$$

$$R(p, w) = \begin{cases} S[p + 1 : n] & \text{if } w \geq n - p, \\ S[p + 1 : p + 1 + w] & \text{otherwise.} \end{cases}$$

The algorithm iterates over each position p in the sequence S, starting from p_{start} and ending at p_{end}. For each p, it performs the following steps:

1. Extracts the central amino acid a_p.

2. Determines the left sequence $L(p, w)$.

3. Determines the right sequence $R(p, w)$.

4. Records the label associated with the position.

ESM-1b fine-tuning

Fine-tuning of the ESM-1b model was performed by integrating a classification head into the original architecture, as detailed in subsection “ESM-1b Model Architecture”. This fine-tuning involves a retraining of all layers of the model. Thus, the initial model trained for general protein sequence analysis is now tailored to epitope prediction task. To this procedure, the dataset comprises sequences from higher-level taxa, as outlined in section “Data extraction and preparation” of the main text and detailed in Supplementary Table 1. The fine-tuning process leverages the ESM-1b model’s capabilities to learn from our specialized dataset, focusing on epitopes and non-epitopes present in these sequences. In the fine-tuning phase, the ESM-1b model processes each training sample, extracted through the sliding window technique. The fine-tuning is executed over three epochs and the model learns to identify and differentiate between epitopes and non-epitopes in the context of the higher-level taxa sequences.

The result of this procedure is a fine-tuned model specifically capable of identifying epitopes in protein sequences derived from higher-level taxa. This model is now able to generate enriched features to epitope prediction classification tasks.

Feature Calculation

In this step, the fine-tuned ESM-1b model, pre-trained with data from higher taxonomic levels, is used to process the amino acid sequences and generate enriched features for epitope prediction tasks at lower taxonomic levels. The complete protein sequence is fed into the optimized ESM-1b model, enabling it to generate a more enriched representation. From this output, only the labeled peptide regions are selected for training. Within this identified regions of interest in the protein sequences, each amino acid is extracted and represented as a 1280-dimensional vector. This feature representation, a numerical array generated by the model, encodes the properties and contextual information that the model has learned. More specifically, this vector representation reflects both its inherent characteristics and its relational context in the protein sequence.

ESM-1b Model Architecture

To enable the fine tuning of ESM-1b in this work, we added a “Classification Head” layer to the original model architecture, which is later removed to enable the extraction of features from the taxon-optimised model. The architecture can be summarized by its key components, which include:

Embeddings:

- *Word Embeddings:* Convert each amino acid in a sequence to a 1280-dimensional vector. The word embeddings are **trainable**, meaning their values are adjusted during model training to improve task-specific performance.
- *Position Embeddings:* Provide positional context to each amino acid in sequences up to 1026 positions long. Unlike the word embeddings, the position embeddings are **fixed** and are not updated during training, ensuring that the positional information remains consistent.

Encoder:

- Comprises 33 layers, each featuring:
  - *Self-Attention Mechanism:* Allows each position to interact with every other position in the sequence.
  - *Feed-Forward Network:* Enhances the model output from the self-attention mechanism by applying transformations.
  - *Layer Normalization:* Applied for training stability.

Contact Prediction Head:

- A specialized component for predicting protein structure contacts. This layer involves determining which pairs of amino acids within a protein sequence are in close proximity to each other in the 3D structure, referred to as contacts.

The ESM-1b model, tailored for protein sequence analysis, comprises a 33-layer encoder, where each position in the protein sequence is represented by a 1280-dimensional vector. While the model supports an embedding size of 1026, the practical window for sequence analysis utilizes 1024 positions. This is due to the allocation of one special token at the beginning and another at the end of the sequence.

Training and Validation Procedures

Cross-Validation

1. **Test and Train Split:** For each taxon, the data is split into five folds. One fold is set aside for testing, and the remaining four are used for training and validation. In this step, a nested cross-validation is performed using the remaining four folds: three folds are used for training and one fold is used for validation, such that the models after the hyperparameter tuning step are assessed on unseen data. The best hyperparameters are defined during this process. After this, the model is retrained

using all four folds, and the optimal threshold for MCC is determined. Finally, the model's performance is evaluated on the test set.

2. **Hyperparameter Search:** Random Forest hyperparameters are optimized using Bayesian optimization as implemented in the *Optuna* package Akiba et al. [2019]:

    - `n_estimators` (range: 100 to 500),
    - `max_depth` (choices: None, 10, 20, 30),
    - `min_samples_split` (range: 2 to 10),
    - `min_samples_leaf` (range: 1 to 10),
    - `max_features` (choices: sqrt, log2, None),
    - `bootstrap` (choices: True, False),
    - `criterion` (choices: gini, entropy)

Special Cases

Due to the scarcity of samples for *Orthopoxvirus*, *Corynebacterium*, and *Measles morbilivirus*, we followed a standard train-test procedure for these cases instead of cross-validation. For each taxon, the data was split into two folds. One fold was set aside for testing, and the remaining fold was used for training. Although cross-validation was not used, the same Bayesian optimization process described earlier was followed to identify the best hyperparameters. The optimal MCC threshold was determined during the training phase and is then applied to evaluate the model's performance on the test dataset.

Performance Indicators

To guarantee a thorough evaluation and align with established benchmarks in the field, a range of performance indicators is employed. These metrics not only allow for a comparison with existing studies but also provide a detailed insight into the predictive accuracy of the models under consideration. The definitions of these indicators involve key terms such as TP (True Positives), TN (True Negatives), FP (False Positives), and FN (False Negatives), which are fundamental to the formulae that follow.

- **Positive Predictive Value (PPV)**: This metric measures the probability that an identified amino acid is accurately part of an epitope sequence. It serves as an indicator of the reliability of a model's positive predictions. Such reliability is relevant for justifying the allocation of resources toward experimental validation of predicted targets. The Positive Predictive Value (PPV) is commonly recognized in the field as Precision.

$$PPV = \frac{TP}{TP + FP}$$

- **Negative Predictive Value (NPV)**: This metric assesses the likelihood that an amino acid classified as not being part of an epitope is correctly identified. It demonstrates the model's proficiency in identifying amino acids that belong to non-epitope peptide sequences.

$$NPV = \frac{TN}{TN + FN}$$

- **Sensitivity (SENS)**: Sensitivity, also called the True Positive Rate (TPR) or Recall, measures the model's ability to correctly identify amino acids that are part of epitope sequences. This metric is crucial for evaluating the model's proficiency in accurately detecting positive amino instances within epitopes.

$$SENS = \frac{TP}{TP + FN}$$

- **Specificity (SPEC)**: Specificity, or the True Negative Rate (TNR), is the analog of sensitivity that focuses on the negative class.

$$SPEC = \frac{TN}{TN + FP}$$

- **Balanced Accuracy (BACC)**: This metric accounts for potential class imbalance by computing the average of sensitivity (true positive rate) and specificity (true negative rate). Unlike standard accuracy, balanced accuracy provides a more reliable evaluation in scenarios where one class may be underrepresented, which is particularly relevant in epitope prediction tasks.

$$BACC = \frac{1}{2}\left(\frac{TP}{TP + FN} + \frac{TN}{TN + FP}\right)$$

- **AUC (Area Under the ROC Curve)**

$$\text{AUC} = \int_0^1 \mathcal{C}(x)\,dx \tag{1}$$

where $\mathcal{C}(x)$ denotes the curve of Sensitivity versus (1 - Specificity), derived by varying the classification threshold from zero and one.

- **F1 Score**

$$\mathrm{F1} = \frac{2 \times \mathrm{PPV} \times \mathrm{SENS}}{\mathrm{PPV} + \mathrm{SENS}} \tag{2}$$

  The F1 Score is a metric that balances Precision and Recall, making it useful for evaluating binary classification models, especially when dealing with imbalanced datasets. It combines Precision (the ability to correctly identify positive samples) and Recall (the ability to capture all positive samples) into a single score. The F1 Score ranges from 0 to 1, where higher values indicate better model performance.

- **MCC (Matthews Correlation Coefficient)**

$$\mathrm{MCC} = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP} + \mathrm{FP})(\mathrm{TP} + \mathrm{FN})(\mathrm{TN} + \mathrm{FP})(\mathrm{TN} + \mathrm{FN})}} \tag{3}$$

  The Matthews Correlation Coefficient (MCC) is a metric that takes into account True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN) to assess the performance of binary classification models. It produces values between -1 and +1, where +1 indicates perfect prediction, 0 corresponds to random prediction, and -1 reflects total disagreement between predictions and actual outcomes. MCC is particularly useful when dealing with imbalanced datasets.

Appendix B

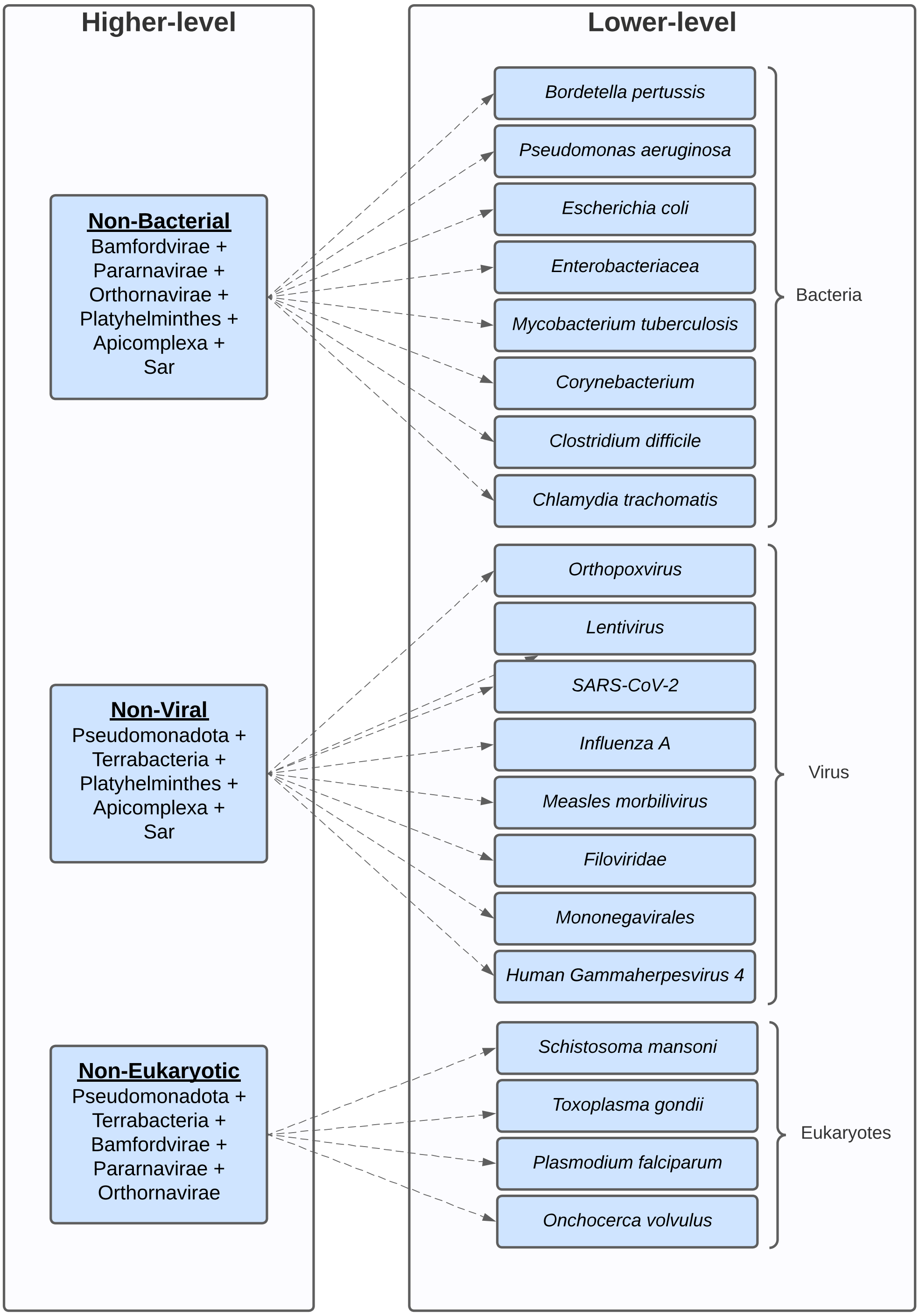


Figure 1: Set of unrelated groups used to establish the non-phylogenetic baselines. These pairs of datasets are used to perform domain adaptation across unrelated biological groups, to test our hypothesis about the effectiveness of using phylogenetically related data for the embedder fine tuning. If the non-phylogenetic baseline models produced equivalent of favorable results, the enhancements observed in epitope prediction tasks could be exclusively attributed to the effect of fine tuning the representation with LBCE data, which would indicate that considering phylogeny is unnecessary. Conversely, favorable results for the EpitopeTransfer models would indicate the presence of some positive effect of the phylogeny-aware data filtering step.

Appendix C

The estimated performance for each method on each dataset is presented. *Method* refers to the employed approach, including the primary method, **EpitopeTransfer**, which leverages phylogenetic information, and internal and external baselines. The internal baselines are **ESM-1b (650M)** baseline, a pretrained protein language model fine-tuned for epitope prediction, and **Non-phylogenetic transfer (NPTransfer)**, a transfer learning method that does not utilize phylogenetic relationships. The external baselines include **BepiPred 3.0**, **Epitope**, and **EpitopeVec**, which are methods developed outside this study and are included for comparative evaluation. *Dataset* corresponds to the data from 20 specific taxa, and *Value* represents the value of each presented metric. The evaluated metrics include **AUC** (Area Under the Curve), **F1** score, **MCC** (Matthews Correlation Coefficient), **Accuracy**, **PPV** (Positive Predictive Value), **NPV** (Negative Predictive Value), **Sensitivity**, and **Specificity**.

Table 1: Comparison of methods for AUC

Method	**Dataset**	**Value**
EpitopeTransfer	B. pertussis	0.555
BepiPred 3.0	B. pertussis	0.365
EpiDope	B. pertussis	0.359
EpitopeVec	B. pertussis	0.750
ESM-1b	B. pertussis	0.473
NPTransfer	B. pertussis	0.412
EpitopeTransfer	C. difficile	0.707
BepiPred 3.0	C. difficile	0.425
EpiDope	C. difficile	0.744
EpitopeVec	C. difficile	0.851
ESM-1b	C. difficile	0.514
NPTransfer	C. difficile	0.584
EpitopeTransfer	Corynebacterium	0.590
BepiPred 3.0	Corynebacterium	0.648
EpiDope	Corynebacterium	0.733
EpitopeVec	Corynebacterium	0.728
ESM-1b	Corynebacterium	0.461
NPTransfer	Corynebacterium	0.450

Method	**Dataset**	**Value**
EpitopeTransfer	C. trachomatis	0.773
BepiPred 3.0	C. trachomatis	0.559
EpiDope	C. trachomatis	0.665
EpitopeVec	C. trachomatis	0.717
ESM-1b	C. trachomatis	0.769
NPTransfer	C. trachomatis	0.775
EpitopeTransfer	E. coli	0.853
BepiPred 3.0	E. coli	0.400
EpiDope	E. coli	0.804
EpitopeVec	E. coli	0.533
ESM-1b	E. coli	0.855
NPTransfer	E. coli	0.830
EpitopeTransfer	Enterobacteriaceae	0.826
BepiPred 3.0	Enterobacteriaceae	0.554
EpiDope	Enterobacteriaceae	0.613
EpitopeVec	Enterobacteriaceae	0.549
ESM-1b	Enterobacteriaceae	0.722
NPTransfer	Enterobacteriaceae	0.728
EpitopeTransfer	Filoviridae	0.972
BepiPred 3.0	Filoviridae	0.538
EpiDope	Filoviridae	0.877
EpitopeVec	Filoviridae	0.752
ESM-1b	Filoviridae	0.944
NPTransfer	Filoviridae	0.966
EpitopeTransfer	Human gammaherpesvirus 4	0.593
BepiPred 3.0	Human gammaherpesvirus 4	0.398
EpiDope	Human gammaherpesvirus 4	0.617
EpitopeVec	Human gammaherpesvirus 4	0.560
ESM-1b	Human gammaherpesvirus 4	0.608
NPTransfer	Human gammaherpesvirus 4	0.593
EpitopeTransfer	Influenza A	0.756
BepiPred 3.0	Influenza A	0.570
EpiDope	Influenza A	0.523

Method	**Dataset**	**Value**
EpitopeVec	Influenza A	0.630
ESM-1b	Influenza A	0.719
NPTransfer	Influenza A	0.632
EpitopeTransfer	Lentivirus	0.789
BepiPred 3.0	Lentivirus	0.581
EpiDope	Lentivirus	0.552
EpitopeVec	Lentivirus	0.596
ESM-1b	Lentivirus	0.794
NPTransfer	Lentivirus	0.648
EpitopeTransfer	M. tuberculosis	0.478
BepiPred 3.0	M. tuberculosis	0.444
EpiDope	M. tuberculosis	0.481
EpitopeVec	M. tuberculosis	0.481
ESM-1b	M. tuberculosis	0.506
NPTransfer	M. tuberculosis	0.440
BepiPred 3.0	Measles morbilivirus	0.381
EpiDope	Measles morbilivirus	0.501
EpitopeVec	Measles morbilivirus	0.538
EpitopeTransfer	Measles morbilivirus	0.522
ESM-1b	Measles morbilivirus	0.530
NPTransfer	Measles morbilivirus	0.599
EpitopeTransfer	Mononegavirales	0.725
BepiPred 3.0	Mononegavirales	0.446
EpiDope	Mononegavirales	0.817
EpitopeVec	Mononegavirales	0.671
ESM-1b	Mononegavirales	0.740
NPTransfer	Mononegavirales	0.739
EpitopeTransfer	Orthopox	0.689
BepiPred 3.0	Orthopox	0.728
EpiDope	Orthopox	0.688
EpitopeVec	Orthopox	0.322
ESM-1b	Orthopox	0.623
NPTransfer	Orthopox	0.564

Method	Dataset	Value
EpitopeTransfer	Ovolvulus	0.626
BepiPred 3.0	Ovolvulus	0.721
EpiDope	Ovolvulus	0.495
EpitopeVec	Ovolvulus	0.585
ESM-1b	Ovolvulus	0.673
NPTransfer	Ovolvulus	0.636
EpitopeTransfer	P. aeruginosa	0.721
BepiPred 3.0	P. aeruginosa	0.040
EpiDope	P. aeruginosa	0.874
EpitopeVec	P. aeruginosa	0.565
ESM-1b	P. aeruginosa	0.669
NPTransfer	P. aeruginosa	0.790
EpitopeTransfer	P. falciparum	0.810
BepiPred 3.0	P. falciparum	0.675
EpiDope	P. falciparum	0.603
EpitopeVec	P. falciparum	0.512
ESM-1b	P. falciparum	0.792
NPTransfer	P. falciparum	0.786
EpitopeTransfer	S. mansoni	0.557
BepiPred 3.0	S. mansoni	0.560
EpiDope	S. mansoni	0.672
EpitopeVec	S. mansoni	0.447
ESM-1b	S. mansoni	0.544
NPTransfer	S. mansoni	0.556
EpitopeTransfer	Sars-cov-2	0.547
BepiPred 3.0	Sars-cov-2	0.569
EpiDope	Sars-cov-2	0.597
EpitopeVec	Sars-cov-2	0.630
ESM-1b	Sars-cov-2	0.539
NPTransfer	Sars-cov-2	0.502
EpitopeTransfer	T. gondii	0.705
BepiPred 3.0	T. gondii	0.454
EpiDope	T. gondii	0.466

Method	Dataset	Value
EpitopeVec	T. gondii	0.620
ESM-1b	T. gondii	0.654
NPTransfer	T. gondii	0.602

Table 2: Comparison of methods for F1

Method	Dataset	Value
EpitopeTransfer	B. pertussis	0.836
BepiPred 3.0	B. pertussis	0.778
EpiDope	B. pertussis	0.288
EpitopeVec	B. pertussis	0.754
ESM-1b	B. pertussis	0.822
NPTransfer	B. pertussis	0.836
EpitopeTransfer	C. difficile	0.236
BepiPred 3.0	C. difficile	0.000
EpiDope	C. difficile	0.000
EpitopeVec	C. difficile	0.282
ESM-1b	C. difficile	0.177
NPTransfer	C. difficile	0.171
EpitopeTransfer	Corynebacterium	0.557
BepiPred 3.0	Corynebacterium	0.303
EpiDope	Corynebacterium	0.286
EpitopeVec	Corynebacterium	0.672
ESM-1b	Corynebacterium	0.454
NPTransfer	Corynebacterium	0.360
EpitopeTransfer	C. trachomatis	0.717
BepiPred 3.0	C. trachomatis	0.583
EpiDope	C. trachomatis	0.311
EpitopeVec	C. trachomatis	0.624
ESM-1b	C. trachomatis	0.618
NPTransfer	C. trachomatis	0.711
EpitopeTransfer	E. coli	0.872
BepiPred 3.0	E. coli	0.375

Method	Dataset	Value
EpiDope	E. coli	0.340
EpitopeVec	E. coli	0.628
ESM-1b	E. coli	0.863
NPTransfer	E. coli	0.874
EpitopeTransfer	Enterobacteriaceae	0.738
BepiPred 3.0	Enterobacteriaceae	0.434
EpiDope	Enterobacteriaceae	0.188
EpitopeVec	Enterobacteriaceae	0.534
ESM-1b	Enterobacteriaceae	0.632
NPTransfer	Enterobacteriaceae	0.619
EpitopeTransfer	Filoviridae	0.780
BepiPred 3.0	Filoviridae	0.235
EpiDope	Filoviridae	0.316
EpitopeVec	Filoviridae	0.271
ESM-1b	Filoviridae	0.603
NPTransfer	Filoviridae	0.742
EpitopeTransfer	Human gammaherpesvirus 4	0.444
BepiPred 3.0	Human gammaherpesvirus 4	0.273
EpiDope	Human gammaherpesvirus 4	0.244
EpitopeVec	Human gammaherpesvirus 4	0.508
ESM-1b	Human gammaherpesvirus 4	0.377
NPTransfer	Human gammaherpesvirus 4	0.429
EpitopeTransfer	Influenza A	0.818
BepiPred 3.0	Influenza A	0.539
EpiDope	Influenza A	0.206
EpitopeVec	Influenza A	0.798
ESM-1b	Influenza A	0.824
NPTransfer	Influenza A	0.814
EpitopeTransfer	Lentivirus	0.925
BepiPred 3.0	Lentivirus	0.490
EpiDope	Lentivirus	0.457
EpitopeVec	Lentivirus	0.571
ESM-1b	Lentivirus	0.877

Method	**Dataset**	**Value**
NPTransfer	Lentivirus	0.847
EpitopeTransfer	M. tuberculosis	0.586
BepiPred 3.0	M. tuberculosis	0.254
EpiDope	M. tuberculosis	0.155
EpitopeVec	M. tuberculosis	0.510
ESM-1b	M. tuberculosis	0.648
NPTransfer	M. tuberculosis	0.578
BepiPred 3.0	Measles morbilivirus	0.281
EpiDope	Measles morbilivirus	0.356
EpitopeVec	Measles morbilivirus	0.587
EpitopeTransfer	Measles morbilivirus	0.000
ESM-1b	Measles morbilivirus	0.000
NPTransfer	Measles morbilivirus	0.000
EpitopeTransfer	Mononegavirales	0.565
BepiPred 3.0	Mononegavirales	0.271
EpiDope	Mononegavirales	0.499
EpitopeVec	Mononegavirales	0.495
ESM-1b	Mononegavirales	0.593
NPTransfer	Mononegavirales	0.547
EpitopeTransfer	Orthopox	0.384
BepiPred 3.0	Orthopox	0.492
EpiDope	Orthopox	0.351
EpitopeVec	Orthopox	0.138
ESM-1b	Orthopox	0.298
NPTransfer	Orthopox	0.274
EpitopeTransfer	Ovolvulus	0.362
BepiPred 3.0	Ovolvulus	0.364
EpiDope	Ovolvulus	0.053
EpitopeVec	Ovolvulus	0.262
ESM-1b	Ovolvulus	0.250
NPTransfer	Ovolvulus	0.347
EpitopeTransfer	P. aeruginosa	0.835
BepiPred 3.0	P. aeruginosa	0.000

Method	Dataset	Value
EpiDope	P. aeruginosa	0.116
EpitopeVec	P. aeruginosa	0.698
ESM-1b	P. aeruginosa	0.691
NPTransfer	P. aeruginosa	0.581
EpitopeTransfer	P. falciparum	0.826
BepiPred 3.0	P. falciparum	0.372
EpiDope	P. falciparum	0.431
EpitopeVec	P. falciparum	0.642
ESM-1b	P. falciparum	0.815
NPTransfer	P. falciparum	0.784
EpitopeTransfer	S. mansoni	0.437
BepiPred 3.0	S. mansoni	0.367
EpiDope	S. mansoni	0.370
EpitopeVec	S. mansoni	0.296
ESM-1b	S. mansoni	0.330
NPTransfer	S. mansoni	0.220
EpitopeTransfer	Sars-cov-2	0.120
BepiPred 3.0	Sars-cov-2	0.136
EpiDope	Sars-cov-2	0.262
EpitopeVec	Sars-cov-2	0.222
ESM-1b	Sars-cov-2	0.164
NPTransfer	Sars-cov-2	0.119
EpitopeTransfer	T. gondii	0.811
BepiPred 3.0	T. gondii	0.703
EpiDope	T. gondii	0.297
EpitopeVec	T. gondii	0.682
ESM-1b	T. gondii	0.806
NPTransfer	T. gondii	0.718

Table 3: Comparison of methods for MCC

Method	Dataset	Value
EpitopeTransfer	B. pertussis	0.000

Method	Dataset	Value
BepiPred 3.0	B. pertussis	-0.008
EpiDope	B. pertussis	-0.108
EpitopeVec	B. pertussis	0.351
ESM-1b	B. pertussis	-0.090
NPTransfer	B. pertussis	0.000
EpitopeTransfer	C. difficile	0.173
BepiPred 3.0	C. difficile	-0.027
EpiDope	C. difficile	0.000
EpitopeVec	C. difficile	0.279
ESM-1b	C. difficile	0.052
NPTransfer	C. difficile	0.000
EpitopeTransfer	Corynebacterium	0.064
BepiPred 3.0	Corynebacterium	0.247
EpiDope	Corynebacterium	0.309
EpitopeVec	Corynebacterium	0.315
ESM-1b	Corynebacterium	-0.134
NPTransfer	Corynebacterium	-0.022
EpitopeTransfer	C. trachomatis	0.447
BepiPred 3.0	C. trachomatis	-0.055
EpiDope	C. trachomatis	0.137
EpitopeVec	C. trachomatis	0.286
ESM-1b	C. trachomatis	0.385
NPTransfer	C. trachomatis	0.440
EpitopeTransfer	E. coli	0.325
BepiPred 3.0	E. coli	-0.136
EpiDope	E. coli	0.248
EpitopeVec	E. coli	0.031
ESM-1b	E. coli	0.217
NPTransfer	E. coli	0.349
EpitopeTransfer	Enterobacteriaceae	0.479
BepiPred 3.0	Enterobacteriaceae	0.054
EpiDope	Enterobacteriaceae	0.144
EpitopeVec	Enterobacteriaceae	0.061

Method	**Dataset**	**Value**
ESM-1b	Enterobacteriaceae	-0.035
NPTransfer	Enterobacteriaceae	-0.113
EpitopeTransfer	Filoviridae	0.766
BepiPred 3.0	Filoviridae	0.143
EpiDope	Filoviridae	0.240
EpitopeVec	Filoviridae	0.202
ESM-1b	Filoviridae	0.558
NPTransfer	Filoviridae	0.716
EpitopeTransfer	Human gammaherpesvirus 4	0.241
BepiPred 3.0	Human gammaherpesvirus 4	-0.180
EpiDope	Human gammaherpesvirus 4	0.068
EpitopeVec	Human gammaherpesvirus 4	0.070
ESM-1b	Human gammaherpesvirus 4	0.249
NPTransfer	Human gammaherpesvirus 4	0.209
EpitopeTransfer	Influenza A	0.176
BepiPred 3.0	Influenza A	0.239
EpiDope	Influenza A	0.054
EpitopeVec	Influenza A	0.139
ESM-1b	Influenza A	-0.058
NPTransfer	Influenza A	0.099
EpitopeTransfer	Lentivirus	0.770
BepiPred 3.0	Lentivirus	0.376
EpiDope	Lentivirus	0.033
EpitopeVec	Lentivirus	0.067
ESM-1b	Lentivirus	0.615
NPTransfer	Lentivirus	0.469
EpitopeTransfer	M. tuberculosis	-0.031
BepiPred 3.0	M. tuberculosis	0.029
EpiDope	M. tuberculosis	0.033
EpitopeVec	M. tuberculosis	-0.008
ESM-1b	M. tuberculosis	0.039
NPTransfer	M. tuberculosis	-0.056
BepiPred 3.0	Measles morbilivirus	-0.310

Method	**Dataset**	**Value**
EpiDope	Measles morbilivirus	0.079
EpitopeVec	Measles morbilivirus	0.091
EpitopeTransfer	Measles morbilivirus	0.000
ESM-1b	Measles morbilivirus	0.000
NPTransfer	Measles morbilivirus	0.000
EpitopeTransfer	Mononegavirales	0.286
BepiPred 3.0	Mononegavirales	-0.136
EpiDope	Mononegavirales	0.336
EpitopeVec	Mononegavirales	0.170
ESM-1b	Mononegavirales	0.321
NPTransfer	Mononegavirales	0.209
EpitopeTransfer	Orthopox	0.226
BepiPred 3.0	Orthopox	0.375
EpiDope	Orthopox	0.163
EpitopeVec	Orthopox	-0.206
ESM-1b	Orthopox	0.101
NPTransfer	Orthopox	-0.020
EpitopeTransfer	Ovolvulus	0.272
BepiPred 3.0	Ovolvulus	0.277
EpiDope	Ovolvulus	-0.055
EpitopeVec	Ovolvulus	0.064
ESM-1b	Ovolvulus	0.202
NPTransfer	Ovolvulus	0.210
EpitopeTransfer	P. aeruginosa	0.147
BepiPred 3.0	P. aeruginosa	-0.258
EpiDope	P. aeruginosa	0.137
EpitopeVec	P. aeruginosa	0.145
ESM-1b	P. aeruginosa	0.263
NPTransfer	P. aeruginosa	0.350
EpitopeTransfer	P. falciparum	0.505
BepiPred 3.0	P. falciparum	0.119
EpiDope	P. falciparum	0.088
EpitopeVec	P. falciparum	0.018

Method	Dataset	Value
ESM-1b	P. falciparum	0.453
NPTransfer	P. falciparum	0.437
EpitopeTransfer	S. mansoni	0.056
BepiPred 3.0	S. mansoni	0.126
EpiDope	S. mansoni	0.185
EpitopeVec	S. mansoni	-0.016
ESM-1b	S. mansoni	0.093
NPTransfer	S. mansoni	0.120
EpitopeTransfer	Sars-cov-2	0.043
BepiPred 3.0	Sars-cov-2	0.011
EpiDope	Sars-cov-2	0.169
EpitopeVec	Sars-cov-2	0.101
ESM-1b	Sars-cov-2	0.072
NPTransfer	Sars-cov-2	0.037
EpitopeTransfer	T. gondii	0.218
BepiPred 3.0	T. gondii	-0.070
EpiDope	T. gondii	0.092
EpitopeVec	T. gondii	0.084
ESM-1b	T. gondii	0.128
NPTransfer	T. gondii	0.108

Table 4: Comparison of methods for Balanced Accuracy

Method	Dataset	Value
EpitopeTransfer	B. pertussis	0.500
BepiPred 3.0	B. pertussis	0.497
EpiDope	B. pertussis	0.451
EpitopeVec	B. pertussis	0.694
ESM-1b	B. pertussis	0.486
NPTransfer	B. pertussis	0.500
EpitopeTransfer	C. difficile	0.647
BepiPred 3.0	C. difficile	0.496
EpiDope	C. difficile	0.500

Method	**Dataset**	**Value**
EpitopeVec	C. difficile	0.737
ESM-1b	C. difficile	0.522
NPTransfer	C. difficile	0.500
EpitopeTransfer	Corynebacterium	0.531
BepiPred 3.0	Corynebacterium	0.576
EpiDope	Corynebacterium	0.583
EpitopeVec	Corynebacterium	0.655
ESM-1b	Corynebacterium	0.433
NPTransfer	Corynebacterium	0.490
EpitopeTransfer	C. trachomatis	0.723
BepiPred 3.0	C. trachomatis	0.476
EpiDope	C. trachomatis	0.549
EpitopeVec	C. trachomatis	0.642
ESM-1b	C. trachomatis	0.682
NPTransfer	C. trachomatis	0.719
EpitopeTransfer	E. coli	0.574
BepiPred 3.0	E. coli	0.429
EpiDope	E. coli	0.603
EpitopeVec	E. coli	0.518
ESM-1b	E. coli	0.536
NPTransfer	E. coli	0.604
EpitopeTransfer	Enterobacteriaceae	0.739
BepiPred 3.0	Enterobacteriaceae	0.526
EpiDope	Enterobacteriaceae	0.536
EpitopeVec	Enterobacteriaceae	0.530
ESM-1b	Enterobacteriaceae	0.495
NPTransfer	Enterobacteriaceae	0.481
EpitopeTransfer	Filoviridae	0.947
BepiPred 3.0	Filoviridae	0.616
EpiDope	Filoviridae	0.618
EpitopeVec	Filoviridae	0.667
ESM-1b	Filoviridae	0.817
NPTransfer	Filoviridae	0.898

Method	**Dataset**	**Value**
EpitopeTransfer	Human gammaherpesvirus 4	0.600
BepiPred 3.0	Human gammaherpesvirus 4	0.415
EpiDope	Human gammaherpesvirus 4	0.523
EpitopeVec	Human gammaherpesvirus 4	0.535
ESM-1b	Human gammaherpesvirus 4	0.590
NPTransfer	Human gammaherpesvirus 4	0.588
EpitopeTransfer	Influenza A	0.579
BepiPred 3.0	Influenza A	0.629
EpiDope	Influenza A	0.519
EpitopeVec	Influenza A	0.566
ESM-1b	Influenza A	0.483
NPTransfer	Influenza A	0.541
EpitopeTransfer	Lentivirus	0.844
BepiPred 3.0	Lentivirus	0.662
EpiDope	Lentivirus	0.516
EpitopeVec	Lentivirus	0.535
ESM-1b	Lentivirus	0.794
NPTransfer	Lentivirus	0.675
EpitopeTransfer	M. tuberculosis	0.486
BepiPred 3.0	M. tuberculosis	0.511
EpiDope	M. tuberculosis	0.509
EpitopeVec	M. tuberculosis	0.496
ESM-1b	M. tuberculosis	0.515
NPTransfer	M. tuberculosis	0.475
BepiPred 3.0	Measles morbilivirus	0.346
EpiDope	Measles morbilivirus	0.533
EpitopeVec	Measles morbilivirus	0.543
EpitopeTransfer	Measles morbilivirus	0.500
ESM-1b	Measles morbilivirus	0.500
NPTransfer	Measles morbilivirus	0.500
EpitopeTransfer	Mononegavirales	0.651
BepiPred 3.0	Mononegavirales	0.430
EpiDope	Mononegavirales	0.646

