UNIVERSITY OF COPENHAGEN
DEPARTMENT OF COMPUTER SCIENCE

Ph.D. Thesis

Sebastián García López

Representation Learning in Protein Science

From Sequence Alignments to Deep Embedding Spaces

Advisor: Wouter Boomsma

This thesis has been submitted to the Ph.D. School of The Faculty of Science, University of Copenhagen on October 15, 2024.

Preface

This work represents the culmination of three years and ten months of continuous research work within the BioML group, vinculated to the Machine Learning Section at the Department of Computer Science (DIKU) at the University of Copenhagen. One year and three months of that period was spent on a collaborative project that was carried out between the BioML group and the Enzyme Research Division of Novonesis A/S, located in Kongens Lyngby, Denmark.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 801199, the Novo Nordisk Foundation through the MLSS Center (Basic Machine Learning Research in Life Science, NNF20OC0062606), and the Pioneer Centre for AI (DNRF grant number P1).

The following thesis is organized into five chapters: Chapter 1 provides the necessary background and context to understand the motivations behind the problems addressed in this work. Chapter 2 prepares the reader to understand the scientific contributions I intend to make in this thesis, which are presented in Chapters 3 and 4. Finally, Chapter 5 offers conclusions and outlines future research perspectives.

Abstract

In the field of protein modelling, protein engineering and bioinformatics in general, two main approaches to protein representation have emerged as the spearheads for a wide range of applications, particularly in machine learning tasks: representations based on Multiple Sequence Alignment (MSA) features and the representation based on embedding spaces. MSA-based representations have long been the gold standard in the field and continue to play an important role, even contributing to the development of algorithms such as AlphaFold2 [1], underlining their continued relevance. However, the rise of embedding spaces has gained tremendous momentum with the advent of Protein Language Models (pLMs) [2], which have become central to many state-of-the-art protein representation algorithms without the need for alignments, such as ESM2 [3] and ProtBERT [4]. Nevertheless, there is no clear guideline or path on when to favour one approach over the other, as both are highly relevant and offer distinct advantages depending on the task, leaving room for further exploration and research in this area.

This thesis aims to offer two contributions to the scientific community concerning these two types of representations. The first contribution is algorithmic, where we propose, as a proof-of-concept, a novel strategy for MSA based on deep generative models and spatial transformations. In this initial contribution, we frame MSA as a spatial transformation problem, providing robust and generalizable alignments for new sequences through the creation of a probabilistic graphical model based on ensembles of variational encoders. The second contribution addresses the prediction of a widely used proxy for protein thermostability: melting temperatures through embedding-based representations. While many state-of-the-art methods in this area depend on global metrics to evaluate model performance, these can often obscure important issues, such as the significant inter-species imbalance within the datasets. This work addresses this challenge and proposes strategies for effectively inducing regression models in which the imbalance between species is prominent.

Resumé

Inden for proteinmodellering, proteindesign og generel bioinformatik har to repræsentationsmetoder vist sig som spydspidser inden for en lang række anvendelser, især inden for maskinlæring: repræsentationer baseret på MSA (Multiple Sequence Alignment) egenskaber og repræsentationer baseret på afbildningsrum. MSA-baserede repræsentationer har i lang tid været den gyldne standard og spiller fortsat en vigtig rolle, hvor deres inklusion i udviklingen af algoritmer såsom AlphaFold2 [1] understreger deres forsatte relevans. Vigtigheden af afbildningsrum har ligeledes fået enormt momentum med fremkomsten af proteinsprogmodeller (pLM'er) [2], der er blevet centrale i mange såkaldte state-of-the-art proteinrepræsentationsmodeller uden behov for alignments, såsom ESM2 [3] og ProtBERT [4]. Ikke desto mindre er der ingen klare retningslinjer for, hvornår man bør bruge den ene tilgang frem for den anden, da begge er yderst relevante og tilbyder forskellige fordele, alt afhængigt af opgaven, hvilket efterlader plads til yderligere undersøgelser og forskning.

Denne afhandling kommer med to bidrag til det videnskabelige samfund vedrørende disse repræsentationsmetoder. Det første bidrag er algoritmisk, hvor vi som proof-of-concept foreslår en ny strategi til MSA-generering baseret på dybe generative modeller og rumlige transformationer. I dette første bidrag formulerer vi MSA-generering som et rumlig transformations-problem, der giver robuste og generaliserbare alignments for nye sekvenser gennem skabelsen af en probabilistisk grafisk model baseret på samlinger af variationelle autoenkodere. Det andet bidrag omhandler forudsigelsen af en ofte brugt proxy for proteiners termostabilitet: deres smeltetemperaturer gennem repræsentationer fra afbildingsrum. Hvor mange state-of-the-art metoder afhænger af globale metrikker til at evaluere modelydeevne kan disse ofte tilsløre vigtige problemer såsom den betydelige ubalance mellem arter i datasæt. Dette bidrag adresserer denne udfordring og foreslår strategier til effektivt at skabe regressesionsmodeller hvori ubalancen mellem arter er fremtrædende.

Acknowledgements

Pursuing a Ph.D. is a journey of both professional and personal rediscovery. Despite the introspection and challenges that come with scientific research training, it is a path where one constantly receives both emotional and technical support. I would like to take this unique opportunity to express my gratitude to everyone who has supported me unconditionally throughout this experience.

First and foremost, I would like to express my deepest gratitude to Wouter Boomsma for all his support throughout my years of training as a researcher. From the very beginning, this journey was full of challenges, as I had been away from academia for nine years before embarking on this path. Wouter opened the door to the academic world that I had long wanted to enter. Throughout this training he has been an incredible mentor, constantly providing support, inspiring ideas and fostering enriching academic discussions that have made this work possible. It has been a true honor to work with someone so brilliant and passionate about science. I also want to thank to Søren Hauberg for his valuable support to making possible the development of one of contributions of this thesis.

I want to thank to all my colleagues in the BioML group with whom I have been privileged to share this journey over the years. It has been a truly enriching experience to work with such clever people from whom I have learnt so much. Their support and solidarity throughout this process has been invaluable. Also, I want to thank to everyone at DIKU for creating such a welcoming and supportive environment. It has been a privilege to work and learn in one of the most enriching environments I have met.

To the Enzyme Research Department at Novonesis, and in particular to Jesper Salomon, I would also like to express my thankfulness. His optimism, enthusiasm and ability to contribute innovative ideas during our meetings with Wouter made my experience over the past year and three months truly unique and enriching.

I would like to express