Method	Dataset	Value
EpitopeVec	Mononegavirales	0.589
ESM-1b	Mononegavirales	0.667
NPTransfer	Mononegavirales	0.581
EpitopeTransfer	Orthopox	0.631
BepiPred 3.0	Orthopox	0.699
EpiDope	Orthopox	0.605
EpitopeVec	Orthopox	0.365
ESM-1b	Orthopox	0.562
NPTransfer	Orthopox	0.488
EpitopeTransfer	Ovolvulus	0.620
BepiPred 3.0	Ovolvulus	0.687
EpiDope	Ovolvulus	0.480
EpitopeVec	Ovolvulus	0.545
ESM-1b	Ovolvulus	0.568
NPTransfer	Ovolvulus	0.647
EpitopeTransfer	P. aeruginosa	0.515
BepiPred 3.0	P. aeruginosa	0.455
EpiDope	P. aeruginosa	0.531
EpitopeVec	P. aeruginosa	0.579
ESM-1b	P. aeruginosa	0.645
NPTransfer	P. aeruginosa	0.680
EpitopeTransfer	P. falciparum	0.743
BepiPred 3.0	P. falciparum	0.550
EpiDope	P. falciparum	0.541
EpitopeVec	P. falciparum	0.509
ESM-1b	P. falciparum	0.711
NPTransfer	P. falciparum	0.721
EpitopeTransfer	S. mansoni	0.525
BepiPred 3.0	S. mansoni	0.562
EpiDope	S. mansoni	0.581
EpitopeVec	S. mansoni	0.491
ESM-1b	S. mansoni	0.544
NPTransfer	S. mansoni	0.538

Method	**Dataset**	**Value**
EpitopeTransfer	Sars-cov-2	0.518
BepiPred 3.0	Sars-cov-2	0.507
EpiDope	Sars-cov-2	0.592
EpitopeVec	Sars-cov-2	0.584
ESM-1b	Sars-cov-2	0.535
NPTransfer	Sars-cov-2	0.516
EpitopeTransfer	T. gondii	0.582
BepiPred 3.0	T. gondii	0.468
EpiDope	T. gondii	0.537
EpitopeVec	T. gondii	0.544
ESM-1b	T. gondii	0.539
NPTransfer	T. gondii	0.554

Table 5: Comparison of methods for PPV

Method	**Dataset**	**Value**
EpitopeTransfer	B. pertussis	0.718
BepiPred 3.0	B. pertussis	0.717
EpiDope	B. pertussis	0.625
EpitopeVec	B. pertussis	0.857
ESM-1b	B. pertussis	0.712
NPTransfer	B. pertussis	0.718
EpitopeTransfer	C. difficile	0.138
BepiPred 3.0	C. difficile	0.000
EpiDope	C. difficile	0.000
EpitopeVec	C. difficile	0.164
ESM-1b	C. difficile	0.097
NPTransfer	C. difficile	0.093
EpitopeTransfer	Corynebacterium	0.500
BepiPred 3.0	Corynebacterium	0.833
EpiDope	Corynebacterium	1.000
EpitopeVec	Corynebacterium	0.603
ESM-1b	Corynebacterium	0.415

Method	Dataset	Value
NPTransfer	Corynebacterium	0.457
EpitopeTransfer	C. trachomatis	0.744
BepiPred 3.0	C. trachomatis	0.491
EpiDope	C. trachomatis	0.667
EpitopeVec	C. trachomatis	0.667
ESM-1b	C. trachomatis	0.780
NPTransfer	C. trachomatis	0.742
EpitopeTransfer	E. coli	0.776
BepiPred 3.0	E. coli	0.656
EpiDope	E. coli	1.000
EpitopeVec	E. coli	0.759
ESM-1b	E. coli	0.760
NPTransfer	E. coli	0.789
EpitopeTransfer	Enterobacteriaceae	0.691
BepiPred 3.0	Enterobacteriaceae	0.507
EpiDope	Enterobacteriaceae	0.733
EpitopeVec	Enterobacteriaceae	0.497
ESM-1b	Enterobacteriaceae	0.467
NPTransfer	Enterobacteriaceae	0.460
EpitopeTransfer	Filoviridae	0.663
BepiPred 3.0	Filoviridae	0.138
EpiDope	Filoviridae	0.321
EpitopeVec	Filoviridae	0.164
ESM-1b	Filoviridae	0.526
NPTransfer	Filoviridae	0.662
EpitopeTransfer	Human gammaherpesvirus 4	0.680
BepiPred 3.0	Human gammaherpesvirus 4	0.326
EpiDope	Human gammaherpesvirus 4	0.542
EpitopeVec	Human gammaherpesvirus 4	0.491
ESM-1b	Human gammaherpesvirus 4	0.746
NPTransfer	Human gammaherpesvirus 4	0.647
EpitopeTransfer	Influenza A	0.787
BepiPred 3.0	Influenza A	0.901

Method	Dataset	Value
EpiDope	Influenza A	0.817
EpitopeVec	Influenza A	0.782
ESM-1b	Influenza A	0.743
NPTransfer	Influenza A	0.768
EpitopeTransfer	Lentivirus	0.860
BepiPred 3.0	Lentivirus	1.000
EpiDope	Lentivirus	0.680
EpitopeVec	Lentivirus	0.692
ESM-1b	Lentivirus	0.844
NPTransfer	Lentivirus	0.750
EpitopeTransfer	M. tuberculosis	0.509
BepiPred 3.0	M. tuberculosis	0.553
EpiDope	M. tuberculosis	0.574
EpitopeVec	M. tuberculosis	0.515
ESM-1b	M. tuberculosis	0.528
NPTransfer	M. tuberculosis	0.501
BepiPred 3.0	Measles morbilivirus	0.293
EpiDope	Measles morbilivirus	0.542
EpitopeVec	Measles morbilivirus	0.502
EpitopeTransfer	Measles morbilivirus	0.000
ESM-1b	Measles morbilivirus	0.000
NPTransfer	Measles morbilivirus	0.000
EpitopeTransfer	Mononegavirales	0.481
BepiPred 3.0	Mononegavirales	0.259
EpiDope	Mononegavirales	0.646
EpitopeVec	Mononegavirales	0.427
ESM-1b	Mononegavirales	0.470
NPTransfer	Mononegavirales	0.385
EpitopeTransfer	Orthopox	0.318
BepiPred 3.0	Orthopox	0.456
EpiDope	Orthopox	0.228
EpitopeVec	Orthopox	0.094
ESM-1b	Orthopox	0.233

Method	Dataset	Value
NPTransfer	Orthopox	0.171
EpitopeTransfer	Ovolvulus	0.423
BepiPred 3.0	Ovolvulus	0.225
EpiDope	Ovolvulus	0.080
EpitopeVec	Ovolvulus	0.176
ESM-1b	Ovolvulus	0.432
NPTransfer	Ovolvulus	0.227
EpitopeTransfer	P. aeruginosa	0.717
BepiPred 3.0	P. aeruginosa	0.000
EpiDope	P. aeruginosa	1.000
EpitopeVec	P. aeruginosa	0.765
ESM-1b	P. aeruginosa	0.828
NPTransfer	P. aeruginosa	0.944
EpitopeTransfer	P. falciparum	0.793
BepiPred 3.0	P. falciparum	0.736
EpiDope	P. falciparum	0.693
EpitopeVec	P. falciparum	0.631
ESM-1b	P. falciparum	0.763
NPTransfer	P. falciparum	0.795
EpitopeTransfer	S. mansoni	0.297
BepiPred 3.0	S. mansoni	0.379
EpiDope	S. mansoni	0.454
EpitopeVec	S. mansoni	0.274
ESM-1b	S. mansoni	0.358
NPTransfer	S. mansoni	0.455
EpitopeTransfer	Sars-cov-2	0.151
BepiPred 3.0	Sars-cov-2	0.110
EpiDope	Sars-cov-2	0.238
EpitopeVec	Sars-cov-2	0.135
ESM-1b	Sars-cov-2	0.170
NPTransfer	Sars-cov-2	0.142
EpitopeTransfer	T. gondii	0.730
BepiPred 3.0	T. gondii	0.671

Method	Dataset	Value
EpiDope	T. gondii	0.787
EpitopeVec	T. gondii	0.720
ESM-1b	T. gondii	0.708
NPTransfer	T. gondii	0.724

Table 6: Comparison of methods for NPV

Method	Dataset	Value
EpitopeTransfer	B. pertussis	0.000
BepiPred 3.0	B. pertussis	0.273
EpiDope	B. pertussis	0.256
EpitopeVec	B. pertussis	0.462
ESM-1b	B. pertussis	0.000
NPTransfer	B. pertussis	0.000
EpitopeTransfer	C. difficile	0.964
BepiPred 3.0	C. difficile	0.906
EpiDope	C. difficile	0.907
EpitopeVec	C. difficile	1.000
ESM-1b	C. difficile	0.964
NPTransfer	C. difficile	0.000
EpitopeTransfer	Corynebacterium	0.565
BepiPred 3.0	Corynebacterium	0.569
EpiDope	Corynebacterium	0.571
EpitopeVec	Corynebacterium	0.717
ESM-1b	Corynebacterium	0.449
NPTransfer	Corynebacterium	0.519
EpitopeTransfer	C. trachomatis	0.704
BepiPred 3.0	C. trachomatis	0.446
EpiDope	C. trachomatis	0.522
EpitopeVec	C. trachomatis	0.621
ESM-1b	C. trachomatis	0.629
NPTransfer	C. trachomatis	0.699
EpitopeTransfer	E. coli	0.941

Method	**Dataset**	**Value**
BepiPred 3.0	E. coli	0.215
EpiDope	E. coli	0.299
EpitopeVec	E. coli	0.268
ESM-1b	E. coli	0.889
NPTransfer	E. coli	0.800
EpitopeTransfer	Enterobacteriaceae	0.789
BepiPred 3.0	Enterobacteriaceae	0.550
EpiDope	Enterobacteriaceae	0.550
EpitopeVec	Enterobacteriaceae	0.563
ESM-1b	Enterobacteriaceae	0.409
NPTransfer	Enterobacteriaceae	0.212
EpitopeTransfer	Filoviridae	0.994
BepiPred 3.0	Filoviridae	0.949
EpiDope	Filoviridae	0.922
EpitopeVec	Filoviridae	0.958
ESM-1b	Filoviridae	0.965
NPTransfer	Filoviridae	0.982
EpitopeTransfer	Human gammaherpesvirus 4	0.609
BepiPred 3.0	Human gammaherpesvirus 4	0.483
EpiDope	Human gammaherpesvirus 4	0.559
EpitopeVec	Human gammaherpesvirus 4	0.579
ESM-1b	Human gammaherpesvirus 4	0.598
NPTransfer	Human gammaherpesvirus 4	0.601
EpitopeTransfer	Influenza A	0.408
BepiPred 3.0	Influenza A	0.321
EpiDope	Influenza A	0.258
EpitopeVec	Influenza A	0.364
ESM-1b	Influenza A	0.157
NPTransfer	Influenza A	0.350
EpitopeTransfer	Lentivirus	1.000
BepiPred 3.0	Lentivirus	0.435
EpiDope	Lentivirus	0.353
EpitopeVec	Lentivirus	0.372

Method	**Dataset**	**Value**
ESM-1b	Lentivirus	0.800
NPTransfer	Lentivirus	0.879
EpitopeTransfer	M. tuberculosis	0.458
BepiPred 3.0	M. tuberculosis	0.488
EpiDope	M. tuberculosis	0.486
EpitopeVec	M. tuberculosis	0.477
ESM-1b	M. tuberculosis	0.524
NPTransfer	M. tuberculosis	0.438
BepiPred 3.0	Measles morbilivirus	0.395
EpiDope	Measles morbilivirus	0.552
EpitopeVec	Measles morbilivirus	0.593
EpitopeTransfer	Measles morbilivirus	0.530
ESM-1b	Measles morbilivirus	0.530
NPTransfer	Measles morbilivirus	0.530
EpitopeTransfer	Mononegavirales	0.790
BepiPred 3.0	Mononegavirales	0.608
EpiDope	Mononegavirales	0.742
EpitopeVec	Mononegavirales	0.734
ESM-1b	Mononegavirales	0.839
NPTransfer	Mononegavirales	0.885
EpitopeTransfer	Orthopox	0.876
BepiPred 3.0	Orthopox	0.897
EpiDope	Orthopox	0.898
EpitopeVec	Orthopox	0.749
ESM-1b	Orthopox	0.850
NPTransfer	Orthopox	0.812
EpitopeTransfer	Ovolvulus	0.885
BepiPred 3.0	Ovolvulus	0.980
EpiDope	Ovolvulus	0.845
EpitopeVec	Ovolvulus	0.870
ESM-1b	Ovolvulus	0.868
NPTransfer	Ovolvulus	0.923
EpitopeTransfer	P. aeruginosa	1.000

Method	**Dataset**	**Value**
BepiPred 3.0	P. aeruginosa	0.270
EpiDope	P. aeruginosa	0.303
EpitopeVec	P. aeruginosa	0.370
ESM-1b	P. aeruginosa	0.411
NPTransfer	P. aeruginosa	0.397
EpitopeTransfer	P. falciparum	0.732
BepiPred 3.0	P. falciparum	0.405
EpiDope	P. falciparum	0.402
EpitopeVec	P. falciparum	0.387
ESM-1b	P. falciparum	0.723
NPTransfer	P. falciparum	0.639
EpitopeTransfer	S. mansoni	0.765
BepiPred 3.0	S. mansoni	0.750
EpiDope	S. mansoni	0.757
EpitopeVec	S. mansoni	0.710
ESM-1b	S. mansoni	0.739
NPTransfer	S. mansoni	0.733
EpitopeTransfer	Sars-cov-2	0.901
BepiPred 3.0	Sars-cov-2	0.899
EpiDope	Sars-cov-2	0.917
EpitopeVec	Sars-cov-2	0.927
ESM-1b	Sars-cov-2	0.905
NPTransfer	Sars-cov-2	0.901
EpitopeTransfer	T. gondii	0.561
BepiPred 3.0	T. gondii	0.254
EpiDope	T. gondii	0.329
EpitopeVec	T. gondii	0.360
ESM-1b	T. gondii	0.500
NPTransfer	T. gondii	0.383

Table 7: Comparison of methods for Sensitivity

Method	Dataset	Value
EpitopeTransfer	B. pertussis	1.000
BepiPred 3.0	B. pertussis	0.850
EpiDope	B. pertussis	0.187
EpitopeVec	B. pertussis	0.673
ESM-1b	B. pertussis	0.972
NPTransfer	B. pertussis	1.000
EpitopeTransfer	C. difficile	0.829
BepiPred 3.0	C. difficile	0.000
EpiDope	C. difficile	0.000
EpitopeVec	C. difficile	1.000
ESM-1b	C. difficile	0.976
NPTransfer	C. difficile	1.000
EpitopeTransfer	Corynebacterium	0.630
BepiPred 3.0	Corynebacterium	0.185
EpiDope	Corynebacterium	0.167
EpitopeVec	Corynebacterium	0.759
ESM-1b	Corynebacterium	0.500
NPTransfer	Corynebacterium	0.296
EpitopeTransfer	C. trachomatis	0.691
BepiPred 3.0	C. trachomatis	0.719
EpiDope	C. trachomatis	0.203
EpitopeVec	C. trachomatis	0.586
ESM-1b	C. trachomatis	0.512
NPTransfer	C. trachomatis	0.684
EpitopeTransfer	E. coli	0.997
BepiPred 3.0	E. coli	0.263
EpiDope	E. coli	0.205
EpitopeVec	E. coli	0.535
ESM-1b	E. coli	0.997
NPTransfer	E. coli	0.981
EpitopeTransfer	Enterobacteriaceae	0.793
BepiPred 3.0	Enterobacteriaceae	0.380

Method	**Dataset**	**Value**
EpiDope	Enterobacteriaceae	0.108
EpitopeVec	Enterobacteriaceae	0.575
ESM-1b	Enterobacteriaceae	0.975
NPTransfer	Enterobacteriaceae	0.949
EpitopeTransfer	Filoviridae	0.948
BepiPred 3.0	Filoviridae	0.793
EpiDope	Filoviridae	0.310
EpitopeVec	Filoviridae	0.793
ESM-1b	Filoviridae	0.707
NPTransfer	Filoviridae	0.845
EpitopeTransfer	Human gammaherpesvirus 4	0.330
BepiPred 3.0	Human gammaherpesvirus 4	0.235
EpiDope	Human gammaherpesvirus 4	0.158
EpitopeVec	Human gammaherpesvirus 4	0.526
ESM-1b	Human gammaherpesvirus 4	0.252
NPTransfer	Human gammaherpesvirus 4	0.321
EpitopeTransfer	Influenza A	0.852
BepiPred 3.0	Influenza A	0.384
EpiDope	Influenza A	0.118
EpitopeVec	Influenza A	0.815
ESM-1b	Influenza A	0.924
NPTransfer	Influenza A	0.866
EpitopeTransfer	Lentivirus	1.000
BepiPred 3.0	Lentivirus	0.324
EpiDope	Lentivirus	0.345
EpitopeVec	Lentivirus	0.486
ESM-1b	Lentivirus	0.912
NPTransfer	Lentivirus	0.973
EpitopeTransfer	M. tuberculosis	0.691
BepiPred 3.0	M. tuberculosis	0.165
EpiDope	M. tuberculosis	0.090
EpitopeVec	M. tuberculosis	0.505
ESM-1b	M. tuberculosis	0.838

Method	Dataset	Value
NPTransfer	M. tuberculosis	0.682
BepiPred 3.0	Measles morbilivirus	0.271
EpiDope	Measles morbilivirus	0.265
EpitopeVec	Measles morbilivirus	0.706
EpitopeTransfer	Measles morbilivirus	0.000
ESM-1b	Measles morbilivirus	0.000
NPTransfer	Measles morbilivirus	0.000
EpitopeTransfer	Mononegavirales	0.685
BepiPred 3.0	Mononegavirales	0.285
EpiDope	Mononegavirales	0.407
EpitopeVec	Mononegavirales	0.587
ESM-1b	Mononegavirales	0.804
NPTransfer	Mononegavirales	0.946
EpitopeTransfer	Orthopox	0.483
BepiPred 3.0	Orthopox	0.534
EpiDope	Orthopox	0.759
EpitopeVec	Orthopox	0.259
ESM-1b	Orthopox	0.414
NPTransfer	Orthopox	0.690
EpitopeTransfer	Ovolvulus	0.317
BepiPred 3.0	Ovolvulus	0.952
EpiDope	Ovolvulus	0.039
EpitopeVec	Ovolvulus	0.513
ESM-1b	Ovolvulus	0.176
NPTransfer	Ovolvulus	0.734
EpitopeTransfer	P. aeruginosa	1.000
BepiPred 3.0	P. aeruginosa	0.000
EpiDope	P. aeruginosa	0.062
EpitopeVec	P. aeruginosa	0.642
ESM-1b	P. aeruginosa	0.593
NPTransfer	P. aeruginosa	0.420
EpitopeTransfer	P. falciparum	0.863
BepiPred 3.0	P. falciparum	0.249

Method	**Dataset**	**Value**
EpiDope	P. falciparum	0.313
EpitopeVec	P. falciparum	0.654
ESM-1b	P. falciparum	0.873
NPTransfer	P. falciparum	0.773
EpitopeTransfer	S. mansoni	0.826
BepiPred 3.0	S. mansoni	0.356
EpiDope	S. mansoni	0.312
EpitopeVec	S. mansoni	0.321
ESM-1b	S. mansoni	0.305
NPTransfer	S. mansoni	0.145
EpitopeTransfer	Sars-cov-2	0.099
BepiPred 3.0	Sars-cov-2	0.178
EpiDope	Sars-cov-2	0.290
EpitopeVec	Sars-cov-2	0.620
ESM-1b	Sars-cov-2	0.159
NPTransfer	Sars-cov-2	0.103
EpitopeTransfer	T. gondii	0.911
BepiPred 3.0	T. gondii	0.738
EpiDope	T. gondii	0.183
EpitopeVec	T. gondii	0.649
ESM-1b	T. gondii	0.936
NPTransfer	T. gondii	0.713

Table 8: Comparison of methods for Specificity

Method	**Dataset**	**Value**
EpitopeTransfer	B. pertussis	0.000
BepiPred 3.0	B. pertussis	0.143
EpiDope	B. pertussis	0.714
EpitopeVec	B. pertussis	0.714
ESM-1b	B. pertussis	0.000
NPTransfer	B. pertussis	0.000
EpitopeTransfer	C. difficile	0.465

Method	**Dataset**	**Value**
BepiPred 3.0	C. difficile	0.992
EpiDope	C. difficile	1.000
EpitopeVec	C. difficile	0.475
ESM-1b	C. difficile	0.068
NPTransfer	C. difficile	0.000
EpitopeTransfer	Corynebacterium	0.433
BepiPred 3.0	Corynebacterium	0.967
EpiDope	Corynebacterium	1.000
EpitopeVec	Corynebacterium	0.550
ESM-1b	Corynebacterium	0.367
NPTransfer	Corynebacterium	0.683
EpitopeTransfer	C. trachomatis	0.755
BepiPred 3.0	C. trachomatis	0.233
EpiDope	C. trachomatis	0.896
EpitopeVec	C. trachomatis	0.699
ESM-1b	C. trachomatis	0.851
NPTransfer	C. trachomatis	0.755
EpitopeTransfer	E. coli	0.151
BepiPred 3.0	E. coli	0.594
EpiDope	E. coli	1.000
EpitopeVec	E. coli	0.500
ESM-1b	E. coli	0.075
NPTransfer	E. coli	0.226
EpitopeTransfer	Enterobacteriaceae	0.686
BepiPred 3.0	Enterobacteriaceae	0.672
EpiDope	Enterobacteriaceae	0.965
EpitopeVec	Enterobacteriaceae	0.485
ESM-1b	Enterobacteriaceae	0.016
NPTransfer	Enterobacteriaceae	0.012
EpitopeTransfer	Filoviridae	0.945
BepiPred 3.0	Filoviridae	0.439
EpiDope	Filoviridae	0.926
EpitopeVec	Filoviridae	0.541

Method	Dataset	Value
ESM-1b	Filoviridae	0.928
NPTransfer	Filoviridae	0.951
EpitopeTransfer	Human gammaherpesvirus 4	0.871
BepiPred 3.0	Human gammaherpesvirus 4	0.595
EpiDope	Human gammaherpesvirus 4	0.889
EpitopeVec	Human gammaherpesvirus 4	0.545
ESM-1b	Human gammaherpesvirus 4	0.928
NPTransfer	Human gammaherpesvirus 4	0.854
EpitopeTransfer	Influenza A	0.307
BepiPred 3.0	Influenza A	0.873
EpiDope	Influenza A	0.921
EpitopeVec	Influenza A	0.317
ESM-1b	Influenza A	0.042
NPTransfer	Influenza A	0.217
EpitopeTransfer	Lentivirus	0.688
BepiPred 3.0	Lentivirus	1.000
EpiDope	Lentivirus	0.688
EpitopeVec	Lentivirus	0.584
ESM-1b	Lentivirus	0.675
NPTransfer	Lentivirus	0.377
EpitopeTransfer	M. tuberculosis	0.281
BepiPred 3.0	M. tuberculosis	0.856
EpiDope	M. tuberculosis	0.928
EpitopeVec	M. tuberculosis	0.487
ESM-1b	M. tuberculosis	0.192
NPTransfer	M. tuberculosis	0.267
BepiPred 3.0	Measles morbilivirus	0.422
EpiDope	Measles morbilivirus	0.802
EpitopeVec	Measles morbilivirus	0.380
EpitopeTransfer	Measles morbilivirus	1.000
ESM-1b	Measles morbilivirus	1.000
NPTransfer	Measles morbilivirus	1.000
EpitopeTransfer	Mononegavirales	0.617

Method	**Dataset**	**Value**
BepiPred 3.0	Mononegavirales	0.576
EpiDope	Mononegavirales	0.884
EpitopeVec	Mononegavirales	0.592
ESM-1b	Mononegavirales	0.529
NPTransfer	Mononegavirales	0.215
EpitopeTransfer	Orthopox	0.779
BepiPred 3.0	Orthopox	0.864
EpiDope	Orthopox	0.452
EpitopeVec	Orthopox	0.471
ESM-1b	Orthopox	0.710
NPTransfer	Orthopox	0.287
EpitopeTransfer	Ovolvulus	0.924
BepiPred 3.0	Ovolvulus	0.422
EpiDope	Ovolvulus	0.920
EpitopeVec	Ovolvulus	0.577
ESM-1b	Ovolvulus	0.959
NPTransfer	Ovolvulus	0.560
EpitopeTransfer	P. aeruginosa	0.030
BepiPred 3.0	P. aeruginosa	0.909
EpiDope	P. aeruginosa	1.000
EpitopeVec	P. aeruginosa	0.515
ESM-1b	P. aeruginosa	0.697
NPTransfer	P. aeruginosa	0.939
EpitopeTransfer	P. falciparum	0.624
BepiPred 3.0	P. falciparum	0.851
EpiDope	P. falciparum	0.769
EpitopeVec	P. falciparum	0.363
ESM-1b	P. falciparum	0.549
NPTransfer	P. falciparum	0.669
EpitopeTransfer	S. mansoni	0.224
BepiPred 3.0	S. mansoni	0.768
EpiDope	S. mansoni	0.851
EpitopeVec	S. mansoni	0.662

Method	Dataset	Value
ESM-1b	S. mansoni	0.783
NPTransfer	S. mansoni	0.931
EpitopeTransfer	Sars-cov-2	0.937
BepiPred 3.0	Sars-cov-2	0.836
EpiDope	Sars-cov-2	0.894
EpitopeVec	Sars-cov-2	0.547
ESM-1b	Sars-cov-2	0.912
NPTransfer	Sars-cov-2	0.929
EpitopeTransfer	T. gondii	0.253
BepiPred 3.0	T. gondii	0.198
EpiDope	T. gondii	0.890
EpitopeVec	T. gondii	0.440
ESM-1b	T. gondii	0.143
NPTransfer	T. gondii	0.396

Metric	EpitopeTrans	BepiPred 3.0	EpiDope	EpitopeVec	ESM-1b	NPTransfer
AUC	0.690 (±0.029)	0.503 (±0.035)	0.634 (±0.032)	0.602 (±0.027)	0.656 (±0.030)	0.642 (±0.032)
F1	0.592 (±0.060)	0.363 (±0.045)	0.276 (±0.029)	0.509 (±0.044)	0.542 (±0.060)	0.529 (±0.061)
MCC	0.258 (±0.052)	0.041 (±0.044)	0.118 (±0.025)	0.112 (±0.029)	0.172 (±0.047)	0.177 (±0.049)
B. ACC	0.623 (±0.028)	0.527 (±0.021)	0.548 (±0.011)	0.566 (±0.019)	0.581 (±0.023)	0.585 (±0.025)
PPV	0.549 (±0.056)	0.462 (±0.066)	0.581 (±0.065)	0.496 (±0.055)	0.529 (±0.058)	0.522 (±0.062)
NPV	0.724 (±0.057)	0.555 (±0.057)	0.571 (±0.054)	0.604 (±0.050)	0.638 (±0.060)	0.584 (±0.066)
Sensit.	0.697 (±0.068)	0.393 (±0.062)	0.226 (±0.037)	0.610 (±0.037)	0.641 (±0.073)	0.656 (±0.073)
Specif.	0.549 (±0.072)	0.660 (±0.061)	0.869 (±0.030)	0.522 (±0.023)	0.521 (±0.083)	0.513 (±0.079)

Table 9: Summary of average test set performance (*mean ±standard error*) for EpitopeTransfer (proposed method) and five baseline methods across 20 selected datasets. Each row corresponds to a performance evaluation metric, and the values indicate the mean performance of each method over all datasets.

Appendix D

Statistical comparisons of median values for each performance metric were performed to assess the significance of differences between EpitopeTransfer (ESM-1b) and the baseline methods. The Wilcoxon signed rank test was used as the primary statistical method to evaluate whether observed differences in medians were statistically meaningful. To account for multiple comparisons, the p-values derived from the tests were adjusted for the false discovery rate using the Benjamini-Hochberg correction.

The analysis includes the following columns: "Pair", which specifies the pairwise comparison (e.g., EpitopeTransfer vs. Baseline); "Medians of diff", representing the median of paired differences (95% CI); "p-value", which indicates the unadjusted significance level from the Wilcoxon test; "FDR", which represents the adjusted p-value following the Benjamini-Hochberg procedure; and "Significant", which highlights whether the corrected p-value falls below the significance threshold of 0.05.

Comparison Results for AUC

Metric	**Pair**	**Medians of diff.**	**p-value**	**FDR**	**Signific**
AUC	EpitopeTransfer vs BepiPred 3	0.193 (0.088, 0.276)	0.00097	0.00483	Yes
AUC	EpitopeTransfer vs EpiDope	0.054 (-0.018, 0.120)	0.12319	0.12319	No
AUC	EpitopeTransfer vs EpitopeVec	0.091 (0.014, 0.178)	0.03234	0.04043	Yes
AUC	EpitopeTransfer vs ESM-1b	0.028 (0.004, 0.060)	0.02299	0.03831	Yes
AUC	EpitopeTransfer vs NPTransfer	0.061 (0.015, 0.083)	0.00392	0.00979	Yes

Table 10: Comparison Results for AUC

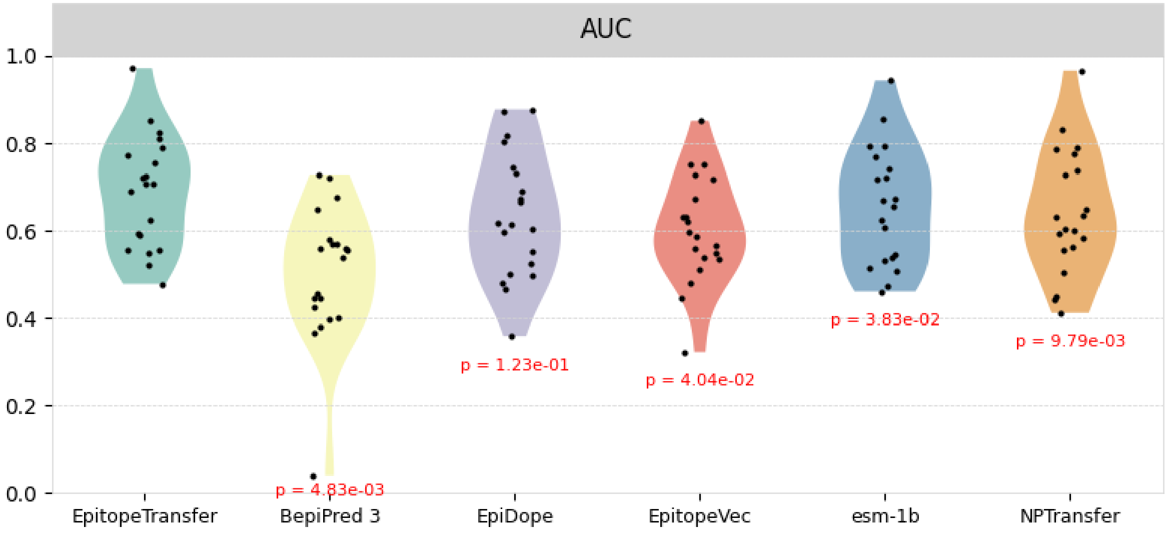


Figure 2: Performance plot for the AUC metric

Comparison Results for Balanced Accuracy

Metric	Pair	Medians of diff.	p-value	FDR	Signific
BACC	EpitopeTransfer vs BepiPred 3	0.088 (0.035, 0.165)	0.01407	0.03402	Yes
BACC	EpitopeTransfer vs EpiDope	0.073 (0.011, 0.140)	0.02041	0.03402	Yes
BACC	EpitopeTransfer vs EpitopeVec	0.057 (-0.013, 0.140)	0.12319	0.12319	No
BACC	EpitopeTransfer vs esm-1b	0.041 (0.011, 0.074)	0.01597	0.03402	Yes
BACC	EpitopeTransfer vs NPTransfer	0.030 (0.002, 0.087)	0.03285	0.04106	Yes

Table 11: Comparison Results for Balanced Accuracy

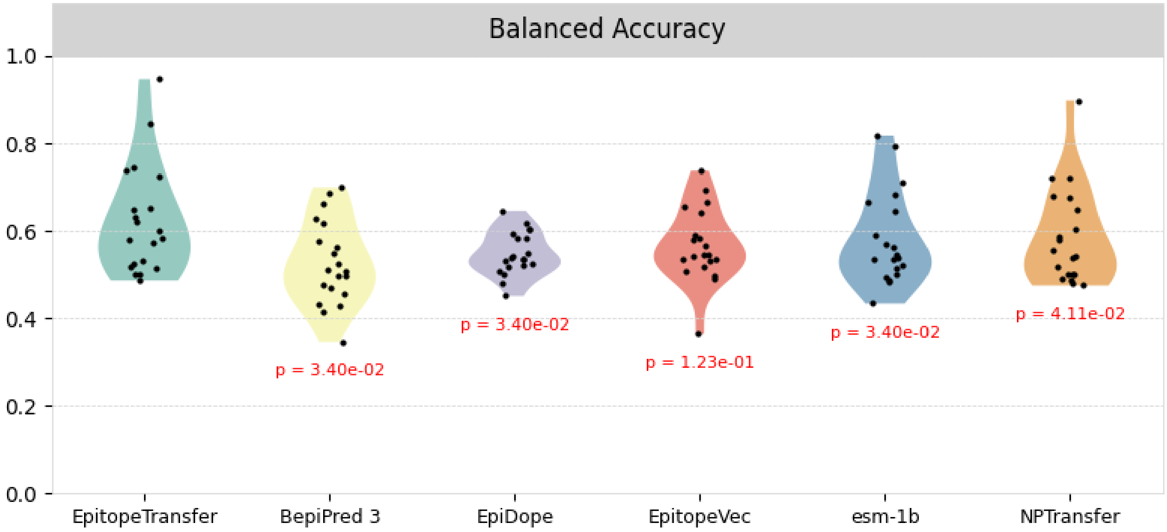


Figure 3: Performance plot for the Balanced Accuracy metric

Comparison Results for F1

Metric	Pair	Medians of diff.	p-value	FDR	Signific
F1	EpitopeTransfer vs BepiPred 3	0.247 (0.139, 0.364)	0.00013	0.00031	Yes
F1	EpitopeTransfer vs EpiDope	0.369 (0.236, 0.477)	0.00003	0.00013	Yes
F1	EpitopeTransfer vs EpitopeVec	0.110 (0.041, 0.187)	0.00284	0.00334	Yes
F1	EpitopeTransfer vs esm-1b	0.055 (0.020, 0.087)	0.00334	0.00334	Yes
F1	EpitopeTransfer vs NPTransfer	0.058 (0.021, 0.114)	0.00030	0.00050	Yes

Table 12: Comparison Results for F1

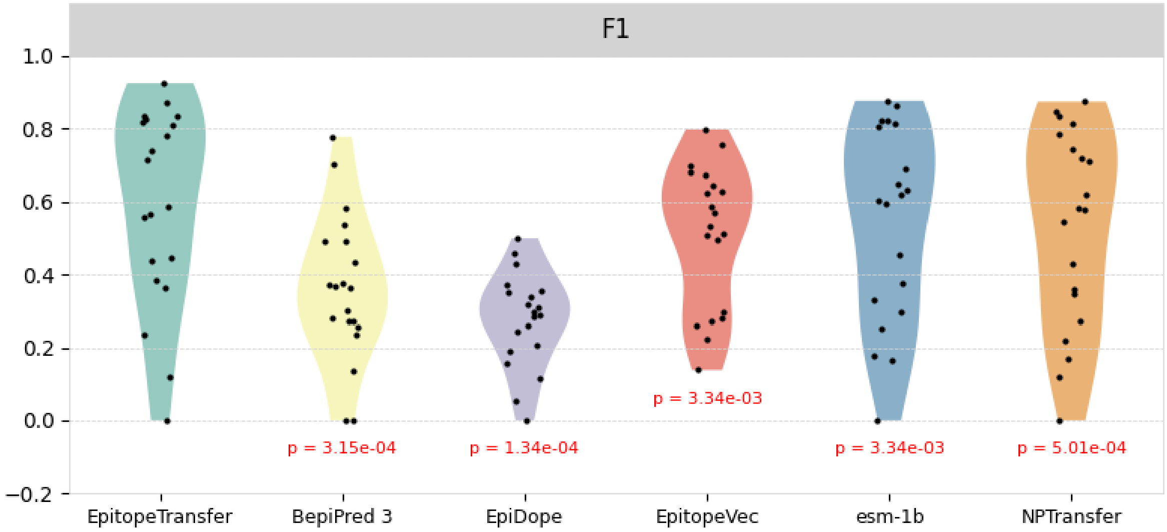


Figure 4: Performance plot for the F1 metric

Comparison Results for MCC

Metric	Pair	Medians of diff.	p-value	FDR	Signific
MCC	EpitopeTransfer vs BepiPred 3	0.204 (0.105, 0.390)	0.00823	0.01698	Yes
MCC	EpitopeTransfer vs EpiDope	0.134 (0.023, 0.254)	0.02041	0.02041	Yes
MCC	EpitopeTransfer vs EpitopeVec	0.150 (0.027, 0.296)	0.02041	0.02041	Yes
MCC	EpitopeTransfer vs ESM-1b	0.082 (0.025, 0.143)	0.00618	0.01698	Yes
MCC	EpitopeTransfer vs NPTransfer	0.065 (0.019, 0.154)	0.01019	0.01698	Yes

Table 13: Comparison Results for MCC

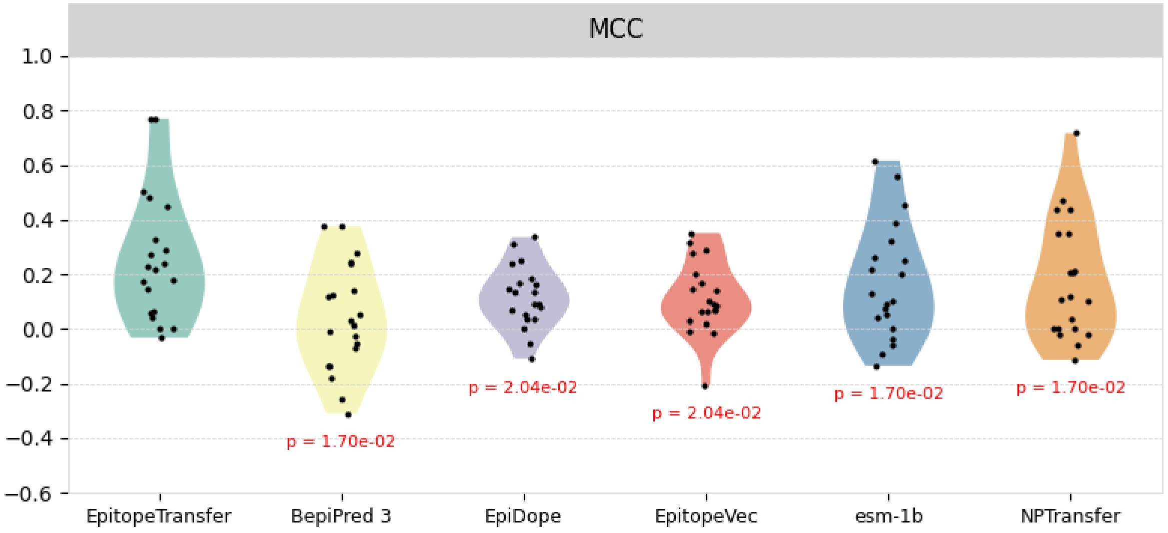


Figure 5: Performance plot for the MCC metric

Comparison Results for NPV

Metric	Pair	Medians of diff.	p-value	FDR	Signific
NPV	EpitopeTransfer vs BepiPred 3	0.140 (0.027, 0.307)	0.01236	0.01545	Yes
NPV	EpitopeTransfer vs EpiDope	0.116 (0.024, 0.315)	0.00533	0.01316	Yes
NPV	EpitopeTransfer vs EpitopeVec	0.087 (0.010, 0.301)	0.02299	0.02299	Yes
NPV	EpitopeTransfer vs esm-1b	0.050 (0.012, 0.158)	0.00789	0.01316	Yes
NPV	EpitopeTransfer vs NPTransfer	0.070 (0.020, 0.304)	0.00405	0.01316	Yes

Table 14: Comparison Results for NPV

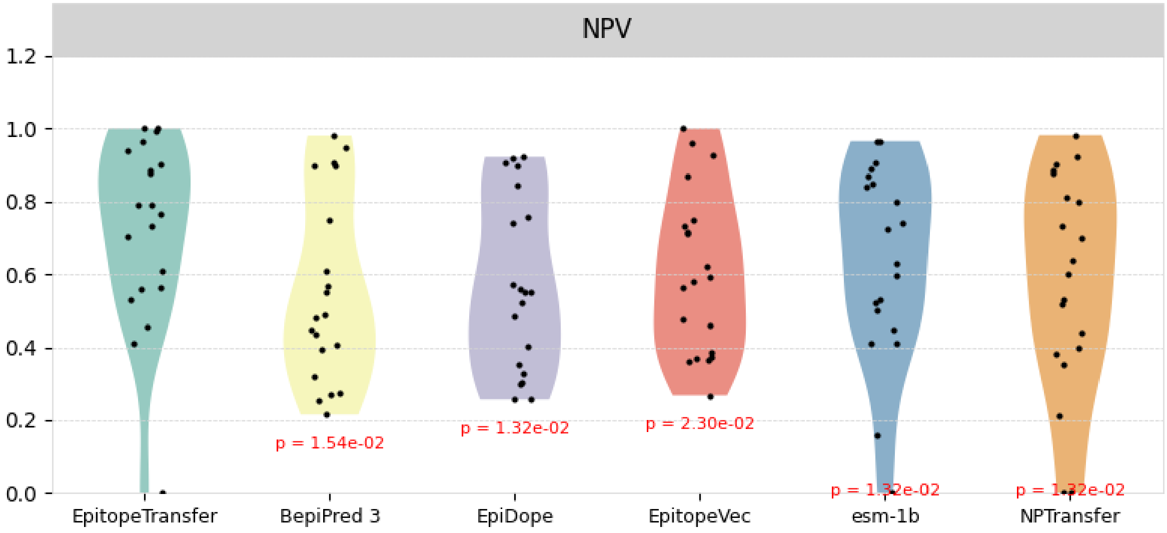


Figure 6: Performance plot for the NPV metric

Comparison Results for PPV

Metric	Pair	Medians of diff.	p-value	FDR	Signific
PPV	EpitopeTransfer vs BepiPred 3	0.089 (-0.012, 0.206)	0.09551	0.15919	No
PPV	EpitopeTransfer vs EpiDope	0.007 (-0.097, 0.097)	0.89057	0.89057	No
PPV	EpitopeTransfer vs EpitopeVec	0.075 (0.009, 0.133)	0.02582	0.12911	No
PPV	EpitopeTransfer vs esm-1b	0.014 (-0.014, 0.050)	0.31241	0.39051	No
PPV	EpitopeTransfer vs NPTransfer	0.025 (-0.002, 0.090)	0.06111	0.15279	No

Table 15: Comparison Results for PPV

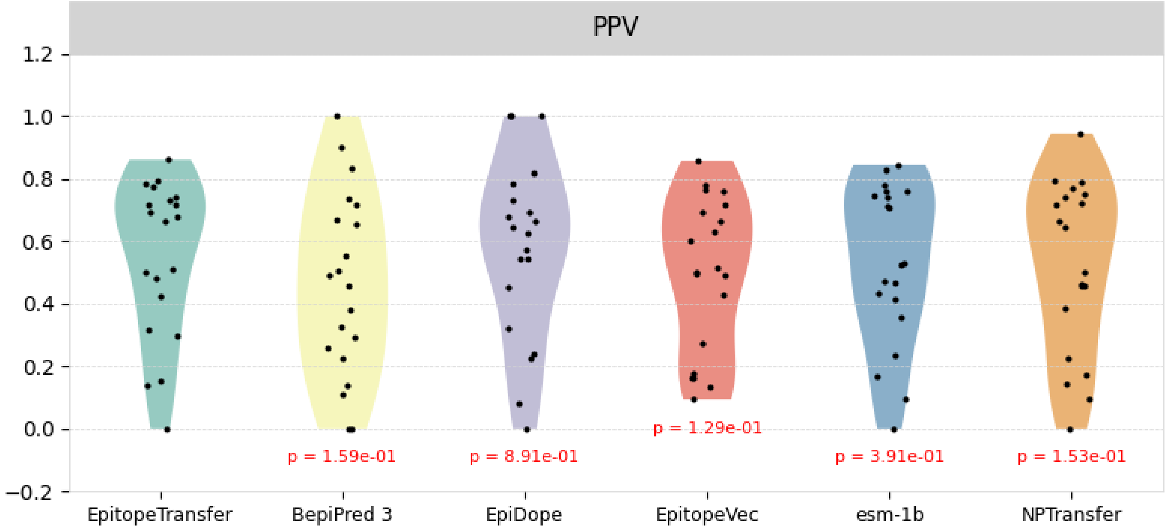


Figure 7: Performance plot for the PPV metric

Comparison Results for Sensitivity

Metric	Pair	Medians of diff.	p-value	FDR	Signific
Sensit.	EpitopeTransfer vs BepiPred 3	0.341 (0.167, 0.529)	0.00169	0.00423	Yes
Sensit.	EpitopeTransfer vs EpiDope	0.554 (0.370, 0.686)	0.00004	0.00019	Yes
Sensit.	EpitopeTransfer vs EpitopeVec	0.155 (0.006, 0.272)	0.04937	0.08228	No
Sensit.	EpitopeTransfer vs esm-1b	0.040 (-0.039, 0.155)	0.31651	0.39564	No
Sensit.	EpitopeTransfer vs NPTransfer	0.012 (-0.090, 0.171)	0.51359	0.51359	No

Table 16: Comparison Results for Sensitivity

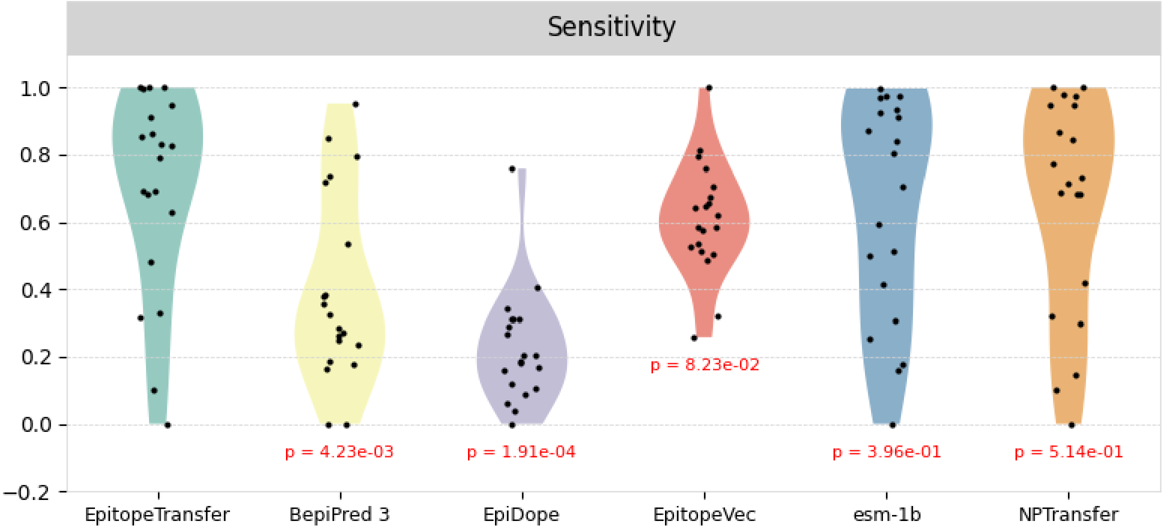


Figure 8: Performance plot for the Sensitivity metric

Comparison Results for Specificity

Metric	Pair	Medians of diff.	p-value	FDR	Signific
Specif.	EpitopeTransfer vs BepiPred 3	-0.152 (-0.377, 0.041)	0.11338	0.28344	No
Specif.	EpitopeTransfer vs EpiDope	-0.366 (-0.591, -0.144)	0.00306	0.01531	Yes
Specif.	EpitopeTransfer vs EpitopeVec	0.007 (-0.167, 0.176)	0.95298	0.95298	No
Specif.	EpitopeTransfer vs esm-1b	0.047 (-0.039, 0.145)	0.17700	0.29499	No
Specif.	EpitopeTransfer vs NPTransfer	0.052 (-0.128, 0.246)	0.50750	0.63438	No

Table 17: Comparison Results for Specificity

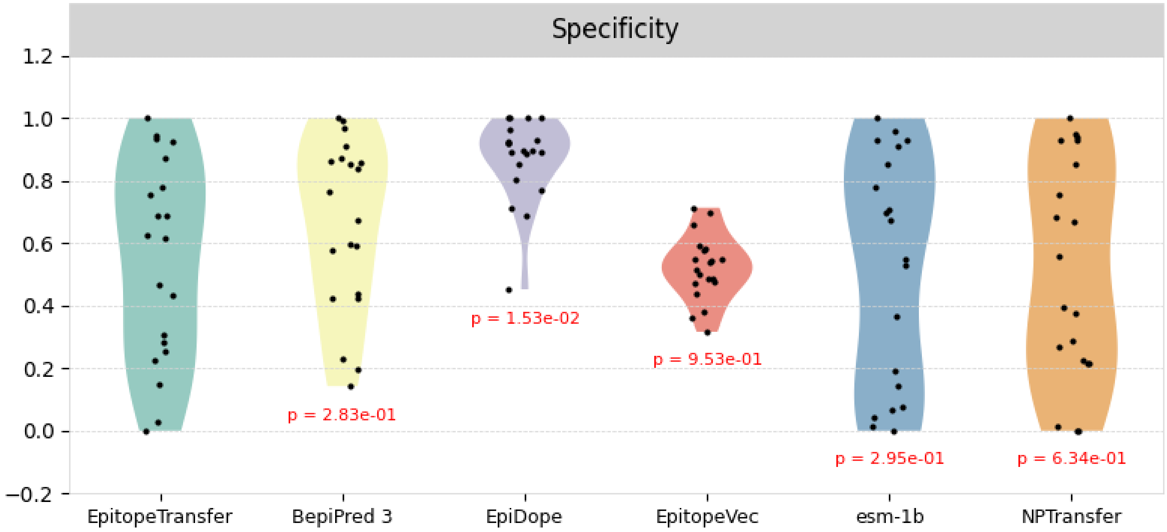


Figure 9: Performance plot for the Specificity metric

Summary of Comparison Results

Metric	Pair	Medians of diff.	p-value	FDR	Significant
AUC	EpitopeTransfer vs BepiPred 3	0.193 (0.088, 0.276)	0.00097	0.00483	Yes
AUC	EpitopeTransfer vs EpiDope	0.054 (-0.018, 0.120)	0.12319	0.12319	No
AUC	EpitopeTransfer vs EpitopeVec	0.091 (0.014, 0.178)	0.03234	0.04043	Yes
AUC	EpitopeTransfer vs esm-1b	0.028 (0.004, 0.060)	0.02299	0.03831	Yes
AUC	EpitopeTransfer vs NPTransfer	0.061 (0.015, 0.083)	0.00392	0.00979	Yes
BACC	EpitopeTransfer vs BepiPred 3	0.088 (0.035, 0.165)	0.01407	0.03402	Yes
BACC	EpitopeTransfer vs EpiDope	0.073 (0.011, 0.140)	0.02041	0.03402	Yes
BACC	EpitopeTransfer vs EpitopeVec	0.057 (-0.013, 0.140)	0.12319	0.12319	No
BACC	EpitopeTransfer vs esm-1b	0.041 (0.011, 0.074)	0.01597	0.03402	Yes
BACC	EpitopeTransfer vs NPTransfer	0.030 (0.002, 0.087)	0.03285	0.04106	Yes
F1	EpitopeTransfer vs BepiPred 3	0.247 (0.139, 0.364)	0.00013	0.00031	Yes
F1	EpitopeTransfer vs EpiDope	0.369 (0.236, 0.477)	0.00003	0.00013	Yes
F1	EpitopeTransfer vs EpitopeVec	0.110 (0.041, 0.187)	0.00284	0.00334	Yes
F1	EpitopeTransfer vs esm-1b	0.055 (0.020, 0.087)	0.00334	0.00334	Yes
F1	EpitopeTransfer vs NPTransfer	0.058 (0.021, 0.114)	0.00030	0.00050	Yes
MCC	EpitopeTransfer vs BepiPred 3	0.204 (0.105, 0.390)	0.00823	0.01698	Yes
MCC	EpitopeTransfer vs EpiDope	0.134 (0.023, 0.254)	0.02041	0.02041	Yes
MCC	EpitopeTransfer vs EpitopeVec	0.150 (0.027, 0.296)	0.02041	0.02041	Yes

MCC	EpitopeTransfer vs esm-1b	0.082 (0.025, 0.143)	0.00618	0.01698	Yes
MCC	EpitopeTransfer vs NPTransfer	0.065 (0.019, 0.154)	0.01019	0.01698	Yes
NPV	EpitopeTransfer vs BepiPred 3	0.140 (0.027, 0.307)	0.01236	0.01545	Yes
NPV	EpitopeTransfer vs EpiDope	0.116 (0.024, 0.315)	0.00533	0.01316	Yes
NPV	EpitopeTransfer vs EpitopeVec	0.087 (0.010, 0.301)	0.02299	0.02299	Yes
NPV	EpitopeTransfer vs esm-1b	0.050 (0.012, 0.158)	0.00789	0.01316	Yes
NPV	EpitopeTransfer vs NPTransfer	0.070 (0.020, 0.304)	0.00405	0.01316	Yes
PPV	EpitopeTransfer vs BepiPred 3	0.089 (-0.012, 0.206)	0.09551	0.15919	No
PPV	EpitopeTransfer vs EpiDope	0.007 (-0.097, 0.097)	0.89057	0.89057	No
PPV	EpitopeTransfer vs EpitopeVec	0.075 (0.009, 0.133)	0.02582	0.12911	No
PPV	EpitopeTransfer vs esm-1b	0.014 (-0.014, 0.050)	0.31241	0.39051	No
PPV	EpitopeTransfer vs NPTransfer	0.025 (-0.002, 0.090)	0.06111	0.15279	No
Sensit.	EpitopeTransfer vs BepiPred 3	0.341 (0.167, 0.529)	0.00169	0.00423	Yes
Sensit.	EpitopeTransfer vs EpiDope	0.554 (0.370, 0.686)	0.00004	0.00019	Yes
Sensit.	EpitopeTransfer vs EpitopeVec	0.155 (0.006, 0.272)	0.04937	0.08228	No
Sensit.	EpitopeTransfer vs esm-1b	0.040 (-0.039, 0.155)	0.31651	0.39564	No
Sensit.	EpitopeTransfer vs NPTransfer	0.012 (-0.090, 0.171)	0.51359	0.51359	No
Specif.	EpitopeTransfer vs BepiPred 3	-0.152 (-0.377, 0.041)	0.11338	0.28344	No
Specif.	EpitopeTransfer vs EpiDope	-0.366 (-0.591, -0.144)	0.00306	0.01531	Yes
Specif.	EpitopeTransfer vs EpitopeVec	0.007 (-0.167, 0.176)	0.95298	0.95298	No
Specif.	EpitopeTransfer vs esm-1b	0.047 (-0.039, 0.145)	0.17700	0.29499	No
Specif.	EpitopeTransfer vs NPTransfer	0.052 (-0.128, 0.246)	0.50750	0.63438	No

Table 19: Summary of Comparison Results across All Metrics

Appendix E

The estimated performance for each method on each dataset is presented below, following the same evaluation framework described in Appendix E. In this appendix, however, all experiments were conducted using ESM-2 (650M) as the base model for every method. *Method* refers to the employed approach, including the primary method, **EpitopeTransfer**, which leverages phylogenetic information, and internal and external baselines. The internal baselines are **ESM-2**, a pretrained protein language model fine-tuned for epitope prediction, and **Non-phylogenetic transfer (NPTransfer)**, a transfer learning method that does not utilize phylogenetic relationships. The external baselines include **BepiPred 3.0**, **EpiDope**, and **EpitopeVec**, which are methods developed outside this study and are included for comparative evaluation. *Dataset* corresponds to the data from 20 specific taxa, and *Value* represents the value of each presented metric. The evaluated metrics include **AUC** (Area Under the Curve), **F1** score, **MCC** (Matthews Correlation Coefficient), **Accuracy**, **PPV** (Positive Predictive Value), **NPV** (Negative Predictive Value), **Sensitivity**, and **Specificity**.

Table 20: Comparison of methods for AUC

Method	Dataset	Value
EpitopeTransfer	B. pertussis	0.533
BepiPred 3.0	B. pertussis	0.365
EpiDope	B. pertussis	0.359
EpitopeVec	B. pertussis	0.750
ESM-2	B. pertussis	0.520
NPTransfer	B. pertussis	0.371
EpitopeTransfer	C. difficile	0.656
BepiPred 3.0	C. difficile	0.425
EpiDope	C. difficile	0.744
EpitopeVec	C. difficile	0.851
ESM-2	C. difficile	0.673
NPTransfer	C. difficile	0.511
EpitopeTransfer	C. trachomatis	0.834
BepiPred 3.0	C. trachomatis	0.559
EpiDope	C. trachomatis	0.665
EpitopeVec	C. trachomatis	0.717
ESM-2	C. trachomatis	0.834

Method	Dataset	Value
NPTransfer	C. trachomatis	0.768
EpitopeTransfer	Corynebacterium	0.632
BepiPred 3.0	Corynebacterium	0.648
EpiDope	Corynebacterium	0.733
EpitopeVec	Corynebacterium	0.728
ESM-2	Corynebacterium	0.605
NPTransfer	Corynebacterium	0.596
EpitopeTransfer	E. coli	0.909
BepiPred 3.0	E. coli	0.400
EpiDope	E. coli	0.804
EpitopeVec	E. coli	0.533
ESM-2	E. coli	0.855
NPTransfer	E. coli	0.810
EpitopeTransfer	Enterobacteriaceae	0.821
BepiPred 3.0	Enterobacteriaceae	0.554
EpiDope	Enterobacteriaceae	0.613
EpitopeVec	Enterobacteriaceae	0.549
ESM-2	Enterobacteriaceae	0.701
NPTransfer	Enterobacteriaceae	0.675
EpitopeTransfer	Filoviridae	0.959
BepiPred 3.0	Filoviridae	0.538
EpiDope	Filoviridae	0.877
EpitopeVec	Filoviridae	0.752
ESM-2	Filoviridae	0.936
NPTransfer	Filoviridae	0.906
EpitopeTransfer	Human gammaherpesvirus 4	0.612
BepiPred 3.0	Human gammaherpesvirus 4	0.398
EpiDope	Human gammaherpesvirus 4	0.617
EpitopeVec	Human gammaherpesvirus 4	0.560
ESM-2	Human gammaherpesvirus 4	0.616
NPTransfer	Human gammaherpesvirus 4	0.612
EpitopeTransfer	Influenza A	0.654
BepiPred 3.0	Influenza A	0.570

Method	**Dataset**	**Value**
EpiDope	Influenza A	0.523
EpitopeVec	Influenza A	0.630
ESM-2	Influenza A	0.654
NPTransfer	Influenza A	0.653
EpitopeTransfer	Lentivirus	0.666
BepiPred 3.0	Lentivirus	0.581
EpiDope	Lentivirus	0.552
EpitopeVec	Lentivirus	0.596
ESM-2	Lentivirus	0.640
NPTransfer	Lentivirus	0.609
EpitopeTransfer	M. tuberculosis	0.479
BepiPred 3.0	M. tuberculosis	0.444
EpiDope	M. tuberculosis	0.481
EpitopeVec	M. tuberculosis	0.481
ESM-2	M. tuberculosis	0.489
NPTransfer	M. tuberculosis	0.456
EpitopeTransfer	Measles morbilivirus	0.595
BepiPred 3.0	Measles morbilivirus	0.381
EpiDope	Measles morbilivirus	0.501
EpitopeVec	Measles morbilivirus	0.538
ESM-2	Measles morbilivirus	0.479
NPTransfer	Measles morbilivirus	0.372
EpitopeTransfer	Mononegavirales	0.787
BepiPred 3.0	Mononegavirales	0.446
EpiDope	Mononegavirales	0.817
EpitopeVec	Mononegavirales	0.671
ESM-2	Mononegavirales	0.731
NPTransfer	Mononegavirales	0.793
EpitopeTransfer	Orthopox	0.649
BepiPred 3.0	Orthopox	0.728
EpiDope	Orthopox	0.688
EpitopeVec	Orthopox	0.322
ESM-2	Orthopox	0.613

Method	**Dataset**	**Value**
NPTransfer	Orthopox	0.634
EpitopeTransfer	Ovolvulus	0.606
BepiPred 3.0	Ovolvulus	0.721
EpiDope	Ovolvulus	0.495
EpitopeVec	Ovolvulus	0.585
ESM-2	Ovolvulus	0.569
NPTransfer	Ovolvulus	0.567
EpitopeTransfer	P. aeruginosa	0.720
BepiPred 3.0	P. aeruginosa	0.040
EpiDope	P. aeruginosa	0.874
EpitopeVec	P. aeruginosa	0.565
ESM-2	P. aeruginosa	0.790
NPTransfer	P. aeruginosa	0.712
EpitopeTransfer	P. falciparum	0.794
BepiPred 3.0	P. falciparum	0.675
EpiDope	P. falciparum	0.603
EpitopeVec	P. falciparum	0.512
ESM-2	P. falciparum	0.685
NPTransfer	P. falciparum	0.796
EpitopeTransfer	S. mansoni	0.539
BepiPred 3.0	S. mansoni	0.560
EpiDope	S. mansoni	0.672
EpitopeVec	S. mansoni	0.447
ESM-2	S. mansoni	0.534
NPTransfer	S. mansoni	0.565
EpitopeTransfer	Sars-cov-2	0.625
BepiPred 3.0	Sars-cov-2	0.569
EpiDope	Sars-cov-2	0.597
EpitopeVec	Sars-cov-2	0.630
ESM-2	Sars-cov-2	0.605
NPTransfer	Sars-cov-2	0.605
EpitopeTransfer	T. gondii	0.651
BepiPred 3.0	T. gondii	0.454

Method	**Dataset**	**Value**
EpiDope	T. gondii	0.466
EpitopeVec	T. gondii	0.620
ESM-2	T. gondii	0.601
NPTransfer	T. gondii	0.582

Table 21: Comparison of methods for F1

Method	**Dataset**	**Value**
EpitopeTransfer	B. pertussis	0.619
BepiPred 3.0	B. pertussis	0.778
EpiDope	B. pertussis	0.288
EpitopeVec	B. pertussis	0.754
ESM-2	B. pertussis	0.639
NPTransfer	B. pertussis	0.688
EpitopeTransfer	C. difficile	0.236
BepiPred 3.0	C. difficile	0.000
EpiDope	C. difficile	0.000
EpitopeVec	C. difficile	0.282
ESM-2	C. difficile	0.087
NPTransfer	C. difficile	0.088
EpitopeTransfer	C. trachomatis	0.774
BepiPred 3.0	C. trachomatis	0.583
EpiDope	C. trachomatis	0.311
EpitopeVec	C. trachomatis	0.624
ESM-2	C. trachomatis	0.662
NPTransfer	C. trachomatis	0.696
EpitopeTransfer	Corynebacterium	0.672
BepiPred 3.0	Corynebacterium	0.303
EpiDope	Corynebacterium	0.286
EpitopeVec	Corynebacterium	0.672
ESM-2	Corynebacterium	0.639
NPTransfer	Corynebacterium	0.623
EpitopeTransfer	E. coli	0.886

Method	Dataset	Value
BepiPred 3.0	E. coli	0.375
EpiDope	E. coli	0.340
EpitopeVec	E. coli	0.628
ESM-2	E. coli	0.855
NPTransfer	E. coli	0.855
EpitopeTransfer	Enterobacteriaceae	0.709
BepiPred 3.0	Enterobacteriaceae	0.434
EpiDope	Enterobacteriaceae	0.188
EpitopeVec	Enterobacteriaceae	0.534
ESM-2	Enterobacteriaceae	0.642
NPTransfer	Enterobacteriaceae	0.651
EpitopeTransfer	Filoviridae	0.651
BepiPred 3.0	Filoviridae	0.235
EpiDope	Filoviridae	0.316
EpitopeVec	Filoviridae	0.271
ESM-2	Filoviridae	0.444
NPTransfer	Filoviridae	0.280
EpitopeTransfer	Human gammaherpesvirus 4	0.341
BepiPred 3.0	Human gammaherpesvirus 4	0.273
EpiDope	Human gammaherpesvirus 4	0.244
EpitopeVec	Human gammaherpesvirus 4	0.508
ESM-2	Human gammaherpesvirus 4	0.421
NPTransfer	Human gammaherpesvirus 4	0.305
EpitopeTransfer	Influenza A	0.782
BepiPred 3.0	Influenza A	0.539
EpiDope	Influenza A	0.206
EpitopeVec	Influenza A	0.798
ESM-2	Influenza A	0.770
NPTransfer	Influenza A	0.783
EpitopeTransfer	Lentivirus	0.821
BepiPred 3.0	Lentivirus	0.490
EpiDope	Lentivirus	0.457
EpitopeVec	Lentivirus	0.571

Method	**Dataset**	**Value**
ESM-2	Lentivirus	0.796
NPTransfer	Lentivirus	0.817
EpitopeTransfer	M. tuberculosis	0.257
BepiPred 3.0	M. tuberculosis	0.254
EpiDope	M. tuberculosis	0.155
EpitopeVec	M. tuberculosis	0.510
ESM-2	M. tuberculosis	0.653
NPTransfer	M. tuberculosis	0.280
EpitopeTransfer	Measles morbilivirus	0.638
BepiPred 3.0	Measles morbilivirus	0.281
EpiDope	Measles morbilivirus	0.356
EpitopeVec	Measles morbilivirus	0.587
ESM-2	Measles morbilivirus	0.553
NPTransfer	Measles morbilivirus	0.521
EpitopeTransfer	Mononegavirales	0.638
BepiPred 3.0	Mononegavirales	0.271
EpiDope	Mononegavirales	0.499
EpitopeVec	Mononegavirales	0.495
ESM-2	Mononegavirales	0.446
NPTransfer	Mononegavirales	0.591
EpitopeTransfer	Orthopox	0.352
BepiPred 3.0	Orthopox	0.492
EpiDope	Orthopox	0.351
EpitopeVec	Orthopox	0.138
ESM-2	Orthopox	0.295
NPTransfer	Orthopox	0.320
EpitopeTransfer	Ovolvulus	0.142
BepiPred 3.0	Ovolvulus	0.364
EpiDope	Ovolvulus	0.053
EpitopeVec	Ovolvulus	0.262
ESM-2	Ovolvulus	0.227
NPTransfer	Ovolvulus	0.215
EpitopeTransfer	P. aeruginosa	0.742

Method	**Dataset**	**Value**
BepiPred 3.0	P. aeruginosa	0.000
EpiDope	P. aeruginosa	0.116
EpitopeVec	P. aeruginosa	0.698
ESM-2	P. aeruginosa	0.650
NPTransfer	P. aeruginosa	0.667
EpitopeTransfer	P. falciparum	0.805
BepiPred 3.0	P. falciparum	0.372
EpiDope	P. falciparum	0.431
EpitopeVec	P. falciparum	0.642
ESM-2	P. falciparum	0.736
NPTransfer	P. falciparum	0.804
EpitopeTransfer	S. mansoni	0.368
BepiPred 3.0	S. mansoni	0.367
EpiDope	S. mansoni	0.370
EpitopeVec	S. mansoni	0.296
ESM-2	S. mansoni	0.445
NPTransfer	S. mansoni	0.153
EpitopeTransfer	Sars-cov-2	0.141
BepiPred 3.0	Sars-cov-2	0.136
EpiDope	Sars-cov-2	0.262
EpitopeVec	Sars-cov-2	0.222
ESM-2	Sars-cov-2	0.198
NPTransfer	Sars-cov-2	0.148
EpitopeTransfer	T. gondii	0.832
BepiPred 3.0	T. gondii	0.703
EpiDope	T. gondii	0.297
EpitopeVec	T. gondii	0.682
ESM-2	T. gondii	0.697
NPTransfer	T. gondii	0.814

Table 22: Comparison of methods for MCC

Method	**Dataset**	**Value**
EpitopeTransfer	B. pertussis	0.085
BepiPred 3.0	B. pertussis	-0.008
EpiDope	B. pertussis	-0.108
EpitopeVec	B. pertussis	0.351
ESM-2	B. pertussis	0.020
NPTransfer	B. pertussis	-0.255
EpitopeTransfer	C. difficile	0.137
BepiPred 3.0	C. difficile	-0.027
EpiDope	C. difficile	0.000
EpitopeVec	C. difficile	0.279
ESM-2	C. difficile	0.113
NPTransfer	C. difficile	-0.017
EpitopeTransfer	C. trachomatis	0.568
BepiPred 3.0	C. trachomatis	-0.055
EpiDope	C. trachomatis	0.137
EpitopeVec	C. trachomatis	0.286
ESM-2	C. trachomatis	0.493
NPTransfer	C. trachomatis	0.370
EpitopeTransfer	Corynebacterium	0.282
BepiPred 3.0	Corynebacterium	0.247
EpiDope	Corynebacterium	0.309
EpitopeVec	Corynebacterium	0.315
ESM-2	Corynebacterium	0.145
NPTransfer	Corynebacterium	0.128
EpitopeTransfer	E. coli	0.442
BepiPred 3.0	E. coli	-0.136
EpiDope	E. coli	0.248
EpitopeVec	E. coli	0.031
ESM-2	E. coli	0.000
NPTransfer	E. coli	0.000
EpitopeTransfer	Enterobacteriaceae	0.427
BepiPred 3.0	Enterobacteriaceae	0.054

Method	**Dataset**	**Value**
EpiDope	Enterobacteriaceae	0.144
EpitopeVec	Enterobacteriaceae	0.061
ESM-2	Enterobacteriaceae	0.310
NPTransfer	Enterobacteriaceae	0.200
EpitopeTransfer	Filoviridae	0.610
BepiPred 3.0	Filoviridae	0.143
EpiDope	Filoviridae	0.240
EpitopeVec	Filoviridae	0.202
ESM-2	Filoviridae	0.388
NPTransfer	Filoviridae	0.228
EpitopeTransfer	Human gammaherpesvirus 4	0.275
BepiPred 3.0	Human gammaherpesvirus 4	-0.180
EpiDope	Human gammaherpesvirus 4	0.068
EpitopeVec	Human gammaherpesvirus 4	0.070
ESM-2	Human gammaherpesvirus 4	0.148
NPTransfer	Human gammaherpesvirus 4	0.210
EpitopeTransfer	Influenza A	0.218
BepiPred 3.0	Influenza A	0.239
EpiDope	Influenza A	0.054
EpitopeVec	Influenza A	0.139
ESM-2	Influenza A	0.171
NPTransfer	Influenza A	0.212
EpitopeTransfer	Lentivirus	0.350
BepiPred 3.0	Lentivirus	0.376
EpiDope	Lentivirus	0.033
EpitopeVec	Lentivirus	0.067
ESM-2	Lentivirus	0.245
NPTransfer	Lentivirus	0.303
EpitopeTransfer	M. tuberculosis	-0.016
BepiPred 3.0	M. tuberculosis	0.029
EpiDope	M. tuberculosis	0.033
EpitopeVec	M. tuberculosis	-0.008
ESM-2	M. tuberculosis	0.035

Method	**Dataset**	**Value**
NPTransfer	M. tuberculosis	-0.071
EpitopeTransfer	Measles morbilivirus	0.068
BepiPred 3.0	Measles morbilivirus	-0.310
EpiDope	Measles morbilivirus	0.079
EpitopeVec	Measles morbilivirus	0.091
ESM-2	Measles morbilivirus	-0.160
NPTransfer	Measles morbilivirus	-0.325
EpitopeTransfer	Mononegavirales	0.428
BepiPred 3.0	Mononegavirales	-0.136
EpiDope	Mononegavirales	0.336
EpitopeVec	Mononegavirales	0.170
ESM-2	Mononegavirales	0.299
NPTransfer	Mononegavirales	0.376
EpitopeTransfer	Orthopox	0.168
BepiPred 3.0	Orthopox	0.375
EpiDope	Orthopox	0.163
EpitopeVec	Orthopox	-0.206
ESM-2	Orthopox	0.078
NPTransfer	Orthopox	0.124
EpitopeTransfer	Ovolvulus	0.095
BepiPred 3.0	Ovolvulus	0.277
EpiDope	Ovolvulus	-0.055
EpitopeVec	Ovolvulus	0.064
ESM-2	Ovolvulus	0.083
NPTransfer	Ovolvulus	0.085
EpitopeTransfer	P. aeruginosa	0.249
BepiPred 3.0	P. aeruginosa	-0.258
EpiDope	P. aeruginosa	0.137
EpitopeVec	P. aeruginosa	0.145
ESM-2	P. aeruginosa	0.407
NPTransfer	P. aeruginosa	0.397
EpitopeTransfer	P. falciparum	0.410
BepiPred 3.0	P. falciparum	0.119

Method	Dataset	Value
EpiDope	P. falciparum	0.088
EpitopeVec	P. falciparum	0.018
ESM-2	P. falciparum	0.270
NPTransfer	P. falciparum	0.425
EpitopeTransfer	S. mansoni	0.069
BepiPred 3.0	S. mansoni	0.126
EpiDope	S. mansoni	0.185
EpitopeVec	S. mansoni	-0.016
ESM-2	S. mansoni	0.042
NPTransfer	S. mansoni	0.036
EpitopeTransfer	Sars-cov-2	0.018
BepiPred 3.0	Sars-cov-2	0.011
EpiDope	Sars-cov-2	0.169
EpitopeVec	Sars-cov-2	0.101
ESM-2	Sars-cov-2	0.095
NPTransfer	Sars-cov-2	0.047
EpitopeTransfer	T. gondii	0.312
BepiPred 3.0	T. gondii	-0.070
EpiDope	T. gondii	0.092
EpitopeVec	T. gondii	0.084
ESM-2	T. gondii	0.166
NPTransfer	T. gondii	-0.039

Table 23: Comparison of methods for Balanced Accuracy

Method	Dataset	Value
EpitopeTransfer	B. pertussis	0.547
BepiPred 3.0	B. pertussis	0.497
EpiDope	B. pertussis	0.451
EpitopeVec	B. pertussis	0.694
ESM-2	B. pertussis	0.511
NPTransfer	B. pertussis	0.384
EpitopeTransfer	C. difficile	0.605

Method	**Dataset**	**Value**
BepiPred 3.0	C. difficile	0.496
EpiDope	C. difficile	0.500
EpitopeVec	C. difficile	0.737
ESM-2	C. difficile	0.521
NPTransfer	C. difficile	0.491
EpitopeTransfer	C. trachomatis	0.783
BepiPred 3.0	C. trachomatis	0.476
EpiDope	C. trachomatis	0.549
EpitopeVec	C. trachomatis	0.642
ESM-2	C. trachomatis	0.727
NPTransfer	C. trachomatis	0.685
EpitopeTransfer	Corynebacterium	0.632
BepiPred 3.0	Corynebacterium	0.576
EpiDope	Corynebacterium	0.583
EpitopeVec	Corynebacterium	0.655
ESM-2	Corynebacterium	0.559
NPTransfer	Corynebacterium	0.556
EpitopeTransfer	E. coli	0.623
BepiPred 3.0	E. coli	0.429
EpiDope	E. coli	0.603
EpitopeVec	E. coli	0.518
ESM-2	E. coli	0.500
NPTransfer	E. coli	0.500
EpitopeTransfer	Enterobacteriaceae	0.713
BepiPred 3.0	Enterobacteriaceae	0.526
EpiDope	Enterobacteriaceae	0.536
EpitopeVec	Enterobacteriaceae	0.530
ESM-2	Enterobacteriaceae	0.655
NPTransfer	Enterobacteriaceae	0.584
EpitopeTransfer	Filoviridae	0.827
BepiPred 3.0	Filoviridae	0.616
EpiDope	Filoviridae	0.618
EpitopeVec	Filoviridae	0.667

Method	**Dataset**	**Value**
ESM-2	Filoviridae	0.682
NPTransfer	Filoviridae	0.591
EpitopeTransfer	Human gammaherpesvirus 4	0.589
BepiPred 3.0	Human gammaherpesvirus 4	0.415
EpiDope	Human gammaherpesvirus 4	0.523
EpitopeVec	Human gammaherpesvirus 4	0.535
ESM-2	Human gammaherpesvirus 4	0.565
NPTransfer	Human gammaherpesvirus 4	0.568
EpitopeTransfer	Influenza A	0.616
BepiPred 3.0	Influenza A	0.629
EpiDope	Influenza A	0.519
EpitopeVec	Influenza A	0.566
ESM-2	Influenza A	0.591
NPTransfer	Influenza A	0.612
EpitopeTransfer	Lentivirus	0.629
BepiPred 3.0	Lentivirus	0.662
EpiDope	Lentivirus	0.516
EpitopeVec	Lentivirus	0.535
ESM-2	Lentivirus	0.592
NPTransfer	Lentivirus	0.575
EpitopeTransfer	M. tuberculosis	0.494
BepiPred 3.0	M. tuberculosis	0.511
EpiDope	M. tuberculosis	0.509
EpitopeVec	M. tuberculosis	0.496
ESM-2	M. tuberculosis	0.513
NPTransfer	M. tuberculosis	0.470
EpitopeTransfer	Measles morbilivirus	0.516
BepiPred 3.0	Measles morbilivirus	0.346
EpiDope	Measles morbilivirus	0.533
EpitopeVec	Measles morbilivirus	0.543
ESM-2	Measles morbilivirus	0.440
NPTransfer	Measles morbilivirus	0.386
EpitopeTransfer	Mononegavirales	0.721

Method	Dataset	Value
BepiPred 3.0	Mononegavirales	0.430
EpiDope	Mononegavirales	0.646
EpitopeVec	Mononegavirales	0.589
ESM-2	Mononegavirales	0.621
NPTransfer	Mononegavirales	0.689
EpitopeTransfer	Orthopox	0.607
BepiPred 3.0	Orthopox	0.699
EpiDope	Orthopox	0.605
EpitopeVec	Orthopox	0.365
ESM-2	Orthopox	0.550
NPTransfer	Orthopox	0.578
EpitopeTransfer	Ovolvulus	0.528
BepiPred 3.0	Ovolvulus	0.687
EpiDope	Ovolvulus	0.480
EpitopeVec	Ovolvulus	0.545
ESM-2	Ovolvulus	0.543
NPTransfer	Ovolvulus	0.541
EpitopeTransfer	P. aeruginosa	0.634
BepiPred 3.0	P. aeruginosa	0.455
EpiDope	P. aeruginosa	0.531
EpitopeVec	P. aeruginosa	0.579
ESM-2	P. aeruginosa	0.717
NPTransfer	P. aeruginosa	0.714
EpitopeTransfer	P. falciparum	0.686
BepiPred 3.0	P. falciparum	0.550
EpiDope	P. falciparum	0.541
EpitopeVec	P. falciparum	0.509
ESM-2	P. falciparum	0.632
NPTransfer	P. falciparum	0.700
EpitopeTransfer	S. mansoni	0.537
BepiPred 3.0	S. mansoni	0.562
EpiDope	S. mansoni	0.581
EpitopeVec	S. mansoni	0.491

Method	Dataset	Value
ESM-2	S. mansoni	0.507
NPTransfer	S. mansoni	0.511
EpitopeTransfer	Sars-cov-2	0.511
BepiPred 3.0	Sars-cov-2	0.507
EpiDope	Sars-cov-2	0.592
EpitopeVec	Sars-cov-2	0.584
ESM-2	Sars-cov-2	0.553
NPTransfer	Sars-cov-2	0.524
EpitopeTransfer	T. gondii	0.613
BepiPred 3.0	T. gondii	0.468
EpiDope	T. gondii	0.537
EpitopeVec	T. gondii	0.544
ESM-2	T. gondii	0.588
NPTransfer	T. gondii	0.498

Table 24: Comparison of methods for PPV

Method	Dataset	Value
EpitopeTransfer	B. pertussis	0.757
BepiPred 3.0	B. pertussis	0.717
EpiDope	B. pertussis	0.625
EpitopeVec	B. pertussis	0.857
ESM-2	B. pertussis	0.726
NPTransfer	B. pertussis	0.658
EpitopeTransfer	C. difficile	0.158
BepiPred 3.0	C. difficile	0.000
EpiDope	C. difficile	0.000
EpitopeVec	C. difficile	0.164
ESM-2	C. difficile	0.400
NPTransfer	C. difficile	0.080
EpitopeTransfer	C. trachomatis	0.817
BepiPred 3.0	C. trachomatis	0.491
EpiDope	C. trachomatis	0.667

Method	**Dataset**	**Value**
EpitopeVec	C. trachomatis	0.667
ESM-2	C. trachomatis	0.877
NPTransfer	C. trachomatis	0.682
EpitopeTransfer	Corynebacterium	0.571
BepiPred 3.0	Corynebacterium	0.833
EpiDope	Corynebacterium	1.000
EpitopeVec	Corynebacterium	0.603
ESM-2	Corynebacterium	0.511
NPTransfer	Corynebacterium	0.512
EpitopeTransfer	E. coli	0.796
BepiPred 3.0	E. coli	0.656
EpiDope	E. coli	1.000
EpitopeVec	E. coli	0.759
ESM-2	E. coli	0.746
NPTransfer	E. coli	0.746
EpitopeTransfer	Enterobacteriaceae	0.674
BepiPred 3.0	Enterobacteriaceae	0.507
EpiDope	Enterobacteriaceae	0.733
EpitopeVec	Enterobacteriaceae	0.497
ESM-2	Enterobacteriaceae	0.627
NPTransfer	Enterobacteriaceae	0.524
EpitopeTransfer	Filoviridae	0.603
BepiPred 3.0	Filoviridae	0.138
EpiDope	Filoviridae	0.321
EpitopeVec	Filoviridae	0.164
ESM-2	Filoviridae	0.480
NPTransfer	Filoviridae	0.371
EpitopeTransfer	Human gammaherpesvirus 4	0.829
BepiPred 3.0	Human gammaherpesvirus 4	0.326
EpiDope	Human gammaherpesvirus 4	0.542
EpitopeVec	Human gammaherpesvirus 4	0.491
ESM-2	Human gammaherpesvirus 4	0.579
NPTransfer	Human gammaherpesvirus 4	0.740

Method	**Dataset**	**Value**
EpitopeTransfer	Influenza A	0.812
BepiPred 3.0	Influenza A	0.901
EpiDope	Influenza A	0.817
EpitopeVec	Influenza A	0.782
ESM-2	Influenza A	0.799
NPTransfer	Influenza A	0.810
EpitopeTransfer	Lentivirus	0.725
BepiPred 3.0	Lentivirus	1.000
EpiDope	Lentivirus	0.680
EpitopeVec	Lentivirus	0.692
ESM-2	Lentivirus	0.707
NPTransfer	Lentivirus	0.693
EpitopeTransfer	M. tuberculosis	0.502
BepiPred 3.0	M. tuberculosis	0.553
EpiDope	M. tuberculosis	0.574
EpitopeVec	M. tuberculosis	0.515
ESM-2	M. tuberculosis	0.526
NPTransfer	M. tuberculosis	0.454
EpitopeTransfer	Measles morbilivirus	0.478
BepiPred 3.0	Measles morbilivirus	0.293
EpiDope	Measles morbilivirus	0.542
EpitopeVec	Measles morbilivirus	0.502
ESM-2	Measles morbilivirus	0.433
NPTransfer	Measles morbilivirus	0.403
EpitopeTransfer	Mononegavirales	0.586
BepiPred 3.0	Mononegavirales	0.259
EpiDope	Mononegavirales	0.646
EpitopeVec	Mononegavirales	0.427
ESM-2	Mononegavirales	0.643
NPTransfer	Mononegavirales	0.587
EpitopeTransfer	Orthopox	0.258
BepiPred 3.0	Orthopox	0.456
EpiDope	Orthopox	0.228

Method	**Dataset**	**Value**
EpitopeVec	Orthopox	0.094
ESM-2	Orthopox	0.212
NPTransfer	Orthopox	0.239
EpitopeTransfer	Ovolvulus	0.306
BepiPred 3.0	Ovolvulus	0.225
EpiDope	Ovolvulus	0.080
EpitopeVec	Ovolvulus	0.176
ESM-2	Ovolvulus	0.216
NPTransfer	Ovolvulus	0.227
EpitopeTransfer	P. aeruginosa	0.800
BepiPred 3.0	P. aeruginosa	0.000
EpiDope	P. aeruginosa	1.000
EpitopeVec	P. aeruginosa	0.765
ESM-2	P. aeruginosa	0.952
NPTransfer	P. aeruginosa	0.933
EpitopeTransfer	P. falciparum	0.743
BepiPred 3.0	P. falciparum	0.736
EpiDope	P. falciparum	0.693
EpitopeVec	P. falciparum	0.631
ESM-2	P. falciparum	0.720
NPTransfer	P. falciparum	0.757
EpitopeTransfer	S. mansoni	0.324
BepiPred 3.0	S. mansoni	0.379
EpiDope	S. mansoni	0.454
EpitopeVec	S. mansoni	0.274
ESM-2	S. mansoni	0.287
NPTransfer	S. mansoni	0.338
EpitopeTransfer	Sars-cov-2	0.115
BepiPred 3.0	Sars-cov-2	0.110
EpiDope	Sars-cov-2	0.238
EpitopeVec	Sars-cov-2	0.135
ESM-2	Sars-cov-2	0.176
NPTransfer	Sars-cov-2	0.142

Method	Dataset	Value
EpitopeTransfer	T. gondii	0.745
BepiPred 3.0	T. gondii	0.671
EpiDope	T. gondii	0.787
EpitopeVec	T. gondii	0.720
ESM-2	T. gondii	0.753
NPTransfer	T. gondii	0.688

Table 25: Comparison of methods for NPV

Method	Dataset	Value
EpitopeTransfer	B. pertussis	0.320
BepiPred 3.0	B. pertussis	0.273
EpiDope	B. pertussis	0.256
EpitopeVec	B. pertussis	0.462
ESM-2	B. pertussis	0.292
NPTransfer	B. pertussis	0.062
EpitopeTransfer	C. difficile	0.931
BepiPred 3.0	C. difficile	0.906
EpiDope	C. difficile	0.907
EpitopeVec	C. difficile	1.000
ESM-2	C. difficile	0.910
NPTransfer	C. difficile	0.905
EpitopeTransfer	C. trachomatis	0.753
BepiPred 3.0	C. trachomatis	0.446
EpiDope	C. trachomatis	0.522
EpitopeVec	C. trachomatis	0.621
ESM-2	C. trachomatis	0.657
NPTransfer	C. trachomatis	0.689
EpitopeTransfer	Corynebacterium	0.730
BepiPred 3.0	Corynebacterium	0.569
EpiDope	Corynebacterium	0.571
EpitopeVec	Corynebacterium	0.717
ESM-2	Corynebacterium	0.667

Method	**Dataset**	**Value**
NPTransfer	Corynebacterium	0.633
EpitopeTransfer	E. coli	1.000
BepiPred 3.0	E. coli	0.215
EpiDope	E. coli	0.299
EpitopeVec	E. coli	0.268
ESM-2	E. coli	0.000
NPTransfer	E. coli	0.000
EpitopeTransfer	Enterobacteriaceae	0.752
BepiPred 3.0	Enterobacteriaceae	0.550
EpiDope	Enterobacteriaceae	0.550
EpitopeVec	Enterobacteriaceae	0.563
ESM-2	Enterobacteriaceae	0.683
NPTransfer	Enterobacteriaceae	0.714
EpitopeTransfer	Filoviridae	0.966
BepiPred 3.0	Filoviridae	0.949
EpiDope	Filoviridae	0.922
EpitopeVec	Filoviridae	0.958
ESM-2	Filoviridae	0.935
NPTransfer	Filoviridae	0.916
EpitopeTransfer	Human gammaherpesvirus 4	0.595
BepiPred 3.0	Human gammaherpesvirus 4	0.483
EpiDope	Human gammaherpesvirus 4	0.559
EpitopeVec	Human gammaherpesvirus 4	0.579
ESM-2	Human gammaherpesvirus 4	0.589
NPTransfer	Human gammaherpesvirus 4	0.583
EpitopeTransfer	Influenza A	0.393
BepiPred 3.0	Influenza A	0.321
EpiDope	Influenza A	0.258
EpitopeVec	Influenza A	0.364
ESM-2	Influenza A	0.362
NPTransfer	Influenza A	0.391
EpitopeTransfer	Lentivirus	0.750
BepiPred 3.0	Lentivirus	0.435

Method	**Dataset**	**Value**
EpiDope	Lentivirus	0.353
EpitopeVec	Lentivirus	0.372
ESM-2	Lentivirus	0.618
NPTransfer	Lentivirus	0.923
EpitopeTransfer	M. tuberculosis	0.478
BepiPred 3.0	M. tuberculosis	0.488
EpiDope	M. tuberculosis	0.486
EpitopeVec	M. tuberculosis	0.477
ESM-2	M. tuberculosis	0.523
NPTransfer	M. tuberculosis	0.462
EpitopeTransfer	Measles morbilivirus	0.667
BepiPred 3.0	Measles morbilivirus	0.395
EpiDope	Measles morbilivirus	0.552
EpitopeVec	Measles morbilivirus	0.593
ESM-2	Measles morbilivirus	0.355
NPTransfer	Measles morbilivirus	0.135
EpitopeTransfer	Mononegavirales	0.826
BepiPred 3.0	Mononegavirales	0.608
EpiDope	Mononegavirales	0.742
EpitopeVec	Mononegavirales	0.734
ESM-2	Mononegavirales	0.725
NPTransfer	Mononegavirales	0.788
EpitopeTransfer	Orthopox	0.874
BepiPred 3.0	Orthopox	0.897
EpiDope	Orthopox	0.898
EpitopeVec	Orthopox	0.749
ESM-2	Orthopox	0.848
NPTransfer	Orthopox	0.859
EpitopeTransfer	Ovolvulus	0.857
BepiPred 3.0	Ovolvulus	0.980
EpiDope	Ovolvulus	0.845
EpitopeVec	Ovolvulus	0.870
ESM-2	Ovolvulus	0.863

Method	**Dataset**	**Value**
NPTransfer	Ovolvulus	0.862
EpitopeTransfer	P. aeruginosa	0.432
BepiPred 3.0	P. aeruginosa	0.270
EpiDope	P. aeruginosa	0.303
EpitopeVec	P. aeruginosa	0.370
ESM-2	P. aeruginosa	0.431
NPTransfer	P. aeruginosa	0.435
EpitopeTransfer	P. falciparum	0.708
BepiPred 3.0	P. falciparum	0.405
EpiDope	P. falciparum	0.402
EpitopeVec	P. falciparum	0.387
ESM-2	P. falciparum	0.555
NPTransfer	P. falciparum	0.694
EpitopeTransfer	S. mansoni	0.740
BepiPred 3.0	S. mansoni	0.750
EpiDope	S. mansoni	0.757
EpitopeVec	S. mansoni	0.710
ESM-2	S. mansoni	0.838
NPTransfer	S. mansoni	0.721
EpitopeTransfer	Sars-cov-2	0.900
BepiPred 3.0	Sars-cov-2	0.899
EpiDope	Sars-cov-2	0.917
EpitopeVec	Sars-cov-2	0.927
ESM-2	Sars-cov-2	0.909
NPTransfer	Sars-cov-2	0.903
EpitopeTransfer	T. gondii	0.684
BepiPred 3.0	T. gondii	0.254
EpiDope	T. gondii	0.329
EpitopeVec	T. gondii	0.360
ESM-2	T. gondii	0.403
NPTransfer	T. gondii	0.000

Table 26: Comparison of methods for Sensitivity

Method	**Dataset**	**Value**
EpitopeTransfer	B. pertussis	0.523
BepiPred 3.0	B. pertussis	0.850
EpiDope	B. pertussis	0.187
EpitopeVec	B. pertussis	0.673
ESM-2	B. pertussis	0.570
NPTransfer	B. pertussis	0.720
EpitopeTransfer	C. difficile	0.463
BepiPred 3.0	C. difficile	0.000
EpiDope	C. difficile	0.000
EpitopeVec	C. difficile	1.000
ESM-2	C. difficile	0.049
NPTransfer	C. difficile	0.098
EpitopeTransfer	C. trachomatis	0.734
BepiPred 3.0	C. trachomatis	0.719
EpiDope	C. trachomatis	0.203
EpitopeVec	C. trachomatis	0.586
ESM-2	C. trachomatis	0.531
NPTransfer	C. trachomatis	0.711
EpitopeTransfer	Corynebacterium	0.815
BepiPred 3.0	Corynebacterium	0.185
EpiDope	Corynebacterium	0.167
EpitopeVec	Corynebacterium	0.759
ESM-2	Corynebacterium	0.852
NPTransfer	Corynebacterium	0.796
EpitopeTransfer	E. coli	1.000
BepiPred 3.0	E. coli	0.263
EpiDope	E. coli	0.205
EpitopeVec	E. coli	0.535
ESM-2	E. coli	1.000
NPTransfer	E. coli	1.000
EpitopeTransfer	Enterobacteriaceae	0.748
BepiPred 3.0	Enterobacteriaceae	0.380

Method	**Dataset**	**Value**
EpiDope	Enterobacteriaceae	0.108
EpitopeVec	Enterobacteriaceae	0.575
ESM-2	Enterobacteriaceae	0.658
NPTransfer	Enterobacteriaceae	0.861
EpitopeTransfer	Filoviridae	0.707
BepiPred 3.0	Filoviridae	0.793
EpiDope	Filoviridae	0.310
EpitopeVec	Filoviridae	0.793
ESM-2	Filoviridae	0.414
NPTransfer	Filoviridae	0.224
EpitopeTransfer	Human gammaherpesvirus 4	0.215
BepiPred 3.0	Human gammaherpesvirus 4	0.235
EpiDope	Human gammaherpesvirus 4	0.158
EpitopeVec	Human gammaherpesvirus 4	0.526
ESM-2	Human gammaherpesvirus 4	0.331
NPTransfer	Human gammaherpesvirus 4	0.192
EpitopeTransfer	Influenza A	0.755
BepiPred 3.0	Influenza A	0.384
EpiDope	Influenza A	0.118
EpitopeVec	Influenza A	0.815
ESM-2	Influenza A	0.743
NPTransfer	Influenza A	0.758
EpitopeTransfer	Lentivirus	0.946
BepiPred 3.0	Lentivirus	0.324
EpiDope	Lentivirus	0.345
EpitopeVec	Lentivirus	0.486
ESM-2	Lentivirus	0.912
NPTransfer	Lentivirus	0.993
EpitopeTransfer	M. tuberculosis	0.173
BepiPred 3.0	M. tuberculosis	0.165
EpiDope	M. tuberculosis	0.090
EpitopeVec	M. tuberculosis	0.505
ESM-2	M. tuberculosis	0.859

Method	Dataset	Value
NPTransfer	M. tuberculosis	0.202
EpitopeTransfer	Measles morbilivirus	0.959
BepiPred 3.0	Measles morbilivirus	0.271
EpiDope	Measles morbilivirus	0.265
EpitopeVec	Measles morbilivirus	0.706
ESM-2	Measles morbilivirus	0.765
NPTransfer	Measles morbilivirus	0.735
EpitopeTransfer	Mononegavirales	0.699
BepiPred 3.0	Mononegavirales	0.285
EpiDope	Mononegavirales	0.407
EpitopeVec	Mononegavirales	0.587
ESM-2	Mononegavirales	0.341
NPTransfer	Mononegavirales	0.595
EpitopeTransfer	Orthopox	0.552
BepiPred 3.0	Orthopox	0.534
EpiDope	Orthopox	0.759
EpitopeVec	Orthopox	0.259
ESM-2	Orthopox	0.483
NPTransfer	Orthopox	0.483
EpitopeTransfer	Ovolvulus	0.092
BepiPred 3.0	Ovolvulus	0.952
EpiDope	Ovolvulus	0.039
EpitopeVec	Ovolvulus	0.513
ESM-2	Ovolvulus	0.238
NPTransfer	Ovolvulus	0.204
EpitopeTransfer	P. aeruginosa	0.691
BepiPred 3.0	P. aeruginosa	0.000
EpiDope	P. aeruginosa	0.062
EpitopeVec	P. aeruginosa	0.642
ESM-2	P. aeruginosa	0.494
NPTransfer	P. aeruginosa	0.519
EpitopeTransfer	P. falciparum	0.877
BepiPred 3.0	P. falciparum	0.249

Method	Dataset	Value
EpiDope	P. falciparum	0.313
EpitopeVec	P. falciparum	0.654
ESM-2	P. falciparum	0.753
NPTransfer	P. falciparum	0.856
EpitopeTransfer	S. mansoni	0.426
BepiPred 3.0	S. mansoni	0.356
EpiDope	S. mansoni	0.312
EpitopeVec	S. mansoni	0.321
ESM-2	S. mansoni	0.987
NPTransfer	S. mansoni	0.099
EpitopeTransfer	Sars-cov-2	0.182
BepiPred 3.0	Sars-cov-2	0.178
EpiDope	Sars-cov-2	0.290
EpitopeVec	Sars-cov-2	0.620
ESM-2	Sars-cov-2	0.227
NPTransfer	Sars-cov-2	0.155
EpitopeTransfer	T. gondii	0.941
BepiPred 3.0	T. gondii	0.738
EpiDope	T. gondii	0.183
EpitopeVec	T. gondii	0.649
ESM-2	T. gondii	0.649
NPTransfer	T. gondii	0.995

Table 27: Comparison of methods for Specificity

Method	Dataset	Value
EpitopeTransfer	B. pertussis	0.571
BepiPred 3.0	B. pertussis	0.143
EpiDope	B. pertussis	0.714
EpitopeVec	B. pertussis	0.714
ESM-2	B. pertussis	0.452
NPTransfer	B. pertussis	0.048
EpitopeTransfer	C. difficile	0.746

Method	Dataset	Value
BepiPred 3.0	C. difficile	0.992
EpiDope	C. difficile	1.000
EpitopeVec	C. difficile	0.475
ESM-2	C. difficile	0.992
NPTransfer	C. difficile	0.884
EpitopeTransfer	C. trachomatis	0.831
BepiPred 3.0	C. trachomatis	0.233
EpiDope	C. trachomatis	0.896
EpitopeVec	C. trachomatis	0.699
ESM-2	C. trachomatis	0.924
NPTransfer	C. trachomatis	0.659
EpitopeTransfer	Corynebacterium	0.450
BepiPred 3.0	Corynebacterium	0.967
EpiDope	Corynebacterium	1.000
EpitopeVec	Corynebacterium	0.550
ESM-2	Corynebacterium	0.267
NPTransfer	Corynebacterium	0.317
EpitopeTransfer	E. coli	0.245
BepiPred 3.0	E. coli	0.594
EpiDope	E. coli	1.000
EpitopeVec	E. coli	0.500
ESM-2	E. coli	0.000
NPTransfer	E. coli	0.000
EpitopeTransfer	Enterobacteriaceae	0.679
BepiPred 3.0	Enterobacteriaceae	0.672
EpiDope	Enterobacteriaceae	0.965
EpitopeVec	Enterobacteriaceae	0.485
ESM-2	Enterobacteriaceae	0.653
NPTransfer	Enterobacteriaceae	0.307
EpitopeTransfer	Filoviridae	0.947
BepiPred 3.0	Filoviridae	0.439
EpiDope	Filoviridae	0.926
EpitopeVec	Filoviridae	0.541

Method	**Dataset**	**Value**
ESM-2	Filoviridae	0.949
NPTransfer	Filoviridae	0.957
EpitopeTransfer	Human gammaherpesvirus 4	0.963
BepiPred 3.0	Human gammaherpesvirus 4	0.595
EpiDope	Human gammaherpesvirus 4	0.889
EpitopeVec	Human gammaherpesvirus 4	0.545
ESM-2	Human gammaherpesvirus 4	0.799
NPTransfer	Human gammaherpesvirus 4	0.944
EpitopeTransfer	Influenza A	0.476
BepiPred 3.0	Influenza A	0.873
EpiDope	Influenza A	0.921
EpitopeVec	Influenza A	0.317
ESM-2	Influenza A	0.439
NPTransfer	Influenza A	0.466
EpitopeTransfer	Lentivirus	0.312
BepiPred 3.0	Lentivirus	1.000
EpiDope	Lentivirus	0.688
EpitopeVec	Lentivirus	0.584
ESM-2	Lentivirus	0.273
NPTransfer	Lentivirus	0.156
EpitopeTransfer	M. tuberculosis	0.815
BepiPred 3.0	M. tuberculosis	0.856
EpiDope	M. tuberculosis	0.928
EpitopeVec	M. tuberculosis	0.487
ESM-2	M. tuberculosis	0.166
NPTransfer	M. tuberculosis	0.738
EpitopeTransfer	Measles morbilivirus	0.073
BepiPred 3.0	Measles morbilivirus	0.422
EpiDope	Measles morbilivirus	0.802
EpitopeVec	Measles morbilivirus	0.380
ESM-2	Measles morbilivirus	0.115
NPTransfer	Measles morbilivirus	0.036
EpitopeTransfer	Mononegavirales	0.744

Method	Dataset	Value
BepiPred 3.0	Mononegavirales	0.576
EpiDope	Mononegavirales	0.884
EpitopeVec	Mononegavirales	0.592
ESM-2	Mononegavirales	0.902
NPTransfer	Mononegavirales	0.783
EpitopeTransfer	Orthopox	0.662
BepiPred 3.0	Orthopox	0.864
EpiDope	Orthopox	0.452
EpitopeVec	Orthopox	0.471
ESM-2	Orthopox	0.618
NPTransfer	Orthopox	0.673
EpitopeTransfer	Ovolvulus	0.963
BepiPred 3.0	Ovolvulus	0.422
EpiDope	Ovolvulus	0.920
EpitopeVec	Ovolvulus	0.577
ESM-2	Ovolvulus	0.848
NPTransfer	Ovolvulus	0.877
EpitopeTransfer	P. aeruginosa	0.576
BepiPred 3.0	P. aeruginosa	0.909
EpiDope	P. aeruginosa	1.000
EpitopeVec	P. aeruginosa	0.515
ESM-2	P. aeruginosa	0.939
NPTransfer	P. aeruginosa	0.909
EpitopeTransfer	P. falciparum	0.495
BepiPred 3.0	P. falciparum	0.851
EpiDope	P. falciparum	0.769
EpitopeVec	P. falciparum	0.363
ESM-2	P. falciparum	0.512
NPTransfer	P. falciparum	0.544
EpitopeTransfer	S. mansoni	0.647
BepiPred 3.0	S. mansoni	0.768
EpiDope	S. mansoni	0.851
EpitopeVec	S. mansoni	0.662

Method	Dataset	Value
ESM-2	S. mansoni	0.027
NPTransfer	S. mansoni	0.923
EpitopeTransfer	Sars-cov-2	0.840
BepiPred 3.0	Sars-cov-2	0.836
EpiDope	Sars-cov-2	0.894
EpitopeVec	Sars-cov-2	0.547
ESM-2	Sars-cov-2	0.879
NPTransfer	Sars-cov-2	0.894
EpitopeTransfer	T. gondii	0.286
BepiPred 3.0	T. gondii	0.198
EpiDope	T. gondii	0.890
EpitopeVec	T. gondii	0.440
ESM-2	T. gondii	0.527
NPTransfer	T. gondii	0.000

Metric	EpitopeTrans	BepiPred 3.0	EpiDope	EpitopeVec	ESM-2	NPTransfer
AUC	0.686 (±0.028)	0.503 (±0.035)	0.634 (±0.032)	0.602 (±0.027)	0.657 (±0.028)	0.630 (±0.032)
F1	0.570 (±0.055)	0.363 (±0.045)	0.276 (±0.029)	0.509 (±0.044)	0.543 (±0.048)	0.515 (±0.059)
MCC	0.260 (±0.041)	0.041 (±0.044)	0.118 (±0.025)	0.112 (±0.029)	0.167 (±0.036)	0.122 (±0.046)
BACC	0.621 (±0.020)	0.527 (±0.021)	0.548 (±0.011)	0.566 (±0.019)	0.578 (±0.017)	0.558 (±0.021)
PPV	0.580 (±0.052)	0.462 (±0.066)	0.581 (±0.065)	0.496 (±0.055)	0.569 (±0.051)	0.529 (±0.054)
NPV	0.718 (±0.043)	0.555 (±0.057)	0.571 (±0.054)	0.604 (±0.050)	0.608 (±0.055)	0.584 (±0.071)
Sensit.	0.625 (±0.064)	0.393 (±0.062)	0.226 (±0.037)	0.610 (±0.037)	0.593 (±0.061)	0.560 (±0.073)
Specif.	0.616 (±0.057)	0.660 (±0.061)	0.869 (±0.030)	0.522 (±0.023)	0.564 (±0.075)	0.556 (±0.080)

Table 28: Summary of average test set performance (*mean ±standard error*) for EpitopeTransfer (proposed method) and five baseline methods across 20 selected datasets. Each row corresponds to a performance evaluation metric, and the values indicate the mean performance of each method over all datasets.

Appendix F

The statistical comparisons for each method on each method is presented below, following the same evaluation framework described in Appendix F. Statistical comparisons of median values for each performance metric were performed to assess the significance of differences between EpitopeTransfer (ESM-2) and the baseline methods. The Wilcoxon signed rank test was used as the primary statistical method to evaluate whether observed differences in medians were statistically meaningful. To account for multiple comparisons, the p-values derived from the tests were adjusted for the false discovery rate using the Benjamini-Hochberg correction.

The analysis includes the following columns: “Pair”, which specifies the pairwise comparison (e.g., EpitopeTransfer vs. Baseline); “Medians of diff”, representing the median of paired differences (95% CI); “p-value”, which indicates the unadjusted significance level from the Wilcoxon test; “FDR”, which represents the adjusted p-value following the Benjamini-Hochberg procedure; and “Significant”, which highlights whether the corrected p-value falls below the significance threshold of 0.05.

Comparison Results for AUC

Metric	Pair	Medians of diff.	p-value	FDR	Signif.
AUC	EpitopeTransfer vs BepiPred 3	0.166 (0.083, 0.270)	0.00032	0.00081	Yes
AUC	EpitopeTransfer vs EpiDope	0.052 (-0.003, 0.112)	0.07585	0.07585	No
AUC	EpitopeTransfer vs EpitopeVec	0.076 (0.021, 0.162)	0.01923	0.02404	Yes
AUC	EpitopeTransfer vs ESM-2	0.025 (0.009, 0.053)	0.00639	0.01065	Yes
AUC	EpitopeTransfer vs NPTransfer	0.047 (0.019, 0.084)	0.00032	0.00081	Yes

Table 29: Comparison Results for AUC

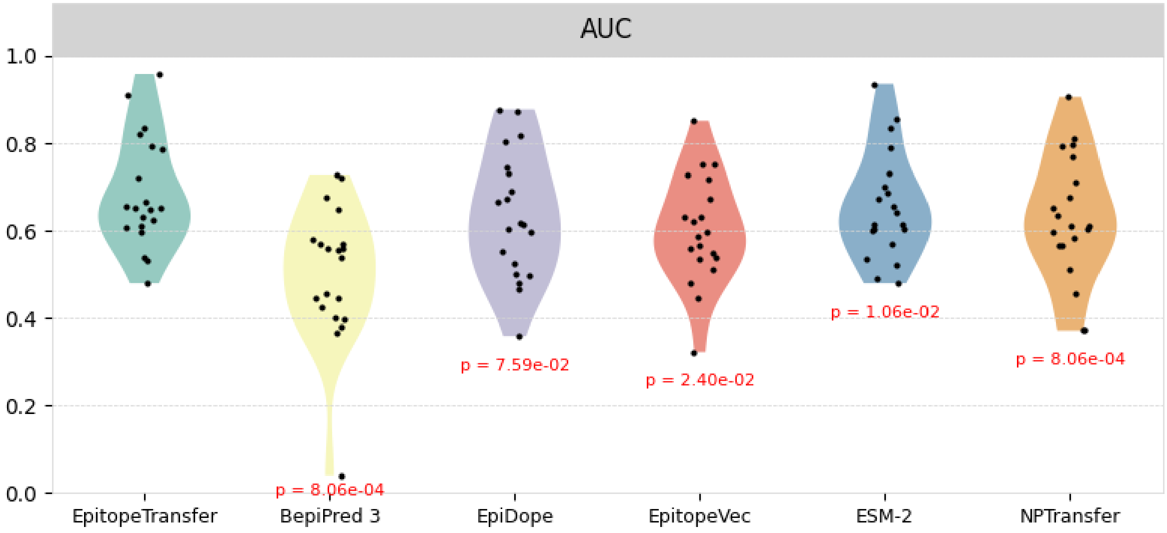


Figure 10: Performance plot for the AUC metric

Comparison Results for Balanced Accuracy

Metric	Pair	Medians of diff.	p-value	FDR	Signif.
BACC	EpitopeTransfer vs BepiPred 3	0.091 (0.028, 0.162)	0.00639	0.00799	Yes
BACC	EpitopeTransfer vs EpiDope	0.074 (0.032, 0.109)	0.00085	0.00213	Yes
BACC	EpitopeTransfer vs EpitopeVec	0.055 (0.009, 0.108)	0.03999	0.03999	Yes
BACC	EpitopeTransfer vs ESM-2	0.044 (0.020, 0.066)	0.00271	0.00452	Yes
BACC	EpitopeTransfer vs NPTransfer	0.060 (0.024, 0.098)	0.00085	0.00213	Yes

Table 30: Comparison Results for Balanced Accuracy

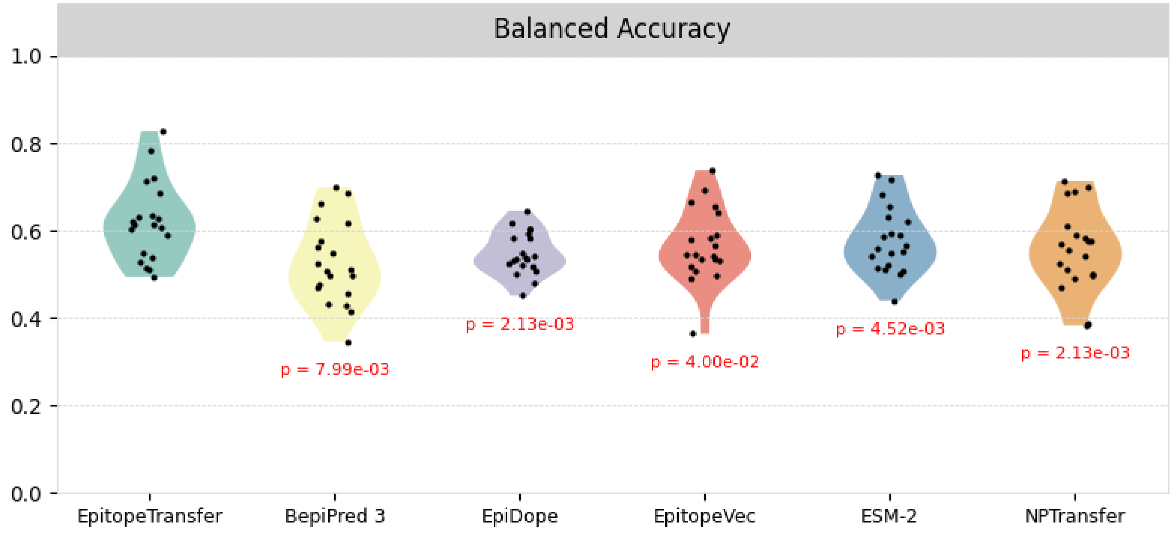


Figure 11: Performance plot for the Balanced Accuracy metric

Comparison Results for F1

Metric	Pair	Medians of diff.	p-value	FDR	Signif.
F1	EpitopeTransfer vs BepiPred 3	0.210 (0.096, 0.331)	0.00102	0.00254	Yes
F1	EpitopeTransfer vs EpiDope	0.306 (0.187, 0.413)	0.00005	0.00024	Yes
F1	EpitopeTransfer vs EpitopeVec	0.064 (-0.015, 0.149)	0.12309	0.12309	No
F1	EpitopeTransfer vs ESM-2	0.043 (-0.011, 0.087)	0.11399	0.12309	No
F1	EpitopeTransfer vs NPTransfer	0.039 (0.009, 0.083)	0.01069	0.01781	Yes

Table 31: Comparison Results for F1

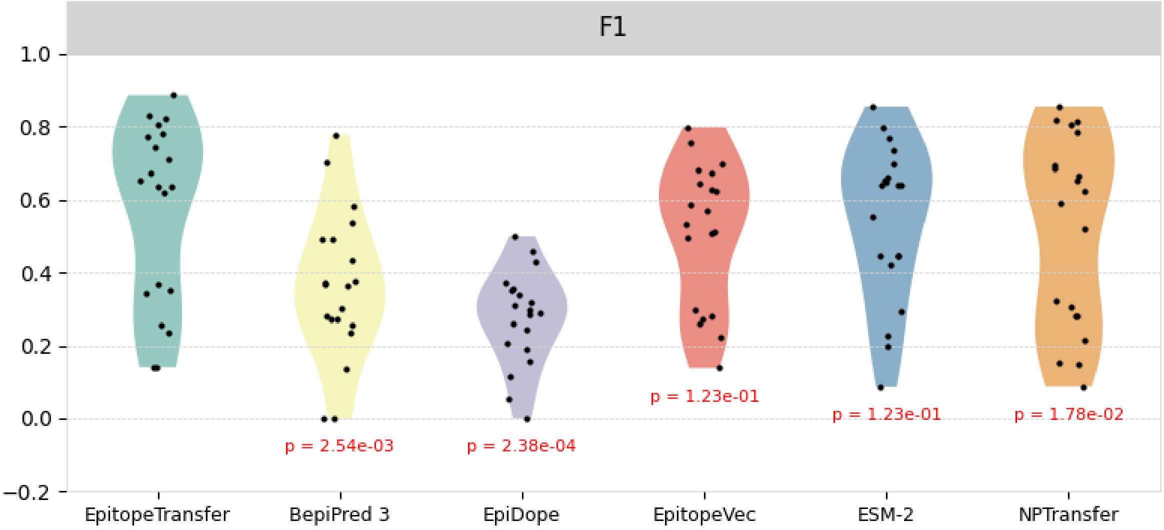


Figure 12: Performance plot for the F1 metric

Comparison Results for MCC

Metric	Pair	Medians of diff.	p-value	FDR	Signif.
MCC	EpitopeTransfer vs BepiPred 3	0.219 (0.082, 0.371)	0.00730	0.00730	Yes
MCC	EpitopeTransfer vs EpiDope	0.148 (0.063, 0.219)	0.00199	0.00496	Yes
MCC	EpitopeTransfer vs EpitopeVec	0.155 (0.048, 0.247)	0.00730	0.00730	Yes
MCC	EpitopeTransfer vs ESM-2	0.087 (0.035, 0.134)	0.00365	0.00609	Yes
MCC	EpitopeTransfer vs NPTransfer	0.123 (0.042, 0.214)	0.00048	0.00241	Yes

Table 32: Comparison Results for MCC

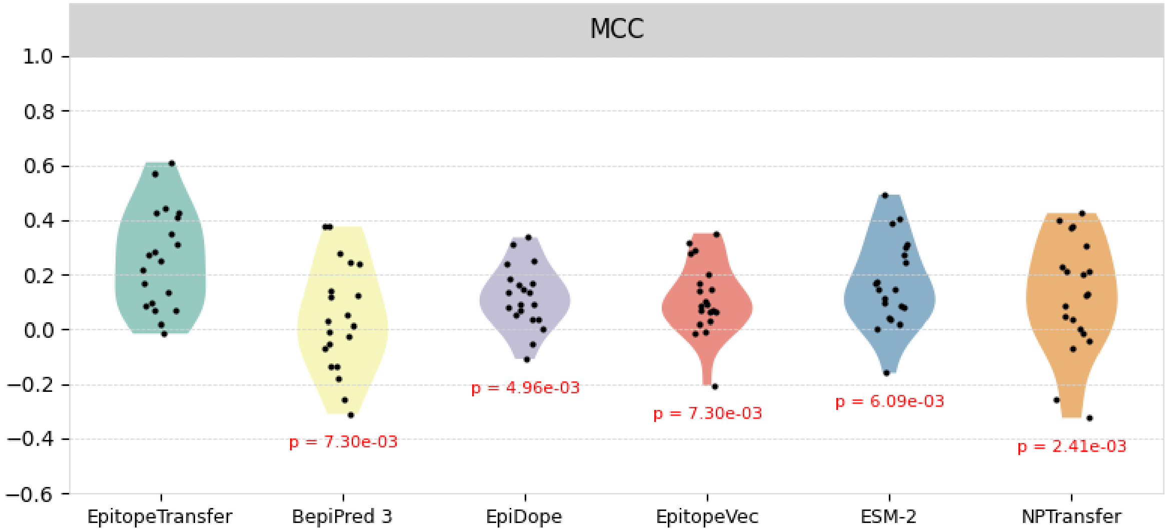


Figure 13: Performance plot for the MCC metric

Comparison Results for NPV

Metric	Pair	Medians of diff.	p-value	FDR	Signif.
NPV	EpitopeTransfer vs BepiPred 3	0.146 (0.060, 0.234)	0.00071	0.00177	Yes
NPV	EpitopeTransfer vs EpiDope	0.117 (0.053, 0.209)	0.00017	0.00084	Yes
NPV	EpitopeTransfer vs EpitopeVec	0.073 (0.015, 0.177)	0.00730	0.00730	Yes
NPV	EpitopeTransfer vs ESM-2	0.059 (0.016, 0.141)	0.00365	0.00457	Yes
NPV	EpitopeTransfer vs NPTransfer	0.038 (0.014, 0.265)	0.00169	0.00282	Yes

Table 33: Comparison Results for NPV

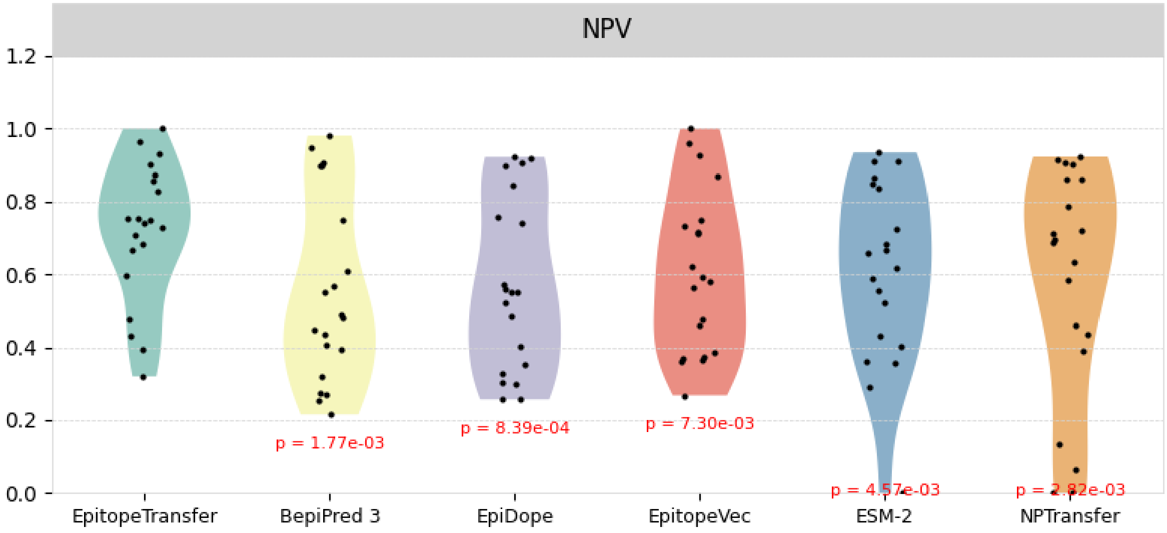


Figure 14: Performance plot for the NPV metric

Comparison Results for PPV

Metric	Pair	Medians of diff.	p-value	FDR	Signif.
PPV	EpitopeTransfer vs BepiPred 3	0.096 (-0.020, 0.236)	0.08969	0.14949	No
PPV	EpitopeTransfer vs EpiDope	-0.005 (-0.079, 0.083)	0.98544	0.98544	No
PPV	EpitopeTransfer vs EpitopeVec	0.072 (0.014, 0.140)	0.00315	0.01055	Yes
PPV	EpitopeTransfer vs ESM-2	0.019 (-0.021, 0.048)	0.43043	0.53804	No
PPV	EpitopeTransfer vs NPTransfer	0.049 (0.019, 0.079)	0.00422	0.01055	Yes

Table 34: Comparison Results for PPV

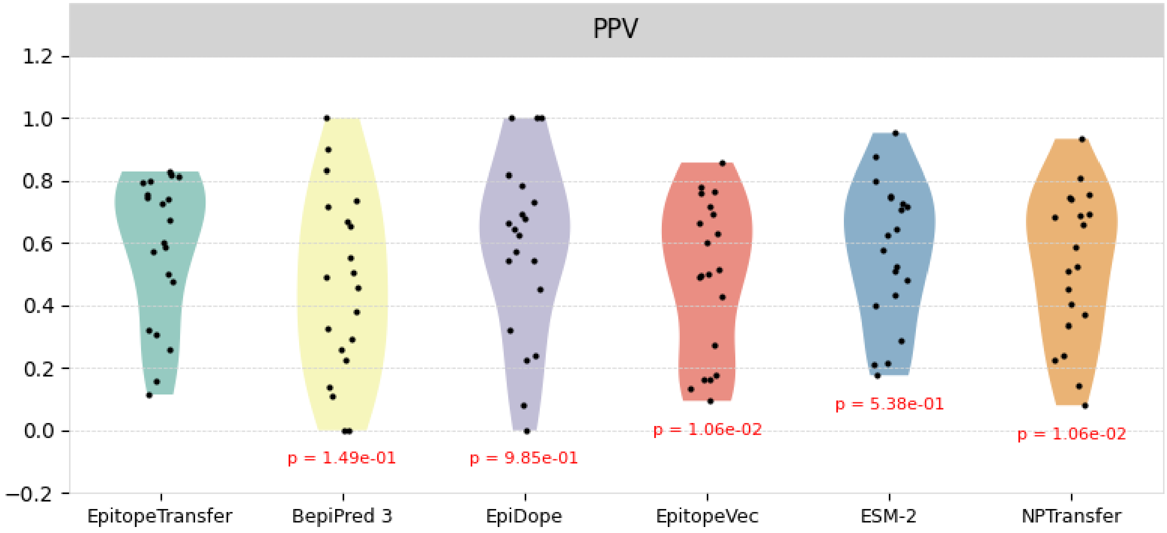


Figure 15: Performance plot for the PPV metric

Comparison Results for Sensitivity

Metric	Pair	Medians of diff.	p-value	FDR	Signif.
Sensit.	EpitopeTransfer vs BepiPred 3	0.271 (0.022, 0.416)	0.01531	0.03828	Yes
Sensit.	EpitopeTransfer vs EpiDope	0.405 (0.258, 0.583)	0.00008	0.00041	Yes
Sensit.	EpitopeTransfer vs EpitopeVec	0.024 (-0.131, 0.174)	0.78413	0.78413	No
Sensit.	EpitopeTransfer vs ESM-2	0.074 (-0.073, 0.181)	0.26844	0.33555	No
Sensit.	EpitopeTransfer vs NPTransfer	0.037 (-0.018, 0.162)	0.23517	0.33555	No

Table 35: Comparison Results for Sensitivity

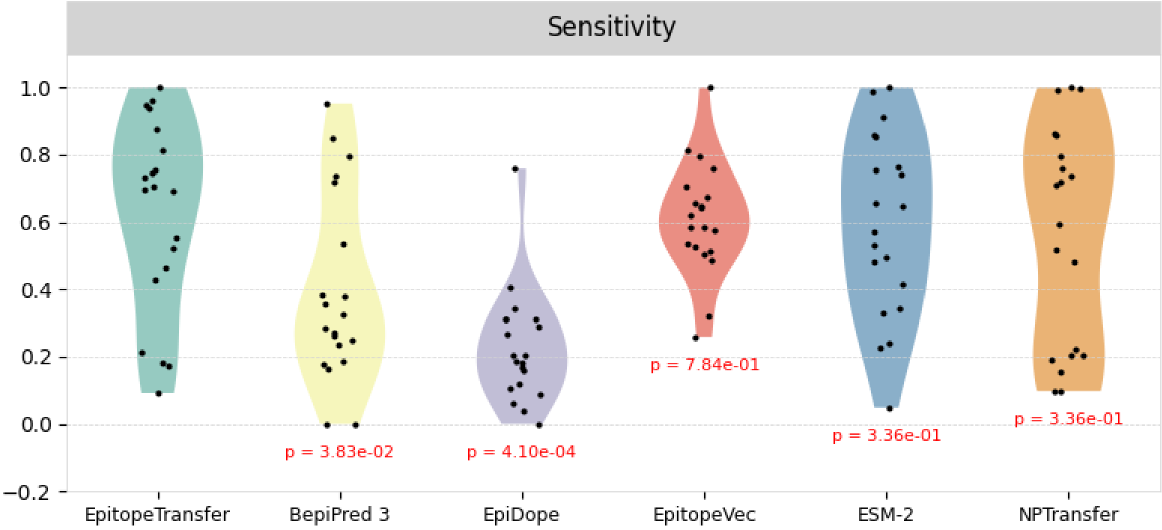


Figure 16: Performance plot for the Sensitivity metric

Comparison Results for Specificity

Metric	Pair	Medians of diff.	p-value	FDR	Signif.
Specif.	EpitopeTransfer vs BepiPred 3	-0.057 (-0.255, 0.128)	0.70118	0.70118	No
Specif.	EpitopeTransfer vs EpiDope	-0.247 (-0.377, -0.113)	0.00048	0.00241	Yes
Specif.	EpitopeTransfer vs EpitopeVec	0.097 (-0.015, 0.223)	0.09731	0.24327	No
Specif.	EpitopeTransfer vs ESM-2	0.034 (-0.063, 0.139)	0.49801	0.62251	No
Specif.	EpitopeTransfer vs NPTransfer	0.053 (-0.026, 0.152)	0.21617	0.36028	No

Table 36: Comparison Results for Specificity

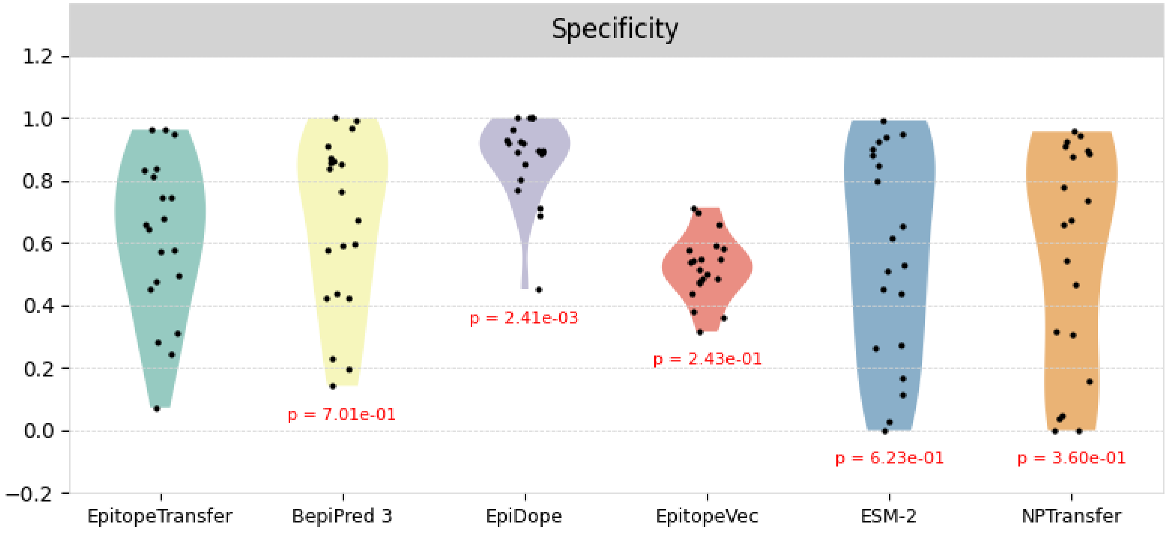


Figure 17: Performance plot for the Specificity metric

Summary of Comparison Results

Metric	Pair	Medians of diff.	p-value	FDR	Signif.
AUC	EpitopeTransfer vs BepiPred 3	0.166 (0.083, 0.270)	0.00032	0.00081	Yes
AUC	EpitopeTransfer vs EpiDope	0.052 (-0.003, 0.112)	0.07585	0.07585	No
AUC	EpitopeTransfer vs EpitopeVec	0.076 (0.021, 0.162)	0.01923	0.02404	Yes
AUC	EpitopeTransfer vs ESM-2	0.025 (0.009, 0.053)	0.00639	0.01065	Yes
AUC	EpitopeTransfer vs NPTransfer	0.047 (0.019, 0.084)	0.00032	0.00081	Yes
BACC	EpitopeTransfer vs BepiPred 3	0.091 (0.028, 0.162)	0.00639	0.00799	Yes
BACC	EpitopeTransfer vs EpiDope	0.074 (0.032, 0.109)	0.00085	0.00213	Yes
BACC	EpitopeTransfer vs EpitopeVec	0.055 (0.009, 0.108)	0.03999	0.03999	Yes
BACC	EpitopeTransfer vs ESM-2	0.044 (0.020, 0.066)	0.00271	0.00452	Yes
BACC	EpitopeTransfer vs NPTransfer	0.060 (0.024, 0.098)	0.00085	0.00213	Yes
F1	EpitopeTransfer vs BepiPred 3	0.210 (0.096, 0.331)	0.00102	0.00254	Yes
F1	EpitopeTransfer vs EpiDope	0.306 (0.187, 0.413)	0.00005	0.00024	Yes
F1	EpitopeTransfer vs EpitopeVec	0.064 (-0.015, 0.149)	0.12309	0.12309	No
F1	EpitopeTransfer vs ESM-2	0.043 (-0.011, 0.087)	0.11399	0.12309	No
F1	EpitopeTransfer vs NPTransfer	0.039 (0.009, 0.083)	0.01069	0.01781	Yes
MCC	EpitopeTransfer vs BepiPred 3	0.219 (0.082, 0.371)	0.00730	0.00730	Yes
MCC	EpitopeTransfer vs EpiDope	0.148 (0.063, 0.219)	0.00199	0.00496	Yes
MCC	EpitopeTransfer vs EpitopeVec	0.155 (0.048, 0.247)	0.00730	0.00730	Yes

MCC	EpitopeTransfer vs ESM-2	0.087 (0.035, 0.134)	0.00365	0.00609	Yes
MCC	EpitopeTransfer vs NPTransfer	0.123 (0.042, 0.214)	0.00048	0.00241	Yes
NPV	EpitopeTransfer vs BepiPred 3	0.146 (0.060, 0.234)	0.00071	0.00177	Yes
NPV	EpitopeTransfer vs EpiDope	0.117 (0.053, 0.209)	0.00017	0.00084	Yes
NPV	EpitopeTransfer vs EpitopeVec	0.073 (0.015, 0.177)	0.00730	0.00730	Yes
NPV	EpitopeTransfer vs ESM-2	0.059 (0.016, 0.141)	0.00365	0.00457	Yes
NPV	EpitopeTransfer vs NPTransfer	0.038 (0.014, 0.265)	0.00169	0.00282	Yes
PPV	EpitopeTransfer vs BepiPred 3	0.096 (-0.020, 0.236)	0.08969	0.14949	No
PPV	EpitopeTransfer vs EpiDope	-0.005 (-0.079, 0.083)	0.98544	0.98544	No
PPV	EpitopeTransfer vs EpitopeVec	0.072 (0.014, 0.140)	0.00315	0.01055	Yes
PPV	EpitopeTransfer vs ESM-2	0.019 (-0.021, 0.048)	0.43043	0.53804	No
PPV	EpitopeTransfer vs NPTransfer	0.049 (0.019, 0.079)	0.00422	0.01055	Yes
Sensit.	EpitopeTransfer vs BepiPred 3	0.271 (0.022, 0.416)	0.01531	0.03828	Yes
Sensit.	EpitopeTransfer vs EpiDope	0.405 (0.258, 0.583)	0.00008	0.00041	Yes
Sensit.	EpitopeTransfer vs EpitopeVec	0.024 (-0.131, 0.174)	0.78413	0.78413	No
Sensit.	EpitopeTransfer vs ESM-2	0.074 (-0.073, 0.181)	0.26844	0.33555	No
Sensit.	EpitopeTransfer vs NPTransfer	0.037 (-0.018, 0.162)	0.23517	0.33555	No
Specif.	EpitopeTransfer vs BepiPred 3	-0.057 (-0.255, 0.128)	0.70118	0.70118	No
Specif.	EpitopeTransfer vs EpiDope	-0.247 (-0.377, -0.113)	0.00048	0.00241	Yes
Specif.	EpitopeTransfer vs EpitopeVec	0.097 (-0.015, 0.223)	0.09731	0.24327	No
Specif.	EpitopeTransfer vs ESM-2	0.034 (-0.063, 0.139)	0.49801	0.62251	No
Specif.	EpitopeTransfer vs NPTransfer	0.053 (-0.026, 0.152)	0.21617	0.36028	No

Table 38: Summary of Comparison Results across All Metrics

Appendix G

The estimated performance for each method on each dataset is presented. *Method* refers to the employed approach, including the primary method, **EpitopeTransfer**, its generalized variant, and the baseline. *Dataset* corresponds to the data from 17 specific taxa, and *Value* represents the value of each presented metric. The evaluated metrics include **AUC** (Area Under the Curve), **F1** score, **MCC** (Matthews Correlation Coefficient), **Accuracy**, **PPV** (Positive Predictive Value), **NPV** (Negative Predictive Value), **Sensitivity**, and **Specificity**.

Table 39: Comparison of methods for AUC

Method	Dataset	Value
EpitopeTransfer	C. difficile	0.590
Baseline	C. difficile	0.519
EpitopeTransfer	C. trachomatis	0.779
Baseline	C. trachomatis	0.680
EpitopeTransfer	Corynebacterium	0.514
Baseline	Corynebacterium	0.637
EpitopeTransfer	Enterobacteriaceae	0.713
Baseline	Enterobacteriaceae	0.536
EpitopeTransfer	Firoviridae	0.940
Baseline	Firoviridae	0.889
EpitopeTransfer	Human gammaherpesvirus 4	0.640
Baseline	Human gammaherpesvirus 4	0.677
EpitopeTransfer	Influenza A	0.694
Baseline	Influenza A	0.717
EpitopeTransfer	Lentivirus	0.767
Baseline	Lentivirus	0.635
EpitopeTransfer	Measles morbilivirus	0.589
Baseline	Measles morbilivirus	0.406
EpitopeTransfer	Mononegavirales	0.780
Baseline	Mononegavirales	0.764
EpitopeTransfer	Orthopoxvirus	0.689
Baseline	Orthopoxvirus	0.493
EpitopeTransfer	Ovolvulus	0.674

Method	Dataset	Value
Baseline	Ovolvulus	0.583
EpitopeTransfer	P. aeruginosa	0.637
Baseline	P. aeruginosa	0.376
EpitopeTransfer	P. falciparum	0.815
Baseline	P. falciparum	0.777
EpitopeTransfer	S. mansoni	0.547
Baseline	S. mansoni	0.615
EpitopeTransfer	Sars-cov-2	0.658
Baseline	Sars-cov-2	0.575
EpitopeTransfer	T. gondii	0.845
Baseline	T. gondii	0.747

Table 40: Comparison of methods for F1

Method	Dataset	Value
EpitopeTransfer	C. difficile	0.157
Baseline	C. difficile	0.000
EpitopeTransfer	C. trachomatis	0.722
Baseline	C. trachomatis	0.619
EpitopeTransfer	Corynebacterium	0.659
Baseline	Corynebacterium	0.036
EpitopeTransfer	Enterobacteriaceae	0.655
Baseline	Enterobacteriaceae	0.479
EpitopeTransfer	Firoviridae	0.592
Baseline	Firoviridae	0.431
EpitopeTransfer	Human gammaherpesvirus 4	0.356
Baseline	Human gammaherpesvirus 4	0.319
EpitopeTransfer	Influenza A	0.601
Baseline	Influenza A	0.547
EpitopeTransfer	Lentivirus	0.667
Baseline	Lentivirus	0.752
EpitopeTransfer	Measles morbilivirus	0.526
Baseline	Measles morbilivirus	0.424

Method	Dataset	Value
EpitopeTransfer	Mononegavirales	0.675
Baseline	Mononegavirales	0.563
EpitopeTransfer	Orthopoxvirus	0.650
Baseline	Orthopoxvirus	0.548
EpitopeTransfer	Ovolvulus	0.324
Baseline	Ovolvulus	0.261
EpitopeTransfer	P. aeruginosa	0.825
Baseline	P. aeruginosa	0.787
EpitopeTransfer	P. falciparum	0.707
Baseline	P. falciparum	0.562
EpitopeTransfer	S. mansoni	0.158
Baseline	S. mansoni	0.448
EpitopeTransfer	Sars-cov-2	0.252
Baseline	Sars-cov-2	0.202
EpitopeTransfer	T. gondii	0.816
Baseline	T. gondii	0.738

Table 41: Comparison of methods for MCC

Method	Dataset	Value
EpitopeTransfer	C. difficile	0.063
Baseline	C. difficile	-0.038
EpitopeTransfer	C. trachomatis	0.469
Baseline	C. trachomatis	0.240
EpitopeTransfer	Corynebacterium	0.181
Baseline	Corynebacterium	0.099
EpitopeTransfer	Enterobacteriaceae	0.235
Baseline	Enterobacteriaceae	0.156
EpitopeTransfer	Firoviridae	0.595
Baseline	Firoviridae	0.368
EpitopeTransfer	Human gammaherpesvirus 4	0.200
Baseline	Human gammaherpesvirus 4	0.178
EpitopeTransfer	Influenza A	0.384

Method	Dataset	Value
Baseline	Influenza A	0.317
EpitopeTransfer	Lentivirus	0.256
Baseline	Lentivirus	0.227
EpitopeTransfer	Measles morbilivirus	0.100
Baseline	Measles morbilivirus	-0.105
EpitopeTransfer	Mononegavirales	0.483
Baseline	Mononegavirales	0.339
EpitopeTransfer	Orthopoxvirus	0.372
Baseline	Orthopoxvirus	0.056
EpitopeTransfer	Ovolvulus	0.167
Baseline	Ovolvulus	0.043
EpitopeTransfer	P. aeruginosa	-0.060
Baseline	P. aeruginosa	-0.163
EpitopeTransfer	P. falciparum	0.422
Baseline	P. falciparum	0.343
EpitopeTransfer	S. mansoni	-0.003
Baseline	S. mansoni	0.157
EpitopeTransfer	Sars-cov-2	0.143
Baseline	Sars-cov-2	0.077
EpitopeTransfer	T. gondii	0.217
Baseline	T. gondii	0.326

Table 42: Comparison of methods for Balanced Accuracy

Method	Dataset	Value
EpitopeTransfer	C. difficile	0.534
Baseline	C. difficile	0.492
EpitopeTransfer	C. trachomatis	0.733
Baseline	C. trachomatis	0.620
EpitopeTransfer	Corynebacterium	0.533
Baseline	Corynebacterium	0.509
EpitopeTransfer	Enterobacteriaceae	0.607
Baseline	Enterobacteriaceae	0.574

Method	Dataset	Value
EpitopeTransfer	Firoviridae	0.922
Baseline	Firoviridae	0.739
EpitopeTransfer	Human gammaherpesvirus 4	0.574
Baseline	Human gammaherpesvirus 4	0.562
EpitopeTransfer	Influenza A	0.708
Baseline	Influenza A	0.668
EpitopeTransfer	Lentivirus	0.635
Baseline	Lentivirus	0.609
EpitopeTransfer	Measles morbilivirus	0.547
Baseline	Measles morbilivirus	0.448
EpitopeTransfer	Mononegavirales	0.751
Baseline	Mononegavirales	0.669
EpitopeTransfer	Orthopoxvirus	0.657
Baseline	Orthopoxvirus	0.522
EpitopeTransfer	Ovolvulus	0.615
Baseline	Ovolvulus	0.529
EpitopeTransfer	P. aeruginosa	0.494
Baseline	P. aeruginosa	0.457
EpitopeTransfer	P. falciparum	0.716
Baseline	P. falciparum	0.661
EpitopeTransfer	S. mansoni	0.499
Baseline	S. mansoni	0.587
EpitopeTransfer	Sars-cov-2	0.607
Baseline	Sars-cov-2	0.555
EpitopeTransfer	T. gondii	0.575
Baseline	T. gondii	0.675

Table 43: Comparison of methods for PPV

Method	Dataset	Value
EpitopeTransfer	C. difficile	0.146
Baseline	C. difficile	0.000
EpitopeTransfer	C. trachomatis	0.764

Method	**Dataset**	**Value**
Baseline	C. trachomatis	0.629
EpitopeTransfer	Corynebacterium	0.491
Baseline	Corynebacterium	1.000
EpitopeTransfer	Enterobacteriaceae	0.546
Baseline	Enterobacteriaceae	0.581
EpitopeTransfer	Firoviridae	0.420
Baseline	Firoviridae	0.330
EpitopeTransfer	Human gammaherpesvirus 4	0.683
Baseline	Human gammaherpesvirus 4	0.673
EpitopeTransfer	Influenza A	0.988
Baseline	Influenza A	0.960
EpitopeTransfer	Lentivirus	0.782
Baseline	Lentivirus	0.728
EpitopeTransfer	Measles morbilivirus	0.402
Baseline	Measles morbilivirus	0.333
EpitopeTransfer	Mononegavirales	0.534
Baseline	Mononegavirales	0.567
EpitopeTransfer	Orthopoxvirus	0.487
Baseline	Orthopoxvirus	0.405
EpitopeTransfer	Ovolvulus	0.222
Baseline	Ovolvulus	0.162
EpitopeTransfer	P. aeruginosa	0.708
Baseline	P. aeruginosa	0.692
EpitopeTransfer	P. falciparum	0.854
Baseline	P. falciparum	0.883
EpitopeTransfer	S. mansoni	0.281
Baseline	S. mansoni	0.360
EpitopeTransfer	Sars-cov-2	0.171
Baseline	Sars-cov-2	0.143
EpitopeTransfer	T. gondii	0.726
Baseline	T. gondii	0.823

Table 44: Comparison of methods for NPV

Method	Dataset	Value
EpitopeTransfer	C. difficile	0.913
Baseline	C. difficile	0.905
EpitopeTransfer	C. trachomatis	0.707
Baseline	C. trachomatis	0.611
EpitopeTransfer	Corynebacterium	1.000
Baseline	Corynebacterium	0.531
EpitopeTransfer	Enterobacteriaceae	0.711
Baseline	Enterobacteriaceae	0.585
EpitopeTransfer	Firoviridae	1.000
Baseline	Firoviridae	0.952
EpitopeTransfer	Human gammaherpesvirus 4	0.589
Baseline	Human gammaherpesvirus 4	0.581
EpitopeTransfer	Influenza A	0.366
Baseline	Influenza A	0.340
EpitopeTransfer	Lentivirus	0.461
Baseline	Lentivirus	0.507
EpitopeTransfer	Measles morbilivirus	0.703
Baseline	Measles morbilivirus	0.561
EpitopeTransfer	Mononegavirales	0.931
Baseline	Mononegavirales	0.773
EpitopeTransfer	Orthopoxvirus	0.955
Baseline	Orthopoxvirus	0.667
EpitopeTransfer	Ovolvulus	0.899
Baseline	Ovolvulus	0.870
EpitopeTransfer	P. aeruginosa	0.000
Baseline	P. aeruginosa	0.000
EpitopeTransfer	P. falciparum	0.557
Baseline	P. falciparum	0.482
EpitopeTransfer	S. mansoni	0.715
Baseline	S. mansoni	0.782
EpitopeTransfer	Sars-cov-2	0.925
Baseline	Sars-cov-2	0.911

Method	Dataset	Value
EpitopeTransfer	T. gondii	0.588
Baseline	T. gondii	0.481

Table 45: Comparison of methods for Sensitivity

Method	Dataset	Value
EpitopeTransfer	C. difficile	0.171
Baseline	C. difficile	0.000
EpitopeTransfer	C. trachomatis	0.684
Baseline	C. trachomatis	0.609
EpitopeTransfer	Corynebacterium	1.000
Baseline	Corynebacterium	0.019
EpitopeTransfer	Enterobacteriaceae	0.818
Baseline	Enterobacteriaceae	0.407
EpitopeTransfer	Firoviridae	1.000
Baseline	Firoviridae	0.621
EpitopeTransfer	Human gammaherpesvirus 4	0.241
Baseline	Human gammaherpesvirus 4	0.209
EpitopeTransfer	Influenza A	0.432
Baseline	Influenza A	0.383
EpitopeTransfer	Lentivirus	0.581
Baseline	Lentivirus	0.777
EpitopeTransfer	Measles morbilivirus	0.761
Baseline	Measles morbilivirus	0.584
EpitopeTransfer	Mononegavirales	0.916
Baseline	Mononegavirales	0.559
EpitopeTransfer	Orthopoxvirus	0.975
Baseline	Orthopoxvirus	0.850
EpitopeTransfer	Ovolvulus	0.599
Baseline	Ovolvulus	0.678
EpitopeTransfer	P. aeruginosa	0.988
Baseline	P. aeruginosa	0.914
EpitopeTransfer	P. falciparum	0.604

Method	Dataset	Value
Baseline	P. falciparum	0.412
EpitopeTransfer	S. mansoni	0.110
Baseline	S. mansoni	0.593
EpitopeTransfer	Sars-cov-2	0.481
Baseline	Sars-cov-2	0.345
EpitopeTransfer	T. gondii	0.931
Baseline	T. gondii	0.668

Table 46: Comparison of methods for Specificity

Method	Dataset	Value
EpitopeTransfer	C. difficile	0.897
Baseline	C. difficile	0.985
EpitopeTransfer	C. trachomatis	0.783
Baseline	C. trachomatis	0.631
EpitopeTransfer	Corynebacterium	0.067
Baseline	Corynebacterium	1.000
EpitopeTransfer	Enterobacteriaceae	0.397
Baseline	Enterobacteriaceae	0.740
EpitopeTransfer	Firoviridae	0.844
Baseline	Firoviridae	0.857
EpitopeTransfer	Human gammaherpesvirus 4	0.907
Baseline	Human gammaherpesvirus 4	0.915
EpitopeTransfer	Influenza A	0.984
Baseline	Influenza A	0.952
EpitopeTransfer	Lentivirus	0.688
Baseline	Lentivirus	0.442
EpitopeTransfer	Measles morbilivirus	0.333
Baseline	Measles morbilivirus	0.312
EpitopeTransfer	Mononegavirales	0.586
Baseline	Mononegavirales	0.778
EpitopeTransfer	Orthopoxvirus	0.339
Baseline	Orthopoxvirus	0.194

Method	Dataset	Value
EpitopeTransfer	Ovolvulus	0.630
Baseline	Ovolvulus	0.380
EpitopeTransfer	P. aeruginosa	0.000
Baseline	P. aeruginosa	0.000
EpitopeTransfer	P. falciparum	0.829
Baseline	P. falciparum	0.909
EpitopeTransfer	S. mansoni	0.888
Baseline	S. mansoni	0.580
EpitopeTransfer	Sars-cov-2	0.734
Baseline	Sars-cov-2	0.764
EpitopeTransfer	T. gondii	0.220
Baseline	T. gondii	0.681

Metric	EpitopeTransfer	Baseline
AUC	0.698 (±0.027)	0.625 (±0.033)
F1	0.549 (±0.053)	0.454 (±0.056)
MCC	0.249 (±0.044)	0.154 (±0.039)
Balanced Accuracy	0.630 (±0.027)	0.581 (±0.020)
PPV	0.541 (±0.060)	0.545 (±0.072)
NPV	0.707 (±0.065)	0.620 (±0.058)
Sensitivity	0.664 (±0.072)	0.508 (±0.063)
Specificity	0.596 (±0.076)	0.654 (±0.072)

Table 47: Summary of average test set performance (*mean ±standard error*) for EpitopeTransfer (proposed method) and the baseline method across 17 selected datasets. Each row corresponds to a performance evaluation metric, and the values indicate the mean performance of each method over all datasets.

Appendix H

Statistical comparisons of median values for each performance metric were performed to assess the significance of differences between EpitopeTransfer, its generalized variant, and the baseline method. The Wilcoxon signed rank test was used as the primary statistical method to evaluate whether observed differences in medians were statistically meaningful. The analysis includes the following columns: “Pair”, which specifies the pairwise comparison (e.g., EpitopeTransfer vs. Baseline); “Medians of Diff.,” representing the median of paired differences (95% CI); “p-value”, which indicates the unadjusted significance level from the Wilcoxon test; and “Significant”, which highlights whether the corrected p-value falls below the significance threshold of 0.05.

Comparison Results for AUC

Metric	Pair	Medians of Diff.	p-value	FDR	Signif.
AUC	EpitopeTransfer vs Baseline	0.075 (0.017, 0.132)	0.00934	0.00934	Yes

Table 48: Comparison Results for AUC

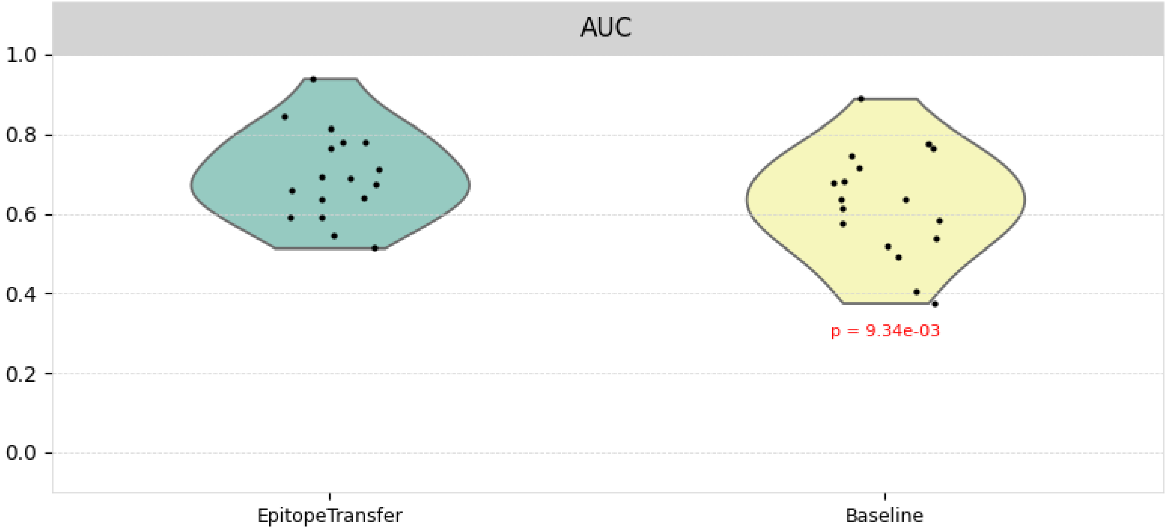


Figure 18: Performance plot for the AUC metric

Comparison Results for Balanced Accuracy

Metric	Pair	Medians of Diff.	p-value	FDR	Signif.
B. ACC	EpitopeTransfer vs Baseline	0.049 (0.024, 0.084)	0.01500	0.01500	Yes

Table 49: Comparison Results for Balanced Accuracy

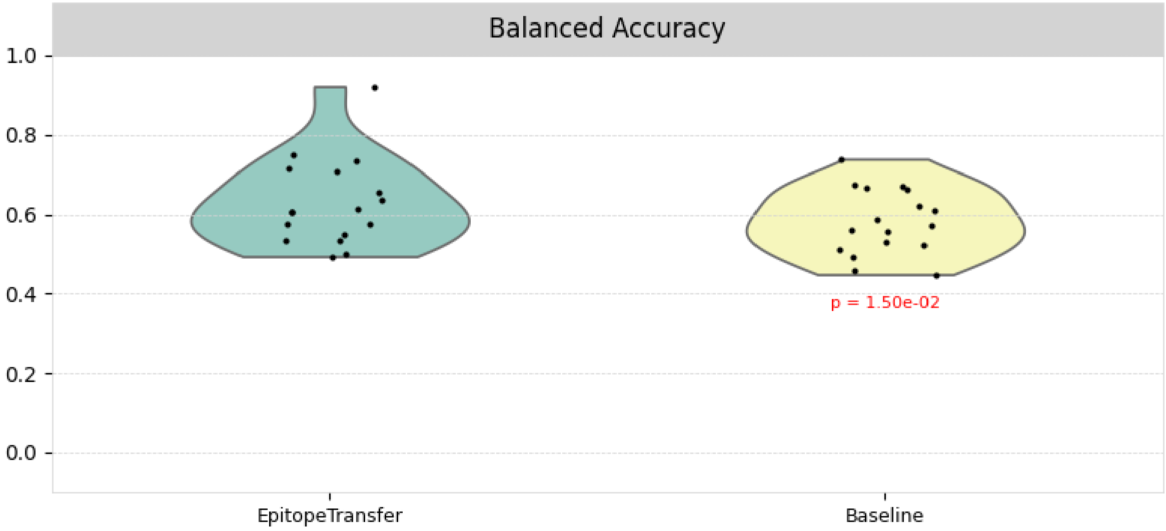


Figure 19: Performance plot for the Balanced Accuracy metric

Comparison Results for F1

Metric	Pair	Medians of Diff.	p-value	FDR	Signif.
F1	EpitopeTransfer vs Baseline	0.091 (0.044, 0.131)	0.00934	0.00934	Yes

Table 50: Comparison Results for F1

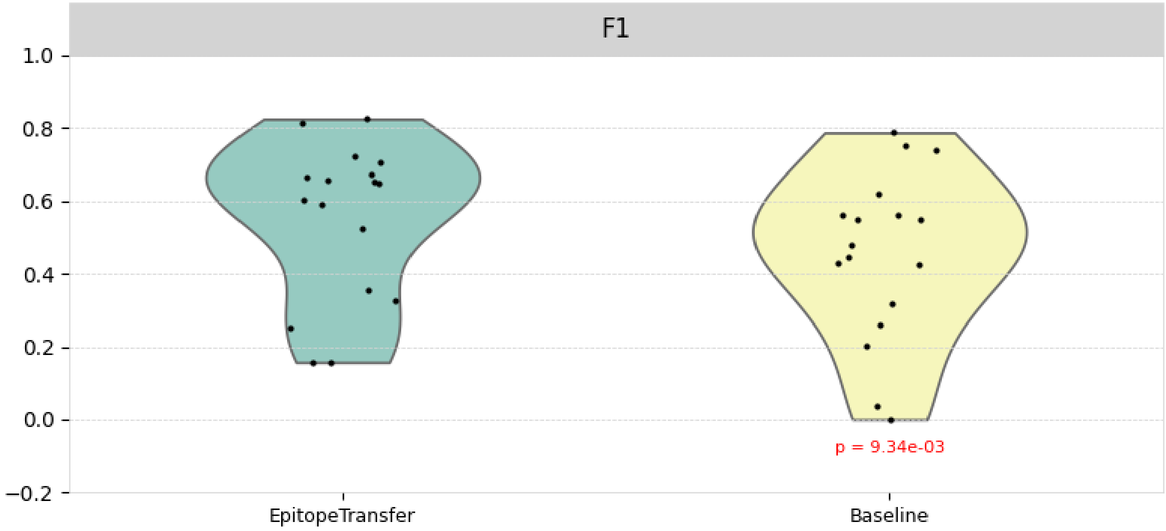


Figure 20: Performance plot for the F1 metric

Comparison Results for MCC

Metric	Pair	Medians of Diff.	p-value	FDR	Signif.
MCC	EpitopeTransfer vs Baseline	0.091 (0.048, 0.154)	0.00934	0.00934	Yes

Table 51: Comparison Results for MCC

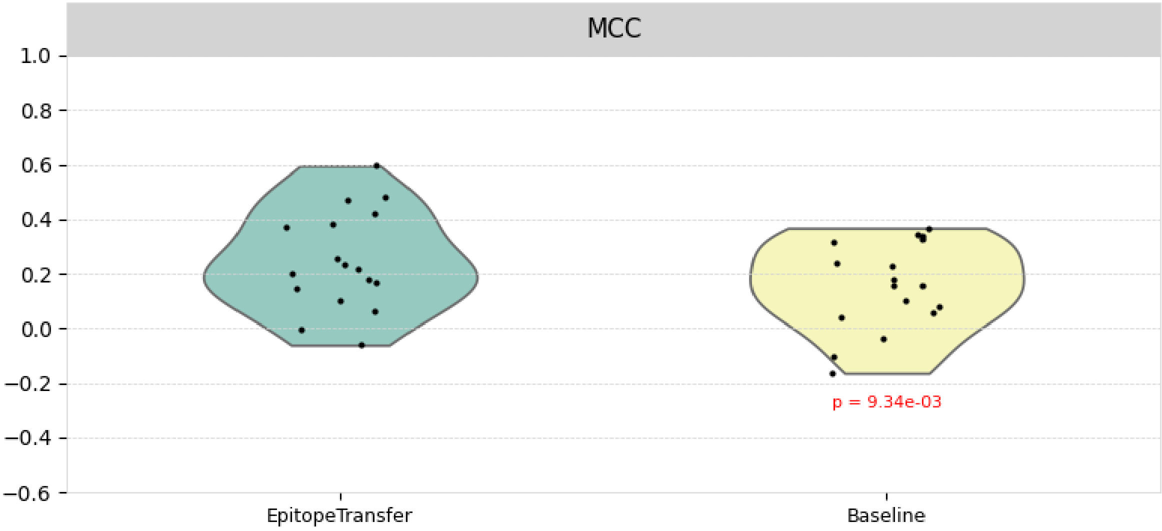


Figure 21: Performance plot for the MCC metric

Comparison Results for NPV

Metric	Pair	Medians of Diff.	p-value	FDR	Signif.
NPV	EpitopeTransfer vs Baseline	0.075 (0.020, 0.148)	0.00567	0.00567	Yes

Table 52: Comparison Results for NPV

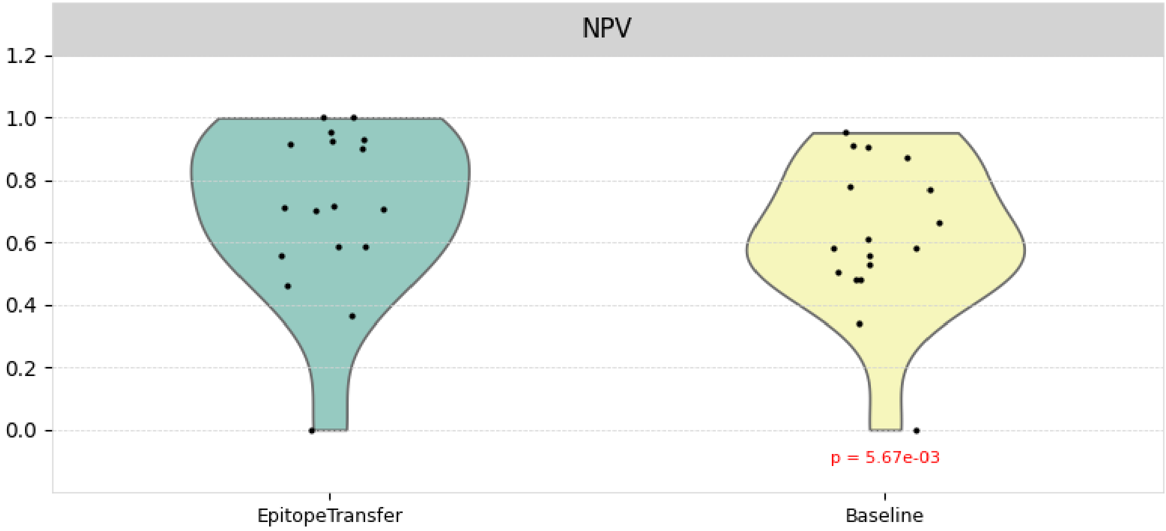


Figure 22: Performance plot for the NPV metric

Comparison Results for PPV

Metric	Pair	Medians of Diff.	p-value	FDR	Signif.
PPV	EpitopeTransfer vs Baseline	0.020 (-0.032, 0.059)	0.45857	0.45857	No

Table 53: Comparison Results for PPV

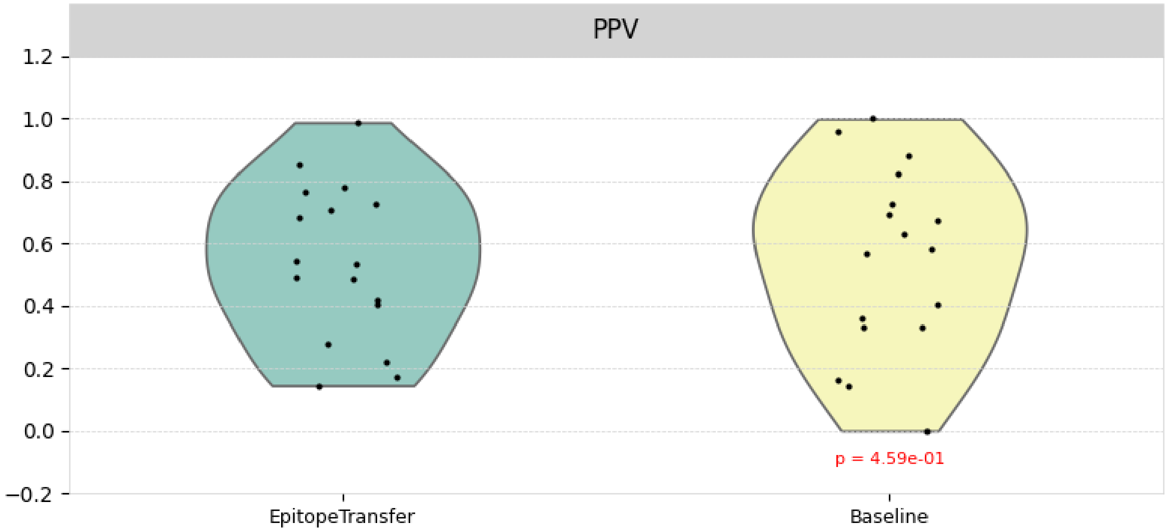


Figure 23: Performance plot for the PPV metric

Comparison Results for Sensitivity

Metric	Pair	Medians of Diff.	p-value	FDR	Signif.
Sensit.	EpitopeTransfer vs Baseline	0.147 (0.032, 0.267)	0.03479	0.03479	Yes

Table 54: Comparison Results for Sensitivity

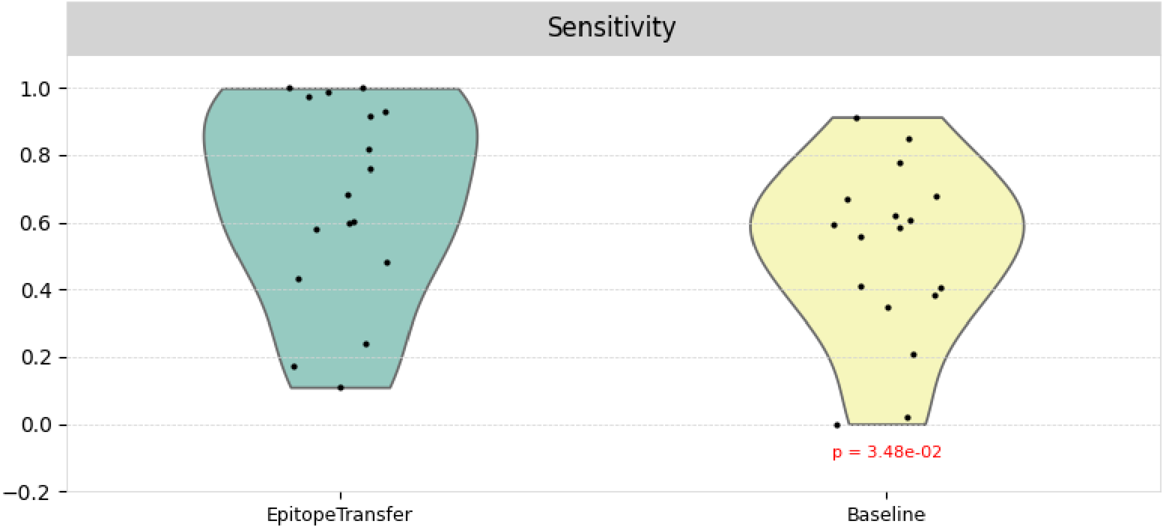


Figure 24: Performance plot for the Sensitivity metric

Comparison Results for Specificity

Metric	Pair	Medians of Diff.	p-value	FDR	Signif.
Specif.	EpitopeTransfer vs Baseline	-0.021 (-0.212, 0.110)	0.73679	0.73679	No

Table 55: Comparison Results for Specificity

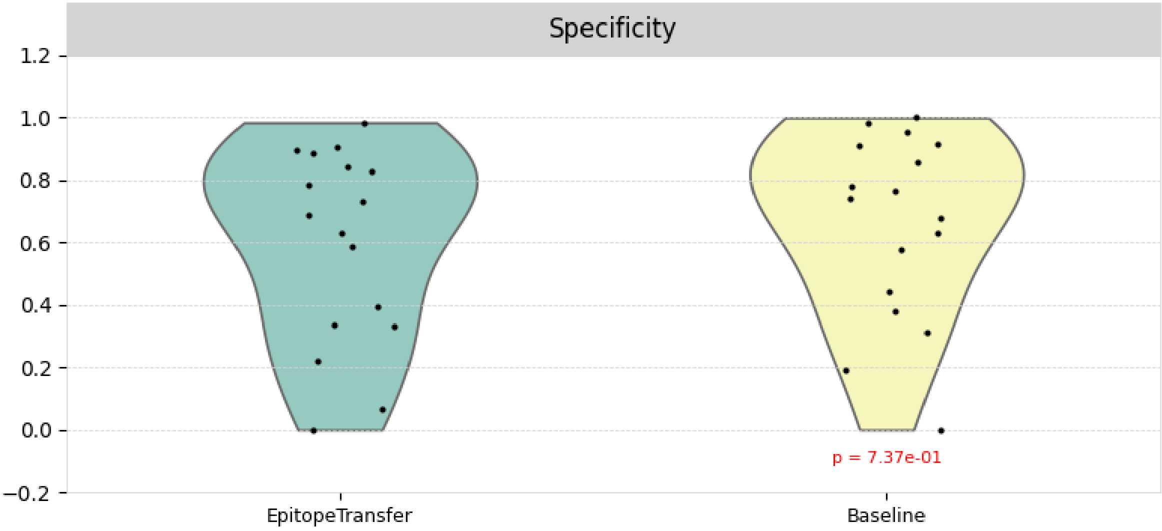


Figure 25: Performance plot for the Specificity metric

Summary of Comparison Results

Metric	Pair	Medians of Diff.	p-value	FDR	Signif.
AUC	EpitopeTransfer vs Baseline	0.075 (0.017, 0.132)	0.00934	0.00934	Yes
B. ACC	EpitopeTransfer vs Baseline	0.049 (0.024, 0.084)	0.01500	0.01500	Yes
F1	EpitopeTransfer vs Baseline	0.091 (0.044, 0.131)	0.00934	0.00934	Yes
MCC	EpitopeTransfer vs Baseline	0.091 (0.048, 0.154)	0.00934	0.00934	Yes
NPV	EpitopeTransfer vs Baseline	0.075 (0.020, 0.148)	0.00567	0.00567	Yes
PPV	EpitopeTransfer vs Baseline	0.020 (-0.032, 0.059)	0.45857	0.45857	No
Sensit.	EpitopeTransfer vs Baseline	0.147 (0.032, 0.267)	0.03479	0.03479	Yes
Specif.	EpitopeTransfer vs Baseline	-0.021 (-0.212, 0.110)	0.73679	0.73679	No

Table 56: Summary of Comparison Results across All Metrics

Appendix I

The estimated performance of **EpitopeTransfer (ESM-1b)** and **EpitopeTransfer (ESM-2)** is presented below for each dataset. The *Method* column refers to the employed approach. The *Dataset* column corresponds to data from 20 specific taxa, and *Value* represents the value for each metric presented. The evaluated metrics include **AUC** (Area Under the Curve), **F1** score, **MCC** (Matthews Correlation Coefficient), **Accuracy**, **PPV** (Positive Predictive Value), **NPV** (Negative Predictive Value), **Sensitivity**, and **Specificity**.

Table 57: Comparison of methods for AUC

Method	Dataset	Value
EpitopeTransferESM2	B. pertussis	0.533
EpitopeTransferESM1b	B. pertussis	0.555
EpitopeTransferESM2	C. difficile	0.656
EpitopeTransferESM1b	C. difficile	0.707
EpitopeTransferESM2	C. trachomatis	0.834
EpitopeTransferESM1b	C. trachomatis	0.773
EpitopeTransferESM2	Corynebacterium	0.632
EpitopeTransferESM1b	Corynebacterium	0.590
EpitopeTransferESM2	E. coli	0.909
EpitopeTransferESM1b	E. coli	0.853
EpitopeTransferESM2	Enterobacteriaceae	0.821
EpitopeTransferESM1b	Enterobacteriaceae	0.826
EpitopeTransferESM2	Filoviridae	0.959
EpitopeTransferESM1b	Filoviridae	0.972
EpitopeTransferESM2	Lentivirus	0.666
EpitopeTransferESM1b	Lentivirus	0.789
EpitopeTransferESM2	M. tuberculosis	0.479
EpitopeTransferESM1b	M. tuberculosis	0.478
EpitopeTransferESM2	Mononegavirales	0.787
EpitopeTransferESM1b	Mononegavirales	0.725
EpitopeTransferESM2	Orthopoxvirus	0.649
EpitopeTransferESM1b	Orthopoxvirus	0.689
EpitopeTransferESM2	Ovolvulus	0.606

Method	**Dataset**	**Value**
EpitopeTransferESM1b	Ovolvulus	0.626
EpitopeTransferESM2	P. aeruginosa	0.720
EpitopeTransferESM1b	P. aeruginosa	0.721
EpitopeTransferESM2	P. falciparum	0.794
EpitopeTransferESM1b	P. falciparum	0.810
EpitopeTransferESM2	S. mansoni	0.539
EpitopeTransferESM1b	S. mansoni	0.557
EpitopeTransferESM2	T. gondii	0.651
EpitopeTransferESM1b	T. gondii	0.705
EpitopeTransferESM2	Human Gammaherpesvirus 4	0.612
EpitopeTransferESM1b	Human Gammaherpesvirus 4	0.593
EpitopeTransferESM2	Influenza A	0.654
EpitopeTransferESM1b	Influenza A	0.756
EpitopeTransferESM2	Measles Morbilivirus	0.595
EpitopeTransferESM1b	Measles Morbilivirus	0.522
EpitopeTransferESM2	Sars-Cov-2	0.625
EpitopeTransferESM1b	Sars-Cov-2	0.547

Table 58: Comparison of methods for F1

Method	**Dataset**	**Value**
EpitopeTransferESM2	B. pertussis	0.619
EpitopeTransferESM1b	B. pertussis	0.836
EpitopeTransferESM2	C. difficile	0.236
EpitopeTransferESM1b	C. difficile	0.236
EpitopeTransferESM2	C. trachomatis	0.774
EpitopeTransferESM1b	C. trachomatis	0.717
EpitopeTransferESM2	Corynebacterium	0.672
EpitopeTransferESM1b	Corynebacterium	0.557
EpitopeTransferESM2	E. coli	0.886
EpitopeTransferESM1b	E. coli	0.872
EpitopeTransferESM2	Enterobacteriaceae	0.709
EpitopeTransferESM1b	Enterobacteriaceae	0.738

Method	Dataset	Value
EpitopeTransferESM2	Filoviridae	0.651
EpitopeTransferESM1b	Filoviridae	0.780
EpitopeTransferESM2	Lentivirus	0.821
EpitopeTransferESM1b	Lentivirus	0.925
EpitopeTransferESM2	M. tuberculosis	0.257
EpitopeTransferESM1b	M. tuberculosis	0.586
EpitopeTransferESM2	Mononegavirales	0.638
EpitopeTransferESM1b	Mononegavirales	0.565
EpitopeTransferESM2	Orthopoxvirus	0.352
EpitopeTransferESM1b	Orthopoxvirus	0.384
EpitopeTransferESM2	Ovolvulus	0.142
EpitopeTransferESM1b	Ovolvulus	0.362
EpitopeTransferESM2	P. aeruginosa	0.742
EpitopeTransferESM1b	P. aeruginosa	0.835
EpitopeTransferESM2	P. falciparum	0.805
EpitopeTransferESM1b	P. falciparum	0.826
EpitopeTransferESM2	S. mansoni	0.368
EpitopeTransferESM1b	S. mansoni	0.437
EpitopeTransferESM2	T. gondii	0.832
EpitopeTransferESM1b	T. gondii	0.811
EpitopeTransferESM2	Human Gammaherpesvirus 4	0.341
EpitopeTransferESM1b	Human Gammaherpesvirus 4	0.444
EpitopeTransferESM2	Influenza A	0.782
EpitopeTransferESM1b	Influenza A	0.818
EpitopeTransferESM2	Measles Morbilivirus	0.638
EpitopeTransferESM1b	Measles Morbilivirus	0.592
EpitopeTransferESM2	Sars-Cov-2	0.000
EpitopeTransferESM1b	Sars-Cov-2	0.120

Table 59: Comparison of methods for MCC

Method	Dataset	Value
EpitopeTransferESM2	B. pertussis	0.085

Method	Dataset	Value
EpitopeTransferESM1b	B. pertussis	0.000
EpitopeTransferESM2	C. difficile	0.137
EpitopeTransferESM1b	C. difficile	0.173
EpitopeTransferESM2	C. trachomatis	0.568
EpitopeTransferESM1b	C. trachomatis	0.447
EpitopeTransferESM2	Corynebacterium	0.282
EpitopeTransferESM1b	Corynebacterium	0.064
EpitopeTransferESM2	E. coli	0.442
EpitopeTransferESM1b	E. coli	0.325
EpitopeTransferESM2	Enterobacteriaceae	0.427
EpitopeTransferESM1b	Enterobacteriaceae	0.479
EpitopeTransferESM2	Filoviridae	0.610
EpitopeTransferESM1b	Filoviridae	0.766
EpitopeTransferESM2	Lentivirus	0.350
EpitopeTransferESM1b	Lentivirus	0.770
EpitopeTransferESM2	M. tuberculosis	-0.016
EpitopeTransferESM1b	M. tuberculosis	-0.031
EpitopeTransferESM2	Mononegavirales	0.428
EpitopeTransferESM1b	Mononegavirales	0.286
EpitopeTransferESM2	Orthopoxvirus	0.168
EpitopeTransferESM1b	Orthopoxvirus	0.226
EpitopeTransferESM2	Ovolvulus	0.095
EpitopeTransferESM1b	Ovolvulus	0.272
EpitopeTransferESM2	P. aeruginosa	0.249
EpitopeTransferESM1b	P. aeruginosa	0.147
EpitopeTransferESM2	P. falciparum	0.410
EpitopeTransferESM1b	P. falciparum	0.505
EpitopeTransferESM2	S. mansoni	0.069
EpitopeTransferESM1b	S. mansoni	0.056
EpitopeTransferESM2	T. gondii	0.312
EpitopeTransferESM1b	T. gondii	0.218
EpitopeTransferESM2	Human Gammaherpesvirus 4	0.275
EpitopeTransferESM1b	Human Gammaherpesvirus 4	0.241

Method	Dataset	Value
EpitopeTransferESM2	Influenza A	0.218
EpitopeTransferESM1b	Influenza A	0.176
EpitopeTransferESM2	Measles Morbilivirus	0.068
EpitopeTransferESM1b	Measles Morbilivirus	-0.136
EpitopeTransferESM2	Sars-Cov-2	0.000
EpitopeTransferESM1b	Sars-Cov-2	0.043

Table 60: Comparison of methods for Balanced Accuracy

Method	Dataset	Value
EpitopeTransferESM2	B. pertussis	0.547
EpitopeTransferESM1b	B. pertussis	0.500
EpitopeTransferESM2	C. difficile	0.605
EpitopeTransferESM1b	C. difficile	0.647
EpitopeTransferESM2	C. trachomatis	0.783
EpitopeTransferESM1b	C. trachomatis	0.723
EpitopeTransferESM2	Corynebacterium	0.632
EpitopeTransferESM1b	Corynebacterium	0.531
EpitopeTransferESM2	E. coli	0.623
EpitopeTransferESM1b	E. coli	0.574
EpitopeTransferESM2	Enterobacteriaceae	0.713
EpitopeTransferESM1b	Enterobacteriaceae	0.739
EpitopeTransferESM2	Filoviridae	0.827
EpitopeTransferESM1b	Filoviridae	0.947
EpitopeTransferESM2	Lentivirus	0.629
EpitopeTransferESM1b	Lentivirus	0.844
EpitopeTransferESM2	M. tuberculosis	0.494
EpitopeTransferESM1b	M. tuberculosis	0.486
EpitopeTransferESM2	Mononegavirales	0.721
EpitopeTransferESM1b	Mononegavirales	0.651
EpitopeTransferESM2	Orthopoxvirus	0.607
EpitopeTransferESM1b	Orthopoxvirus	0.631
EpitopeTransferESM2	Ovolvulus	0.528

Method	Dataset	Value
EpitopeTransferESM1b	Ovolvulus	0.620
EpitopeTransferESM2	P. aeruginosa	0.634
EpitopeTransferESM1b	P. aeruginosa	0.515
EpitopeTransferESM2	P. falciparum	0.686
EpitopeTransferESM1b	P. falciparum	0.743
EpitopeTransferESM2	S. mansoni	0.537
EpitopeTransferESM1b	S. mansoni	0.525
EpitopeTransferESM2	T. gondii	0.613
EpitopeTransferESM1b	T. gondii	0.582
EpitopeTransferESM2	Human Gammaherpesvirus 4	0.589
EpitopeTransferESM1b	Human Gammaherpesvirus 4	0.600
EpitopeTransferESM2	Influenza A	0.616
EpitopeTransferESM1b	Influenza A	0.579
EpitopeTransferESM2	Measles Morbilivirus	0.516
EpitopeTransferESM1b	Measles Morbilivirus	0.461
EpitopeTransferESM2	Sars-Cov-2	0.500
EpitopeTransferESM1b	Sars-Cov-2	0.518

Table 61: Comparison of methods for PPV

Method	Dataset	Value
EpitopeTransferESM2	B. pertussis	0.757
EpitopeTransferESM1b	B. pertussis	0.718
EpitopeTransferESM2	C. difficile	0.158
EpitopeTransferESM1b	C. difficile	0.138
EpitopeTransferESM2	C. trachomatis	0.817
EpitopeTransferESM1b	C. trachomatis	0.744
EpitopeTransferESM2	Corynebacterium	0.571
EpitopeTransferESM1b	Corynebacterium	0.500
EpitopeTransferESM2	E. coli	0.796
EpitopeTransferESM1b	E. coli	0.776
EpitopeTransferESM2	Enterobacteriaceae	0.674
EpitopeTransferESM1b	Enterobacteriaceae	0.691

Method	Dataset	Value
EpitopeTransferESM2	Filoviridae	0.603
EpitopeTransferESM1b	Filoviridae	0.663
EpitopeTransferESM2	Lentivirus	0.725
EpitopeTransferESM1b	Lentivirus	0.860
EpitopeTransferESM2	M. tuberculosis	0.502
EpitopeTransferESM1b	M. tuberculosis	0.509
EpitopeTransferESM2	Mononegavirales	0.586
EpitopeTransferESM1b	Mononegavirales	0.481
EpitopeTransferESM2	Orthopoxvirus	0.258
EpitopeTransferESM1b	Orthopoxvirus	0.318
EpitopeTransferESM2	Ovolvulus	0.306
EpitopeTransferESM1b	Ovolvulus	0.423
EpitopeTransferESM2	P. aeruginosa	0.800
EpitopeTransferESM1b	P. aeruginosa	0.717
EpitopeTransferESM2	P. falciparum	0.743
EpitopeTransferESM1b	P. falciparum	0.793
EpitopeTransferESM2	S. mansoni	0.324
EpitopeTransferESM1b	S. mansoni	0.297
EpitopeTransferESM2	T. gondii	0.745
EpitopeTransferESM1b	T. gondii	0.730
EpitopeTransferESM2	Human Gammaherpesvirus 4	0.829
EpitopeTransferESM1b	Human Gammaherpesvirus 4	0.680
EpitopeTransferESM2	Influenza A	0.812
EpitopeTransferESM1b	Influenza A	0.787
EpitopeTransferESM2	Measles Morbilivirus	0.478
EpitopeTransferESM1b	Measles Morbilivirus	0.448
EpitopeTransferESM2	Sars-Cov-2	0.000
EpitopeTransferESM1b	Sars-Cov-2	0.151

Table 62: Comparison of methods for NPV

Method	Dataset	Value
EpitopeTransferESM2	B. pertussis	0.320

Method	Dataset	Value
EpitopeTransferESM1b	B. pertussis	0.000
EpitopeTransferESM2	C. difficile	0.931
EpitopeTransferESM1b	C. difficile	0.964
EpitopeTransferESM2	C. trachomatis	0.753
EpitopeTransferESM1b	C. trachomatis	0.704
EpitopeTransferESM2	Corynebacterium	0.730
EpitopeTransferESM1b	Corynebacterium	0.565
EpitopeTransferESM2	E. coli	1.000
EpitopeTransferESM1b	E. coli	0.941
EpitopeTransferESM2	Enterobacteriaceae	0.752
EpitopeTransferESM1b	Enterobacteriaceae	0.789
EpitopeTransferESM2	Filoviridae	0.966
EpitopeTransferESM1b	Filoviridae	0.994
EpitopeTransferESM2	Lentivirus	0.750
EpitopeTransferESM1b	Lentivirus	1.000
EpitopeTransferESM2	M. tuberculosis	0.478
EpitopeTransferESM1b	M. tuberculosis	0.458
EpitopeTransferESM2	Mononegavirales	0.826
EpitopeTransferESM1b	Mononegavirales	0.790
EpitopeTransferESM2	Orthopoxvirus	0.874
EpitopeTransferESM1b	Orthopoxvirus	0.876
EpitopeTransferESM2	Ovolvulus	0.857
EpitopeTransferESM1b	Ovolvulus	0.885
EpitopeTransferESM2	P. aeruginosa	0.432
EpitopeTransferESM1b	P. aeruginosa	1.000
EpitopeTransferESM2	P. falciparum	0.708
EpitopeTransferESM1b	P. falciparum	0.732
EpitopeTransferESM2	S. mansoni	0.740
EpitopeTransferESM1b	S. mansoni	0.765
EpitopeTransferESM2	T. gondii	0.684
EpitopeTransferESM1b	T. gondii	0.561
EpitopeTransferESM2	Human Gammaherpesvirus 4	0.595
EpitopeTransferESM1b	Human Gammaherpesvirus 4	0.609

Method	Dataset	Value
EpitopeTransferESM2	Influenza A	0.393
EpitopeTransferESM1b	Influenza A	0.408
EpitopeTransferESM2	Measles Morbilivirus	0.667
EpitopeTransferESM1b	Measles Morbilivirus	0.312
EpitopeTransferESM2	Sars-Cov-2	0.898
EpitopeTransferESM1b	Sars-Cov-2	0.901

Table 63: Comparison of methods for Sensitivity

Method	Dataset	Value
EpitopeTransferESM2	B. pertussis	0.523
EpitopeTransferESM1b	B. pertussis	1.000
EpitopeTransferESM2	C. difficile	0.463
EpitopeTransferESM1b	C. difficile	0.829
EpitopeTransferESM2	C. trachomatis	0.734
EpitopeTransferESM1b	C. trachomatis	0.691
EpitopeTransferESM2	Corynebacterium	0.815
EpitopeTransferESM1b	Corynebacterium	0.630
EpitopeTransferESM2	E. coli	1.000
EpitopeTransferESM1b	E. coli	0.997
EpitopeTransferESM2	Enterobacteriaceae	0.748
EpitopeTransferESM1b	Enterobacteriaceae	0.793
EpitopeTransferESM2	Filoviridae	0.707
EpitopeTransferESM1b	Filoviridae	0.948
EpitopeTransferESM2	Lentivirus	0.946
EpitopeTransferESM1b	Lentivirus	1.000
EpitopeTransferESM2	M. tuberculosis	0.173
EpitopeTransferESM1b	M. tuberculosis	0.691
EpitopeTransferESM2	Mononegavirales	0.699
EpitopeTransferESM1b	Mononegavirales	0.685
EpitopeTransferESM2	Orthopoxvirus	0.552
EpitopeTransferESM1b	Orthopoxvirus	0.483
EpitopeTransferESM2	Ovolvulus	0.092

Method	Dataset	Value
EpitopeTransferESM1b	Ovolvulus	0.317
EpitopeTransferESM2	P. aeruginosa	0.691
EpitopeTransferESM1b	P. aeruginosa	1.000
EpitopeTransferESM2	P. falciparum	0.877
EpitopeTransferESM1b	P. falciparum	0.863
EpitopeTransferESM2	S. mansoni	0.426
EpitopeTransferESM1b	S. mansoni	0.826
EpitopeTransferESM2	T. gondii	0.941
EpitopeTransferESM1b	T. gondii	0.911
EpitopeTransferESM2	Human Gammaherpesvirus 4	0.215
EpitopeTransferESM1b	Human Gammaherpesvirus 4	0.330
EpitopeTransferESM2	Influenza A	0.755
EpitopeTransferESM1b	Influenza A	0.852
EpitopeTransferESM2	Measles Morbilivirus	0.959
EpitopeTransferESM1b	Measles Morbilivirus	0.871
EpitopeTransferESM2	Sars-Cov-2	0.000
EpitopeTransferESM1b	Sars-Cov-2	0.099

Table 64: Comparison of methods for Specificity

Method	Dataset	Value
EpitopeTransferESM2	B. pertussis	0.571
EpitopeTransferESM1b	B. pertussis	0.000
EpitopeTransferESM2	C. difficile	0.746
EpitopeTransferESM1b	C. difficile	0.465
EpitopeTransferESM2	C. trachomatis	0.831
EpitopeTransferESM1b	C. trachomatis	0.755
EpitopeTransferESM2	Corynebacterium	0.450
EpitopeTransferESM1b	Corynebacterium	0.433
EpitopeTransferESM2	E. coli	0.245
EpitopeTransferESM1b	E. coli	0.151
EpitopeTransferESM2	Enterobacteriaceae	0.679
EpitopeTransferESM1b	Enterobacteriaceae	0.686

Method	**Dataset**	**Value**
EpitopeTransferESM2	Filoviridae	0.947
EpitopeTransferESM1b	Filoviridae	0.945
EpitopeTransferESM2	Lentivirus	0.312
EpitopeTransferESM1b	Lentivirus	0.688
EpitopeTransferESM2	M. tuberculosis	0.815
EpitopeTransferESM1b	M. tuberculosis	0.281
EpitopeTransferESM2	Mononegavirales	0.744
EpitopeTransferESM1b	Mononegavirales	0.617
EpitopeTransferESM2	Orthopoxvirus	0.662
EpitopeTransferESM1b	Orthopoxvirus	0.779
EpitopeTransferESM2	Ovolvulus	0.963
EpitopeTransferESM1b	Ovolvulus	0.924
EpitopeTransferESM2	P. aeruginosa	0.576
EpitopeTransferESM1b	P. aeruginosa	0.030
EpitopeTransferESM2	P. falciparum	0.495
EpitopeTransferESM1b	P. falciparum	0.624
EpitopeTransferESM2	S. mansoni	0.647
EpitopeTransferESM1b	S. mansoni	0.224
EpitopeTransferESM2	T. gondii	0.286
EpitopeTransferESM1b	T. gondii	0.253
EpitopeTransferESM2	Human Gammaherpesvirus 4	0.963
EpitopeTransferESM1b	Human Gammaherpesvirus 4	0.871
EpitopeTransferESM2	Influenza A	0.476
EpitopeTransferESM1b	Influenza A	0.307
EpitopeTransferESM2	Measles Morbilivirus	0.073
EpitopeTransferESM1b	Measles Morbilivirus	0.052
EpitopeTransferESM2	Sars-Cov-2	1.000
EpitopeTransferESM1b	Sars-Cov-2	0.937

Table 65: Average performance of methods across all datasets (mean ±SEM)

Metric	**EpitopeTransferESM2**	**EpitopeTransferESM1b**
AUC	0.686 (±0.028)	0.690 (±0.029)

Metric	**EpitopeTransferESM2**	**EpitopeTransferESM1b**
F1	0.563 (±0.059)	0.622 (±0.052)
MCC	0.259 (±0.041)	0.251 (±0.055)
Balanced Accuracy	0.620 (±0.020)	0.621 (±0.028)
PPV	0.574 (±0.055)	0.571 (±0.049)
NPV	0.718 (±0.043)	0.713 (±0.060)
Sensitivity	0.616 (±0.068)	0.741 (±0.057)
Specificity	0.624 (±0.059)	0.501 (±0.072)

Appendix J

Statistical comparisons of median values for each performance metric were performed to assess the significance of differences between **EpitopeTransferESM2** and **EpitopeTransferESM1b** (ET denotes EpitopeTransfer in the tables presented below). The Wilcoxon signed rank test was used as the primary statistical method to evaluate whether observed differences in medians were statistically meaningful.

The analysis includes the following columns: "Pair", which specifies the pairwise comparison (e.g., EpitopeTransfer vs. Baseline); "Medians of Diff.," representing the median of paired differences (95% CI); "p-value", which indicates the unadjusted significance level from the Wilcoxon test; and "Significant", which highlights whether the corrected p-value falls below the significance threshold of 0.05.

Comparison Results for AUC

Metric	Pair	Medians of diff.	p-value	Signif.
AUC	ET_ESM2 vs ET_ESM1b	-0.003 (-0.029, 0.024)	0.86949	No

Table 66: Comparison Results for AUC

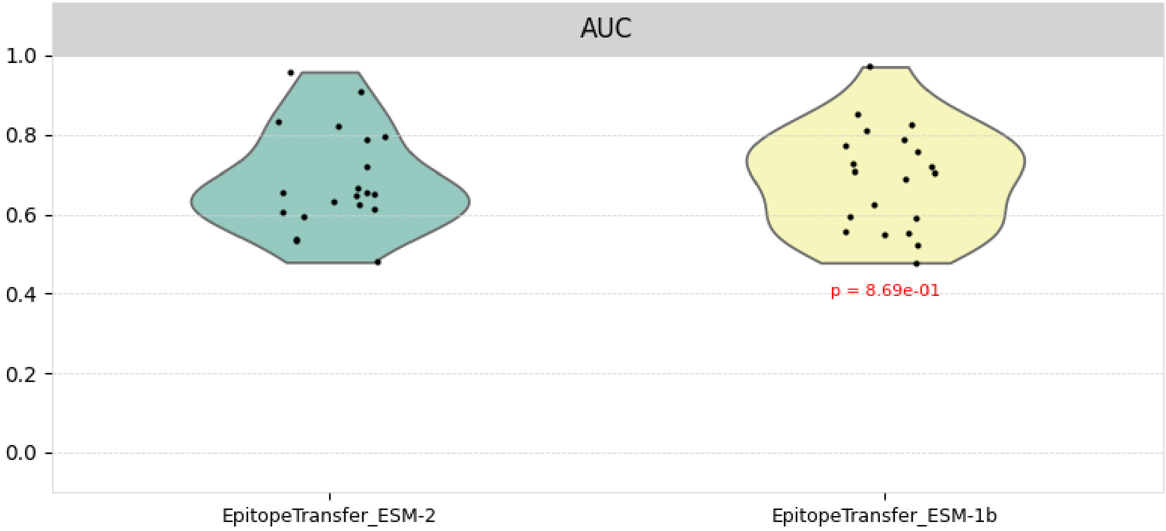


Figure 26: Performance plot for the AUC metric

Comparison Results for Balanced Accuracy

Metric	Pair	Medians of diff.	p-value	Signif.
BACC	ET_ESM2 vs ET_ESM1b	0.007 (-0.033, 0.039)	0.70118	No

Table 67: Comparison Results for Balanced Accuracy

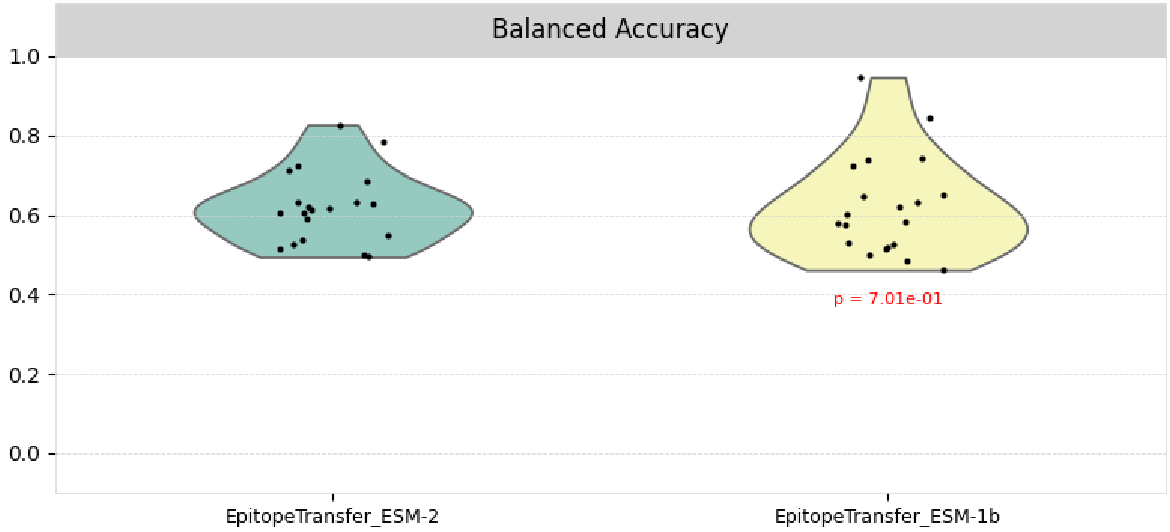


Figure 27: Performance plot for the Balanced Accuracy metric

Comparison Results for F1

Metric	Pair	Medians of diff.	p-value	Signif.
F1	ET_ESM2 vs ET_ESM1b	-0.051 (-0.107, -0.004)	0.03277	Yes

Table 68: Comparison Results for F1

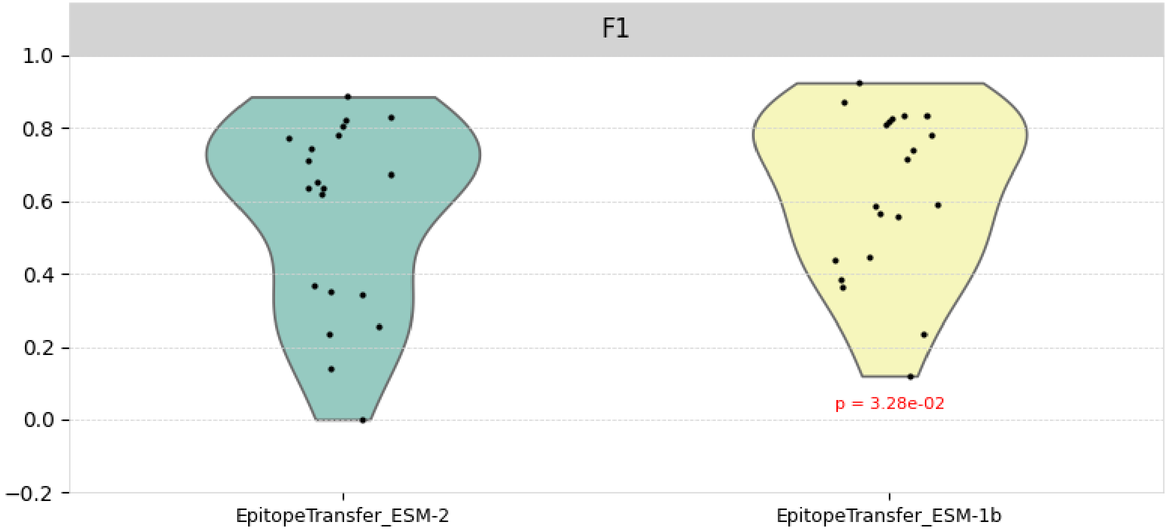


Figure 28: Performance plot for the F1 metric

Comparison Results for MCC

Metric	Pair	Medians of diff.	p-value	Signif.
MCC	ET_ESM2 vs ET_ESM1b	0.023 (-0.046, 0.080)	0.57060	No

Table 69: Comparison Results for MCC

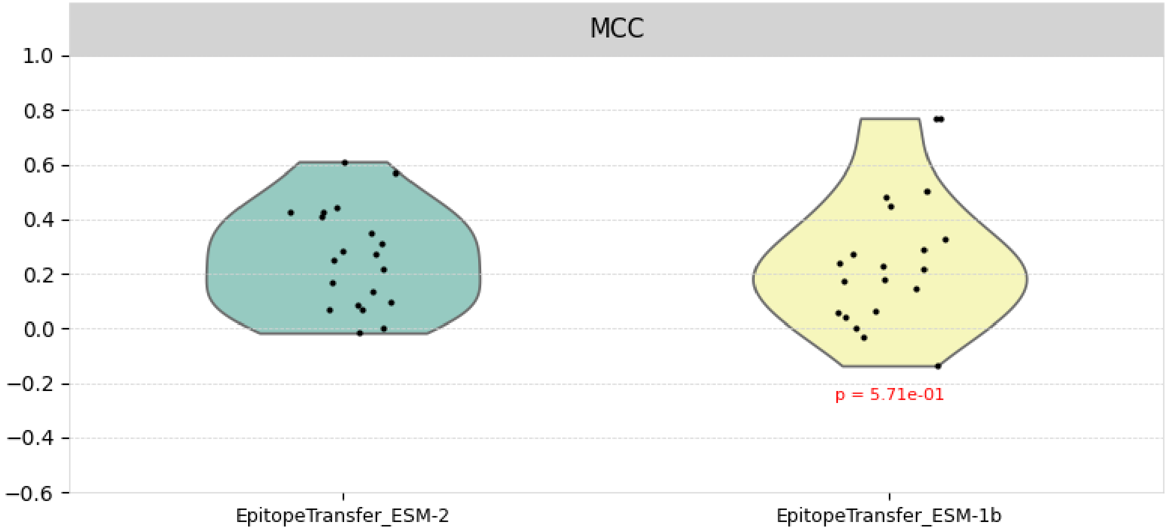


Figure 29: Performance plot for the MCC metric

Comparison Results for NPV

Metric	Pair	Medians of diff.	p-value	Signif.
NPV	ET_ESM2 vs ET_ESM1b	0.006 (-0.026, 0.069)	0.84082	No

Table 70: Comparison Results for NPV

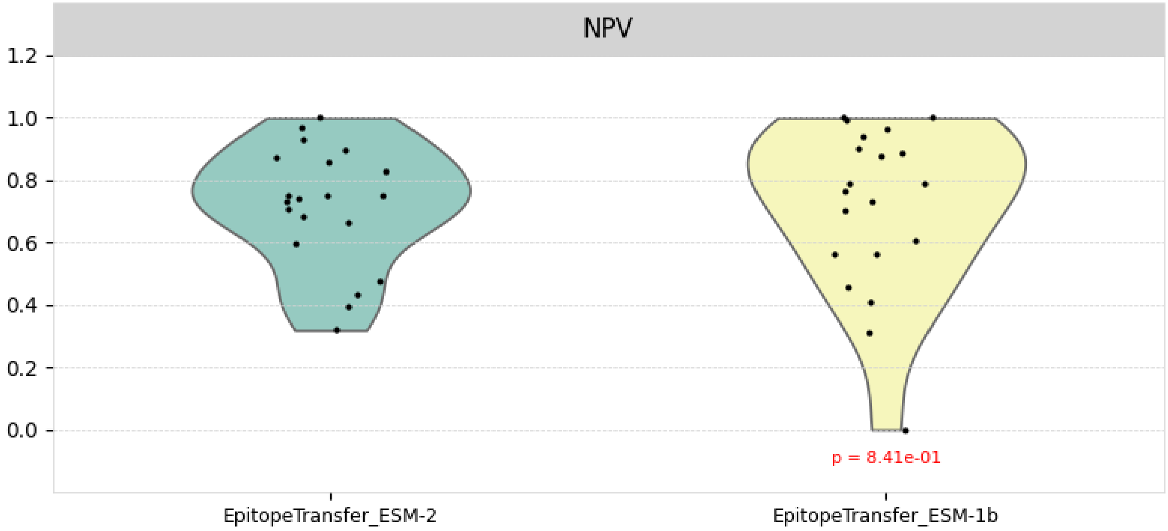


Figure 30: Performance plot for the NPV metric

Comparison Results for PPV

Metric	Pair	Medians of diff.	p-value	Signif.
PPV	ET_ESM2 vs ET_ESM1b	0.008 (-0.034, 0.044)	0.64766	No

Table 71: Comparison Results for PPV

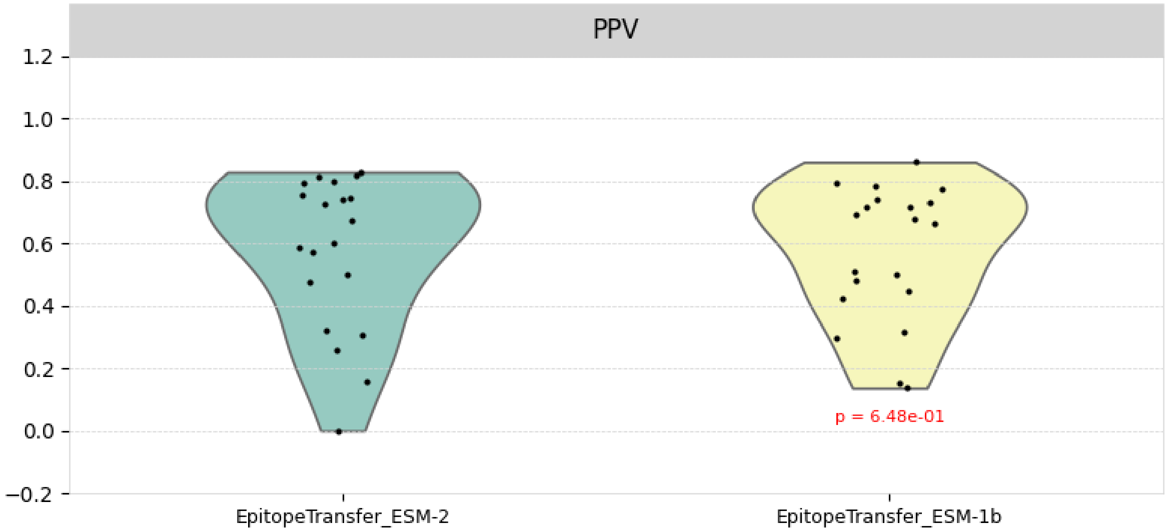


Figure 31: Performance plot for the PPV metric

Comparison Results for Sensitivity

Metric	Pair	Medians of diff.	p-value	Signif.
Sensit.	ET_ESM2 vs ET_ESM1b	-0.109 (-0.224, -0.014)	0.02395	Yes

Table 72: Comparison Results for Sensitivity

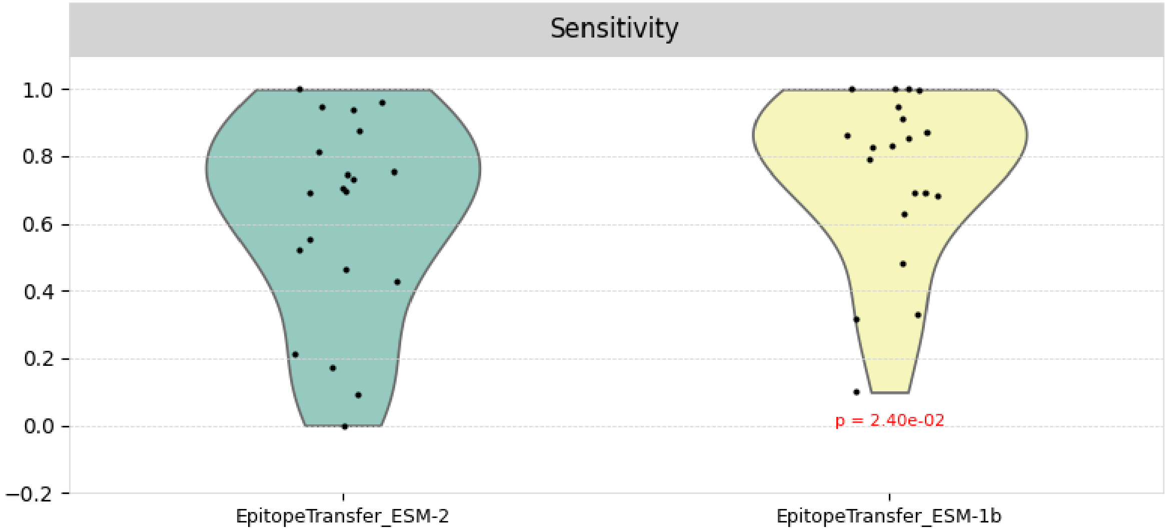


Figure 32: Performance plot for the Sensitivity metric

Comparison Results for Specificity

Metric	Pair	Medians of diff.	p-value	Signif.
Specif.	ET_ESM2 vs ET_ESM1b	0.084 (0.019, 0.264)	0.01718	Yes

Table 73: Comparison Results for Specificity

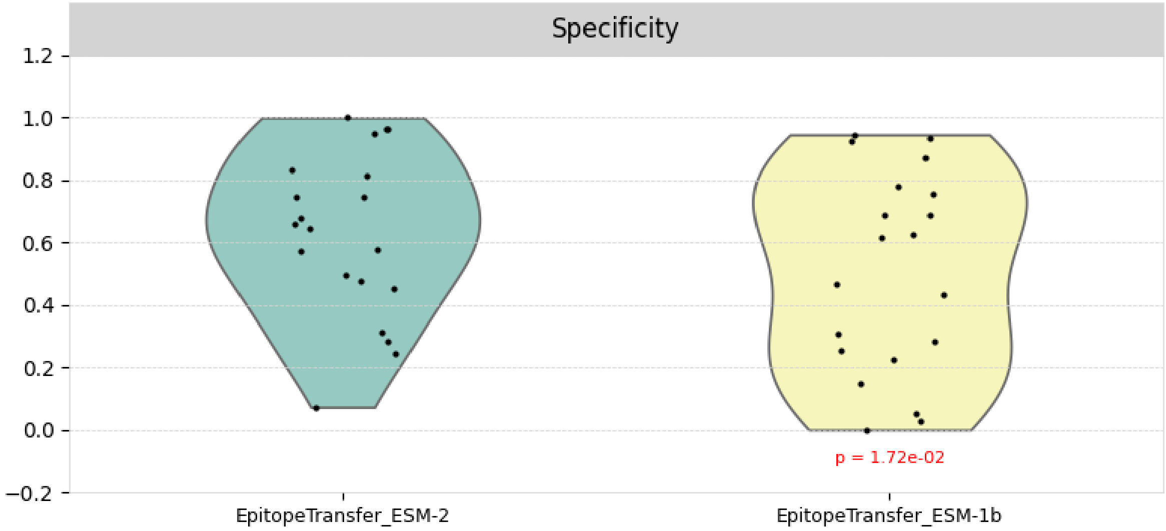


Figure 33: Performance plot for the Specificity metric

Summary of Comparison Results

Metric	Pair	Medians of diff.	p-value	Signif.
AUC	ET_ESM2 vs ET_ESM1b	-0.003 (-0.029, 0.024)	0.86949	No
BACC	ET_ESM2 vs ET_ESM1b	0.007 (-0.033, 0.039)	0.70118	No
F1	ET_ESM2 vs ET_ESM1b	-0.051 (-0.107, -0.004)	0.03277	Yes
MCC	ET_ESM2 vs ET_ESM1b	0.023 (-0.046, 0.080)	0.57060	No
NPV	ET_ESM2 vs ET_ESM1b	0.006 (-0.026, 0.069)	0.84082	No
PPV	ET_ESM2 vs ET_ESM1b	0.008 (-0.034, 0.044)	0.64766	No
Sensit.	ET_ESM2 vs ET_ESM1b	-0.109 (-0.224, -0.014)	0.02395	Yes
Specif.	ET_ESM2 vs ET_ESM1b	0.084 (0.019, 0.264)	0.01718	Yes

Table 74: Summary of comparison results across all metrics.

References

Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. Optuna: A next-generation hyperparameter optimization framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '19, page 26232631, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450362016. doi: 10.1145/3292500.3330701. URL `https://doi.org/10.1145/3292500.3330701`. 60, 86

D. Altschuh, A.M. Lesk, A.C. Bloomer, and A. Klug. Correlation of co-ordinated amino acid substitutions with function in viruses related to tobacco mosaic virus. *Journal of Molecular Biology*, 193(4):693–707, 1987. ISSN 0022-2836. doi: https://doi.org/10.1016/0022-2836(87)90352-4. URL `https://www.sciencedirect.com/science/article/pii/0022283687903524`. 38

Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In Doina Precup and Yee Whye Teh, editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 214–223. PMLR, 06–11 Aug 2017. URL `https://proceedings.mlr.press/v70/arjovsky17a.html`. 30

Jodie Ashford, João Reis-Cunha, Igor Lobo, Francisco Lobo, and Felipe Campelo. Organism-specific training improves performance of linear b-cell epitope prediction. *Bioinformatics*, 37(24):4826–4834, 07 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab536. URL `https://doi.org/10.1093/bioinformatics/btab536`. 35, 36, 42

Akash Bahai, Ehsaneddin Asgari, Mohammad R K Mofrad, Andreas Kloetgen, and Alice C McHardy. Epitopevec: linear epitope prediction using deep protein sequence embeddings. *Bioinformatics*, 37(23):4517–4525, 06 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btab467. URL `https://doi.org/10.1093/bioinformatics/btab467`. 36, 40, 64

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473, 2014. URL `https://api.semanticscholar.org/CorpusID:11212020`. 13

Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, 1970. 9

Jerome R Bellegarda. Statistical language model adaptation: review and perspectives. *Speech Communication*, 42(1):93–108, 2004. ISSN 0167-6393. doi: https://doi.org/10.1016/j.specom.2003.08.002. URL https://www.sciencedirect.com/science/article/pii/S0167639303001055. Adaptation Methods for Speech Recognition. 6

Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, May 2010. ISSN 1573-0565. doi: 10.1007/s10994-009-5152-4. URL https://doi.org/10.1007/s10994-009-5152-4. 1, 23, 26, 27

Emily M. Bender, Timnit Gebru, Angelina McMillan-Major, and Shmargaret Shmitchell. On the dangers of stochastic parrots: Can language models be too big? In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, FAccT '21, page 610623, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383097. doi: 10.1145/3442188.3445922. URL https://doi.org/10.1145/3442188.3445922. 20

Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994. doi: 10.1109/72.279181. 11, 12, 13, 14, 20

Yoshua Bengio, Réjean Ducharme, and Pascal Vincent. A neural probabilistic language model. In T. Leen, T. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems*, volume 13. MIT Press, 2000. URL https://proceedings.neurips.cc/paper_files/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf. 4, 5, 6, 10, 39

James Bergstra, Rémi Bardenet, Yoshua Bengio, and Balázs Kégl. Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper_files/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf. 60

Martin J. Blythe and Darren R. Flower. Benchmarking b cell epitope prediction: Underperformance of existing methods. *Protein Science*, 14, 2005. URL https://api.semanticscholar.org/CorpusID:45951800. 39

Nadav Brandes, Dan Ofer, Yam Peleg, Nadav Rappoport, and Michal Linial. Proteinbert: a universal deep-learning model of protein sequence and function. *Bioinformatics*, 38 (8):2102–2110, 02 2022. ISSN 1367-4803. doi: 10.1093/bioinformatics/btac020. URL https://doi.org/10.1093/bioinformatics/btac020. 19

T. Brants, Ashok Popat, Peng Xu, Franz Josef Och, and Jeffrey Dean. Large language models in machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2007. URL https://api.semanticscholar.org/CorpusID:633992. 6

Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001. ISSN 1573-0565. doi: 10.1023/A:1010933404324. URL https://doi.org/10.1023/A:1010933404324. 50

Benjamin Buchfink, Klaus Reuter, and Hajk-Georg Drost. Sensitive protein alignments at tree-of-life scale using diamond. *Nature Methods*, 18(4):366–368, 2021. ISSN 1548-7105. doi: 10.1038/s41592-021-01101-x. URL `https://doi.org/10.1038/s41592-021-01101-x`. 57

Leandro A. Bugnon, Emilio Fenoy, Alejandro A. Edera, Jonathan Raad, Georgina Stegmayer, and Diego H. Milone. Transfer learning: The key to functionally annotate the protein universe. *Patterns*, 4(2):100691, 2023. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2023.100691. URL `https://www.sciencedirect.com/science/article/pii/S2666389923000223`. 41

T. Caelli and B. McCane. Components analysis of hidden markov models in computer vision. In *12th International Conference on Image Analysis and Processing, 2003.Proceedings.*, pages 510–515, 2003. doi: 10.1109/ICIAP.2003.1234101. 7

F. Campelo and J. Ashford. epitopes: Processing, feature extraction and modelling of epitope data from the immune epitope database (iedb). "", 2022. URL `https://github.com/fcampelo/epitopes/tree/devel-next`. 57

Felipe Campelo, Ana Laura Grossi de Oliveira, João Reis-Cunha, Vanessa Gomes Fraga, Pedro Henrique Bastos, Jodie Ashford, Anikó Ekárt, Talita Emile Ribeiro Adelino, Marcos Vinicius Ferreira Silva, Felipe Campos de Melo Iani, Augusto César Parreiras de Jesus, Daniella Castanheira Bartholomeu, Giliane de Souza Trindade, Ricardo Toshio Fujiwara, Lilian Lacerda Bueno, and Francisco Pereira Lobo. Phylogeny-aware linear b-cell epitope predictor detects targets associated with immune response to orthopoxviruses. *Briefings in Bioinformatics*, 25(6):bbae527, 11 2024. ISSN 1477-4054. doi: 10.1093/bib/bbae527. URL `https://doi.org/10.1093/bib/bbae527`. 2, 35, 36

Stanley F. Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech and Language*, 13(4):359–394, 1999. ISSN 0885-2308. doi: https://doi.org/10.1006/csla.1999.0128. URL `https://www.sciencedirect.com/science/article/pii/S0885230899901286`. 6, 10, 20

Xilun Chen, Yu Sun, Ben Athiwaratkun, Claire Cardie, and Kilian Weinberger. Adversarial deep averaging networks for cross-lingual sentiment classification. *Transactions of the Association for Computational Linguistics*, 6:557–570, 2018. doi: 10.1162/tacl_a_00039. URL `https://aclanthology.org/Q18-1039/`. 25

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoderdecoder for statistical machine translation. In *Conference on Empirical Methods in Natural Language Processing*, 2014. URL `https://api.semanticscholar.org/CorpusID:5590763`. 11

Ratul Chowdhury, Nazim Bouatta, Surojit Biswas, Christina Floristean, Anant Kharkar, Koushik Roy, Charlotte Rochereau, Gustaf Ahdritz, Joanna Zhang, George M. Church, Peter K. Sorger, and Mohammed AlQuraishi. Single-sequence protein structure prediction using a language model and deep learning. *Nature Biotechnology*, 40(11):

1617–1623, 2022. ISSN 1546-1696. doi: 10.1038/s41587-022-01432-w. URL `https://doi.org/10.1038/s41587-022-01432-w`. 40

Alexandra Chronopoulou, Matthew Peters, and Jesse Dodge. Efficient hierarchical domain adaptation for pretrained language models. In Marine Carpuat, Marie-Catherine de Marneffe, and Ivan Vladimir Meza Ruiz, editors, *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1336–1351, Seattle, United States, July 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.naacl-main.96. URL `https://aclanthology.org/2022.naacl-main.96/`. 32

Junyoung Chung, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *NIPS 2014 Workshop on Deep Learning, December 2014*, 2014. 12

Joakim Nøddeskov Clifford, Magnus Haraldson Høie, Sebastian Deleuran, Bjoern Peters, Morten Nielsen, and Paolo Marcatili. Bepipred-3.0: Improved b-cell epitope prediction using protein language models. *Protein Science*, 31(12):e4497, 2022. doi: https://doi.org/10.1002/pro.4497. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.4497`. 36, 40, 64

Maximilian Collatz, Florian Mock, Emanuel Barth, Martin Hölzer, Konrad Sachse, and Manja Marz. Epidope: a deep neural network for linear b-cell epitope prediction. *Bioinformatics*, 37(4):448–455, 09 2020. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa773. URL `https://doi.org/10.1093/bioinformatics/btaa773`. 36, 40, 64

Allen Collins, Anna Thanukos, and Isaac Krone. Reading trees: A quick review, 2020. URL `https://evolution.berkeley.edu/phylogenetic-systematics/reading-trees-a-quick-review/`. Originally written in 1994, updated in 2006 and 2020. 37

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12(null):24932537, November 2011. ISSN 1532-4435. 13

The UniProt Consortium. Uniprot: the universal protein knowledgebase in 2023. *Nucleic Acids Research*, 51(D1):D523–D531, 11 2022. ISSN 0305-1048. doi: 10.1093/nar/gkac1052. URL `https://doi.org/10.1093/nar/gkac1052`. 57

NCBI Resource Coordinators. Database resources of the national center for biotechnology information. *Nucleic Acids Research*, 44(D1):D7–D19, 11 2015. ISSN 0305-1048. doi: 10.1093/nar/gkv1290. URL `https://doi.org/10.1093/nar/gkv1290`. 57

Gabriela Csurka. *A Comprehensive Survey on Domain Adaptation for Visual Applications*, pages 1–35. Springer International Publishing, Cham, 2017. ISBN 978-3-319-58347-1. doi: 10.1007/978-3-319-58347-1_1. URL `https://doi.org/10.1007/978-3-319-58347-1_1`. 22

Bruna Moreira da Silva, David B Ascher, and Douglas E V Pires. epitope1d: accurate taxonomy-aware b-cell linear epitope prediction. *Briefings in Bioinformatics*, 24(3):bbad114, 04 2023. ISSN 1477-4054. doi: 10.1093/bib/bbad114. URL https://doi.org/10.1093/bib/bbad114. 38, 41

Hal Daumé III. Frustratingly easy domain adaptation. In Annie Zaenen and Antal van den Bosch, editors, *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL https://aclanthology.org/P07-1033/. 26

Patricia Medyna Lauritzen de Lucena Drumond, Lindeberg Pessoa Leite, Teofilo E. de Campos, and Fabricio Ataides Braz. Layoutqtlayout quadrant tags to embed visual features for document analysis. *Engineering Applications of Artificial Intelligence*, 122: 106091, 2023. ISSN 0952-1976. doi: https://doi.org/10.1016/j.engappai.2023.106091. URL https://www.sciencedirect.com/science/article/pii/S0952197623002750. 18

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423. URL https://aclanthology.org/N19-1423/. 17, 20, 39

Sushil Chandra Dimri, Richa Indu, Harendra Singh Negi, Neeraj Panwar, and Moksh Sarda. Hidden markov model - applications, strengths, and weaknesses. In *2024 2nd International Conference on Device Intelligence, Computing and Communication Technologies (DICCT)*, pages 300–305, 2024. doi: 10.1109/DICCT61038.2024.10532827. 9

S R Eddy. Profile hidden markov models. *Bioinformatics*, 14(9):755–763, 01 1998. ISSN 1367-4803. doi: 10.1093/bioinformatics/14.9.755. URL https://doi.org/10.1093/bioinformatics/14.9.755. 7

Jeffrey L. Elman. Finding structure in time. *Cognitive Science*, 14(2):179–211, 1990. doi: https://doi.org/10.1207/s15516709cog1402_1. URL https://onlinelibrary.wiley.com/doi/abs/10.1207/s15516709cog1402_1. 10

Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rihawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Debsindhu Bhowmik, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(10), 07 2021. ISSN ISSN 0162-8828. doi: 10.1109/TPAMI.2021.3095381. URL https://www.osti.gov/biblio/1968363. 18, 19, 36, 40

Ahmed Elnaggar, Hazem Essam, Wafaa Salah-Eldin, Walid Moustafa, Mohamed Elkerdawy, Charlotte Rochereau, and Burkhard Rost. Ankh: Optimized protein language model unlocks general-purpose modelling, 2023. URL https://arxiv.org/abs/2301.06568. 19

E. A Emini, J. V HUGHES, D. S PERLOW, and J BOGER. Induction of hepatitis a virus-neutralizing antibody by a virus-specific synthetic peptide. *Journal of Virology*, 55(3):836–839, 1985. ISSN 0022-538X. 39

Emilio Fenoy, Alejando A Edera, and Georgina Stegmayer. Transfer learning in proteins: evaluating novel protein learned representations for bioinformatics tasks. *Briefings in Bioinformatics*, 23(4):bbac232, 06 2022. ISSN 1477-4054. doi: 10.1093/bib/bbac232. URL `https://doi.org/10.1093/bib/bbac232`. 36

Jenny Rose Finkel and Christopher D. Manning. Hierarchical Bayesian domain adaptation. In Mari Ostendorf, Michael Collins, Shri Narayanan, Douglas W. Oard, and Lucy Vanderwende, editors, *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 602–610, Boulder, Colorado, June 2009. Association for Computational Linguistics. URL `https://aclanthology.org/N09-1068/`. 32

Björn Forsström, Barbara Bisawska Axnäs, Johan Rockberg, Hanna Danielsson, Anna Bohlin, and Mathias Uhlen. Dissecting antibodies with regards to linear and conformational epitopes. *PLOS ONE*, 10(3):1–11, 03 2015. doi: 10.1371/journal.pone.0121673. URL `https://doi.org/10.1371/journal.pone.0121673`. 35

Yaroslav Ganin, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky. Domain-adversarial training of neural networks. *J. Mach. Learn. Res.*, 17(1):20962030, January 2016. ISSN 1532-4435. 23

Xuexia Gao and Nan Zhu. Hidden markov model and its application in natural language processing. *Information Technology Journal*, 12:4256–4261, 12 2013. doi: 10.3923/itj.2013.4256.4261. 7

Muhammad Ghifary, W. Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation, 2016. URL `https://arxiv.org/abs/1607.03516`. 24

Ulrike Göbel, Chris Sander, Reinhard Schneider, and Alfonso Valencia. Correlated mutations and residue contacts in proteins. *Proteins: Structure*, 18, 1994. URL `https://api.semanticscholar.org/CorpusID:14978727`. 39

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Commun. ACM*, 63(11):139144, October 2020. ISSN 0001-0782. doi: 10.1145/3422622. URL `https://doi.org/10.1145/3422622`. 23

Joshua T. Goodman. A bit of progress in language modeling. *Computer Speech and Language*, 15(4):403–434, 2001. ISSN 0885-2308. doi: https://doi.org/10.1006/csla.2001.0174. URL `https://www.sciencedirect.com/science/article/pii/S0885230801901743`. 5

Alexander Gorbalenya, Mart Krupovic, Arcady Mushegian, Andrew Kropinski, Stuart Siddell, Arvind Varsani, Michael Adams, Andrew Davison, Bas Dutilh, Balazs Harrach, Robert Harrison, Sandra Junglen, A.M.Q. King, Nick Knowles, Elliot Lefkowitz, Max Nibert, Luisa Rubino, Sead Sabanadzovic, Hélène Sanfaçon, and Jens Kuhn. The new scope of virus taxonomy: partitioning the virosphere into 15 hierarchical ranks. *Nature Microbiology*, 5:668–674, 04 2020. doi: 10.1038/s41564-020-0709-x. 37

Russell D. Gray and Quentin D. Atkinson. Language-tree divergence times support the anatolian theory of indo-european origin. *Nature*, 426(6965):435–439, November 2003. ISSN 1476-4687. doi: 10.1038/nature02029. URL `https://doi.org/10.1038/nature02029`. 2

Han Guo, Ramakanth Pasunuru, and Mohit Bansal. Multi-source domain adaptation for text classification via distancenet-bandits. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34:7830–7838, 04 2020. doi: 10.1609/aaai.v34i05.6288. 1, 31

Jiang Guo, Darsh Shah, and Regina Barzilay. Multi-source domain adaptation with mixture of experts. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4694–4703, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1498. URL `https://aclanthology.org/D18-1498/`. 1, 30

Ian W. Hamley. Peptides for vaccine development. *ACS Applied Bio Materials*, 5 (3):905–944, 2022. doi: 10.1021/acsabm.1c01238. URL `https://doi.org/10.1021/acsabm.1c01238`. 2, 36

Michael Heinzinger, Ahmed Elnaggar, Yu Wang, Christian Dallago, Dmitrii Nechaev, Florian Matthes, and Burkhard Rost. Modeling aspects of the language of life through transfer-learning protein sequences. *BMC Bioinformatics*, 20(1):723, 2019. ISSN 1471-2105. doi: 10.1186/s12859-019-3220-8. URL `https://doi.org/10.1186/s12859-019-3220-8`. 18, 41

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Computation*, 9:1735–1780, 11 1997. doi: 10.1162/neco.1997.9.8.1735. 11

Kuan-Ying A. Huang, Xiaorui Chen, Arpita Mohapatra, Hong Thuy Vy Nguyen, Lisa Schimanski, Tiong Kit Tan, Pramila Rijal, Susan K. Vester, Rory A. Hills, Mark Howarth, Jennifer R. Keeffe, Alexander A. Cohen, Leesa M. Kakutani, Yi-Min Wu, Md Shahed-Al-Mahmud, Yu-Chi Chou, Pamela J. Bjorkman, Alain R. Townsend, and Che Ma. Structural basis for a conserved neutralization epitope on the receptor-binding domain of sars-cov-2. *Nature Communications*, 14(1):311, 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-35949-8. URL `https://doi.org/10.1038/s41467-023-35949-8`. 38

Hanako Ishimaru, Mitsuhiro Nishimura, Hideki Shigematsu, Maria Istiqomah Marini, Natsumi Hasegawa, Rei Takamiya, Sachiyo Iwata, and Yasuko Mori. Epitopes of an antibody that neutralizes a wide range of sars-cov-2 variants in a conserved subdomain

1 of the spike protein. *Journal of Virology*, 98(5):e00416–24, 2024. doi: 10.1128/jvi.00416-24. URL https://journals.asm.org/doi/abs/10.1128/jvi.00416-24. 38

Sarthak Jain and Byron C. Wallace. Attention is not Explanation. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3543–3556, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1357. URL https://aclanthology.org/N19-1357/. 20

C. Janeway. *Immunobiology*. Garland Science, New York, 9th edition, 2012. 36

Frederick Jelinek. Continuous speech recognition by statistical methods. *Proceedings of the IEEE*, 64:532–556, 1976. URL https://api.semanticscholar.org/CorpusID:31408841. 5

Martin Closter Jespersen, Bjoern Peters, Morten Nielsen, and Paolo Marcatili. Bepipred-2.0: improving sequence-based b-cell epitope prediction using conformational epitopes. *Nucleic Acids Research*, 45(W1):W24–W29, 05 2017. ISSN 0305-1048. doi: 10.1093/nar/gkx346. URL https://doi.org/10.1093/nar/gkx346. 40

Hewei Jiang, Yang Li, and Sheng-Ce Tao. Sars-cov-2 peptides/epitopes for specific and sensitive diagnosis. *Cellular and Molecular Immunology*, 20(5):540–542, mar 2023. doi: 10.1038/s41423-023-01001-4. URL https://doi.org/10.1038/s41423-023-01001-4. 35

Daniel Jurafsky and James H. Martin. *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition with Language Models*. Prentice Hall - Online, 3rd edition, 2025. URL https://web.stanford.edu/~jurafsky/slp3/. Online manuscript released January 12, 2025. 4, 5, 6, 7, 9, 10

Guoliang Kang, Lu Jiang, Yunchao Wei, Yi Yang, and Alexander Hauptmann. Contrastive adaptation network for single- and multi-source domain adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 44(4):1793–1804, 2022. doi: 10.1109/TPAMI.2020.3029948. 26, 27

P. Andrew Karplus and Georg E. Schulz. Prediction of chain flexibility in proteins. *Naturwissenschaften*, 72:212–213, 1985. URL https://api.semanticscholar.org/CorpusID:37937734. 39

Yoon Kim. Convolutional neural networks for sentence classification. In *Conference on Empirical Methods in Natural Language Processing*, 2014. URL https://api.semanticscholar.org/CorpusID:9672033. 13, 20

A. Kolaskar and Prasad Tongaonkar. A semiempirical method for prediction of antigenic determinants on protein antigens. *FEBS Letters*, 276, 1990. URL https://api.semanticscholar.org/CorpusID:43441837. 39

Jens H. Kuhn, Gaya K. Amarasinghe, Christopher F. Basler, Sina Bavari, Alexander Bukreyev, Kartik Chandran, Ian Crozier, Olga Dolnik, John M. Dye, Pierre B. H. Formenty, Anthony Griffiths, Roger Hewson, Gary P. Kobinger, Eric M. Leroy, Elke Mühlberger, Sergey V. Netesov ( ), Gustavo Palacios, Bernadett Pályi, Janusz T. Pawska, Sophie J. Smither, Ayato Takada (), Jonathan S. Towner, Victoria Wahl, and ICTV Report Consortium. Ictv virus taxonomy profile: Filoviridae. *Journal of General Virology*, 100(6):911–912, 2019. ISSN 1465-2099. doi: https://doi.org/10.1099/jgv.0.001252. URL https://www.microbiologyresearch.org/content/journal/jgv/10.1099/jgv.0.001252. 65

Miguel Lacerda, Konrad Scheffler, and Cathal Seoighe. Epitope discovery with phylogenetic hidden markov models. *Molecular Biology and Evolution*, 27(5):1212–1220, 01 2010. ISSN 0737-4038. doi: 10.1093/molbev/msq008. URL https://doi.org/10.1093/molbev/msq008. 38, 41

Jens Erik Pontoppidan Larsen, Ole Lund, and Morten Nielsen. Improved method for predicting linear b-cell epitopes. *Immunome Research*, 2:2 – 2, 2006. URL https://api.semanticscholar.org/CorpusID:2341929. 39

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. doi: 10.1109/5.726791. 12

Michael Levitt. A simplified representation of protein conformations for rapid simulation of protein folding. *Journal of Molecular Biology*, 104(1):59–107, 1976. ISSN 0022-2836. doi: https://doi.org/10.1016/0022-2836(76)90004-8. URL https://www.sciencedirect.com/science/article/pii/0022283676900048. 39

Yanghao Li, Naiyan Wang, Jianping Shi, Jiaying Liu, and Xiaodi Hou. Revisiting batch normalization for practical domain adaptation. In *ICLR (Workshop)*. OpenReview.net, 2017. 25

Yunsheng Li, Lu Yuan, Yinpeng Chen, Pei Wang, and Nuno Vasconcelos. Dynamic Transfer for Multi-Source Domain Adaptation . In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10993–11002, Los Alamitos, CA, USA, June 2021. IEEE Computer Society. doi: 10.1109/CVPR46437.2021.01085. URL https://doi.ieeecomputersociety.org/10.1109/CVPR46437.2021.01085. 31

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Salvatore Candido, and Alexander Rives. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023. doi: 10.1126/science.ade2574. URL https://www.science.org/doi/abs/10.1126/science.ade2574. 36, 40, 58, 59, 61

Tao Liu, Kaiwen Shi, and Wuju Li. Deep learning methods improve linear b-cell epitope prediction. *BioData Mining*, 13(1):1, 2020a. ISSN 1756-0381. doi: 10.1186/s13040-020-00211-0. URL https://doi.org/10.1186/s13040-020-00211-0. 40

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Ro{bert}a: A robustly optimized {bert} pretraining approach. *arXiv*, 2020b. URL `https://openreview.net/forum?id=SyxS0T4tvS`. 17, 20

Yuan Liu, Dianke Li, Xin Zhang, Simin Xia, Yingjie Qu, Xinping Ling, Yang Li, Xiangren Kong, Lingqiang Zhang, Chun-Ping Cui, and Dong Li. A protein sequence-based deep transfer learning framework for identifying human proteome-wide deubiquitinase-substrate interactions. *Nature Communications*, 15(1):4519, 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-48446-3. URL `https://doi.org/10.1038/s41467-024-48446-3`. 36

Mingsheng Long, Han Zhu, Jianmin Wang, and Michael I. Jordan. Deep transfer learning with joint adaptation networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 22082217. JMLR.org, 2017. 23

Thang Luong, Hieu Pham, and Christopher D. Manning. Effective approaches to attention-based neural machine translation. In Lluís Màrquez, Chris Callison-Burch, and Jian Su, editors, *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1412–1421, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1166. URL `https://aclanthology.org/D15-1166/`. 13

Anastasia Makarova, Huibin Shen, Valerio Perrone, Aaron Klein, Jean Baptiste Faddoul, Andreas Krause, Matthias Seeger, and Cedric Archambeau. Automatic termination for hyperparameter optimization. In Isabelle Guyon, Marius Lindauer, Mihaela van der Schaar, Frank Hutter, and Roman Garnett, editors, *Proceedings of the First International Conference on Automated Machine Learning*, volume 188 of *Proceedings of Machine Learning Research*, pages 7/1–21. PMLR, 25–27 Jul 2022. URL `https://proceedings.mlr.press/v188/makarova22a.html`. 81

Balachandran Manavalan, Rajiv Gandhi Govindaraj, Tae Hwan Shin, Myeong Ok Kim, and Gwang Lee. ibce-el: A new ensemble learning framework for improved linear b-cell epitope prediction. *Frontiers in Immunology*, 9, 2018. ISSN 1664-3224. doi: 10.3389/fimmu.2018.01695. URL `https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2018.01695`. 40

Christopher D Manning and Hinrich Schütze. Foundations of statistical natural language processing, 1999. URL `https://nlp.stanford.edu/fsnlp/`. Accessed: 2024-09-01. 4, 8

Yishay Mansour, Mehryar Mohri, and Afshin Rostamizadeh. Domain adaptation with multiple sources. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008. URL `https://proceedings.neurips.cc/paper_files/paper/2008/file/0e65972dce68dad4d52d063967f0a705-Paper.pdf`. 29

Merriam-Webster. Antigen, 2023. URL `https://www.merriam-webster.com/dictionary/antigen`. Accessed on December 7, 2023. 34

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013. URL `https://proceedings.neurips.cc/paper_files/paper/2013/file/9aa42b31882ec039965f3c4923ce901b-Paper.pdf`. 4, 10

Saeid Motiian, Marco Piccirilli, Donald A. Adjeroh, and Gianfranco Doretto. Unified deep supervised domain adaptation and generalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017. 26

Juan Mucci, Santiago J. Carmona, Romina Volcovich, Jaime Altcheh, Estefanía Bracamonte, Jorge D. Marco, Morten Nielsen, Carlos A. Buscaglia, and Fernán Agüero. Next-generation elisa diagnostic assay for chagas disease based on the combination of short peptidic epitopes. *PLOS Neglected Tropical Diseases*, 11(10):1–19, 10 2017. doi: 10.1371/journal.pntd.0005972. URL `https://doi.org/10.1371/journal.pntd.0005972`. 2, 36

Van-Anh Nguyen, Tuan Nguyen, Trung Le, Quan Hung Tran, and Dinh Q. Phung. Stem: An approach to multi-source domain adaptation with guarantees. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9332–9343, 2021. URL `https://api.semanticscholar.org/CorpusID:244129594`. 1

H. Allen Orr. Fitness and its role in evolutionary genetics. *EBSCOhost MEDLINE Complete*, 10(8):531–539, 2009. doi: 10.1038/nrg2603. 38

Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010. doi: 10.1109/TKDE.2009.191. 1, 21

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1202. URL `https://aclanthology.org/N18-1202/`. 41

T. Petrie. Probabilistic functions of finite state markov chains. *The Annals of Mathematical Statistics*, 40(1):97–115, 1969. ISSN 00034851, 21688990. URL `http://www.jstor.org/stable/2239201`. 9

Julia Ponomarenko and Marc Van Regenmortel. B cell epitope prediction. *Structural Bioinformatics*, 01 2009. 34, 35

XiPeng Qiu, TianXiang Sun, YiGe Xu, YunFan Shao, Ning Dai, and XuanJing Huang. Pre-trained models for natural language processing: A survey. *Science China Technological Sciences*, 63(10):1872–1897, October 2020. ISSN 1869-1900. doi: 10.1007/s11431-020-1647-3. URL `https://doi.org/10.1007/s11431-020-1647-3`. 17, 18

Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proc. IEEE*, 77:257–286, 1989. URL `https://api.semanticscholar.org/CorpusID:13618539`. 7, 8, 9, 20

Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. *OpenAI's website*, 2018. URL `https://api.semanticscholar.org/CorpusID:49313245`. 17, 20

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1), January 2020. ISSN 1532-4435. 18

Anant Raj, Vinay P Namboodiri, and Tinne Tuytelaars. Mind the gap: Subspace based hierarchical domain adaptation, 2014. 31, 32

Alan Ramponi and Barbara Plank. Neural unsupervised domain adaptation in NLP—A survey. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6838–6855, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.603. URL `https://aclanthology.org/2020.coling-main.603/`. 22

Roshan Rao, Nicholas Bhattacharya, Neil Thomas, Yan Duan, Xi Chen, John Canny, Pieter Abbeel, and Yun S. Song. Evaluating protein transfer learning with tape. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*, page 869, Red Hook, NY, USA, 2019. Curran Associates Inc. 41, 42

Chuan-Xian Ren, Yong-Hui Liu, Xi-Wen Zhang, and Ke-Kun Huang. Multi-source unsupervised domain adaptation via pseudo target domain. *IEEE Transactions on Image Processing*, 31:2122–2135, 2022. doi: 10.1109/TIP.2022.3152052. 31

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C. Lawrence Zitnick, Jerry Ma, and Rob Fergus. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. *Proceedings of the National Academy of Sciences*, 118(15):e2016239118, 2021. doi: 10.1073/pnas.2016239118. URL `https://www.pnas.org/doi/abs/10.1073/pnas.2016239118`. 18, 36, 39, 40, 49, 58

Sara Rosenthal, Pepa Atanasova, Georgi Karadzhov, Marcos Zampieri, and Preslav Nakov. SOLID: A large-scale semi-supervised dataset for offensive language identification. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli, editors, *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 915–928, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.findings-acl.80. URL `https://aclanthology.org/2021.findings-acl.80/`. 2, 28

David E. Rumelhart, James L. McClelland, and PDP Research Group. *Parallel Distributed Processing, Volume 1: Explorations in the Microstructure of Cognition: Foundations*. The MIT Press, 07 1986. ISBN 9780262291408. doi: 10.7551/mitpress/5236.001.0001. URL https://doi.org/10.7551/mitpress/5236.001.0001. 10

Sudipto Saha and G. P. S. Raghava. Prediction of continuous b-cell epitopes in an antigen using recurrent neural network. *Proteins: Structure, Function, and Bioinformatics*, 65(1):40–48, 2006. doi: https://doi.org/10.1002/prot.21078. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/prot.21078. 39

Jose L. Sanchez-Trincado, Marta Gomez-Perosanz, and Pedro A. Reche. Fundamentals and methods for t- and b-cell epitope prediction. *Journal of Immunology Research*, 2017(1):2680160, 2017. doi: https://doi.org/10.1155/2017/2680160. URL https://onlinelibrary.wiley.com/doi/abs/10.1155/2017/2680160. 34, 36

Robert Schmirler, Michael Heinzinger, and Burkhard Rost. Fine-tuning protein language models boosts predictions across diverse tasks. *Nature Communications*, 15(1):7407, 2024. ISSN 2041-1723. doi: 10.1038/s41467-024-51844-2. URL https://doi.org/10.1038/s41467-024-51844-2. 36

Sofia Serrano and Noah A. Smith. Is attention interpretable? In Anna Korhonen, David Traum, and Lluís Màrquez, editors, *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2931–2951, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1282. URL https://aclanthology.org/P19-1282/. 20

C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423, 1948. doi: 10.1002/j.1538-7305.1948.tb01338.x. 5

Tatiana I. Shashkova, Dmitriy Umerenkov, Mikhail Salnikov, Pavel V. Strashnov, Alina V. Konstantinova, Ivan Lebed, Dmitriy N. Shcherbinin, Marina N. Asatryan, Olga L. Kardymon, and Nikita V. Ivanisenko. Sema: Antigen b-cell conformational epitope prediction using deep transfer learning. *Frontiers in Immunology*, 13, 2022. ISSN 1664-3224. doi: 10.3389/fimmu.2022.960985. URL https://www.frontiersin.org/journals/immunology/articles/10.3389/fimmu.2022.960985. 41

Weike Shen, Yuan Cao, Lei Cha, Zhang Xufei, Xiaomin Ying, Wei Zhang, Kun Ge, Wuju Li, and Li Zhong. Predicting linear b-cell epitopes using amino acid anchoring pair composition. *BioData Mining*, 8(1), apr 2015. doi: 10.1186/s13040-015-0047-3. URL https://doi.org/10.1186/s13040-015-0047-3. 39

Harinder Singh, Hifzur Rahman Ansari, and Gajendra P. S. Raghava. Improved method for linear b-cell epitope prediction using antigens primary sequence. *PLOS ONE*, 8(5): 1–8, 05 2013. doi: 10.1371/journal.pone.0062216. URL https://doi.org/10.1371/journal.pone.0062216. 39

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568: 127063, 2024. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2023.127063.

URL https://www.sciencedirect.com/science/article/pii/S0925231223011864. 59

R. Sun, M. G. Qian, and X. Zhang. T and b cell epitope analysis for the immunogenicity evaluation and mitigation of antibody-based therapeutics. *mAbs*, 16(1), 2024. doi: 10.1080/19420862.2024.2324836. 2, 36

Shiliang Sun, Honglei Shi, and Yuanbin Wu. A survey of multi-source domain adaptation. *Information Fusion*, 24:84–92, 2015. ISSN 1566-2535. doi: https://doi.org/10.1016/j.inffus.2014.12.003. URL https://www.sciencedirect.com/science/article/pii/S1566253514001316. 29, 30

Gongbo Tang, Mathias Müller, Annette Rios, and Rico Sennrich. Why self-attention? a targeted evaluation of neural machine translation architectures, 2018. URL https://arxiv.org/abs/1808.08946. 13

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, page 990998, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. doi: 10.1145/1401890.1402008. URL https://doi.org/10.1145/1401890.1402008. 2

Antti Tarvainen and Harri Valpola. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 11951204, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964. 26

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. FEVER: a large-scale dataset for fact extraction and VERification. In Marilyn Walker, Heng Ji, and Amanda Stent, editors, *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1074. URL https://aclanthology.org/N18-1074/. 2

Ilya O Tolstikhin, Bharath K. Sriperumbudur, and Bernhard Schölkopf. Minimax estimation of maximum mean discrepancy with radial kernels. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/5055cbf43fac3f7e2336b27310f0b9ef-Paper.pdf. 30

Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance, 2014. URL https://arxiv.org/abs/1412.3474. 23

Eric Tzeng, Judy Hoffman, Kate Saenko, and Trevor Darrell. Adversarial discriminative domain adaptation. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2962–2971, 2017. doi: 10.1109/CVPR.2017.316. 24, 25

Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *ArXiv*, abs/1607.08022, 2016. URL `https://api.semanticscholar.org/CorpusID:16516553`. 25

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86):2579–2605, 2008. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`. 43

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL `https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf`. 13, 14, 16, 20, 36

Randi Vita, Swapnil Mahajan, James A Overton, Sandeep Kumar Dhanda, Sheridan Martini, Jason R Cantrell, Daniel K Wheeler, Alessandro Sette, and Bjoern Peters. The immune epitope database (iedb): 2018 update. *Nucleic Acids Research*, 47(D1): D339–D343, 10 2018. ISSN 0305-1048. doi: 10.1093/nar/gky1006. URL `https://doi.org/10.1093/nar/gky1006`. 57

Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Trans. Inf. Theory*, 13:260–269, 1967. URL `https://api.semanticscholar.org/CorpusID:15843983`. 9

Jun Wen, Junsong Yuan, Qian Zheng, Risheng Liu, Zhefeng Gong, and Nenggan Zheng. Hierarchical domain adaptation with local feature patterns. *Pattern Recognition*, 124: 108445, 2022. ISSN 0031-3203. doi: https://doi.org/10.1016/j.patcog.2021.108445. URL `https://www.sciencedirect.com/science/article/pii/S003132032100621X`. 32

Carl R. Woese, Otto Kandler, and Mark L. Wheelis. Towards a natural system of organisms: proposal for the domains archaea, bacteria, and eucarya. *Proceedings of the National Academy of Sciences of the United States of America*, 87:4576 – 4579, 1990. URL `https://api.semanticscholar.org/CorpusID:4000940`. 37

Ruijia Xu, Ziliang Chen, Wangmeng Zuo, Junjie Yan, and Liang Lin. Deep cocktail network: Multi-source unsupervised domain adaptation with category shift. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018. 30

Wei Xu, Yi Wan, and Dong Zhao. Sfa: Efficient attention mechanism for superior cnn performance. *Neural Processing Letters*, 57(2):38, 2025. ISSN 1573-773X. doi: 10.1007/s11063-025-11748-8. URL `https://doi.org/10.1007/s11063-025-11748-8`. Available at `https://github.com/Xuwei86/SFA`. 14

Xiangli Yang, Zixing Song, Irwin King, and Zenglin Xu. A survey on deep semi-supervised learning. *IEEE Transactions on Knowledge and Data Engineering*, 35(9):8934–8954, 2023. doi: 10.1109/TKDE.2022.3220219. 27

Xingdong Yang and Xinglong Yu. An introduction to epitope prediction methods and software. *Reviews in Medical Virology*, 19(2):77–96, 2009. doi: https://doi.org/10.1002/rmv.602. URL `https://onlinelibrary.wiley.com/doi/abs/10.1002/rmv.602`. 39

Charles Yanofsky, Virginia Horn, and Deanna Thorpe. Protein structure relationships revealed by mutational analysis. *Science*, 146(3651):1593–1594, 1964. doi: 10.1126/science.146.3651.1593. URL `https://www.science.org/doi/abs/10.1126/science.146.3651.1593`. 38

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. ABCNN: Attention-based convolutional neural network for modeling sentence pairs. *Transactions of the Association for Computational Linguistics*, 4:259–272, 2016. doi: 10.1162/tacl_a_00097. URL `https://aclanthology.org/Q16-1019/`. 13

Rui Zhang and Bret Ulery. Synthetic vaccine characterization and design. *Journal of Bionanoscience*, 12:1–11, 02 2018. doi: 10.1166/jbns.2018.1498. 35

Zhiwei Zhao and Youzheng Wu. Attention-based convolutional neural networks for sentence classification. In *Interspeech*, 2016. URL `https://api.semanticscholar.org/CorpusID:9868483`. 13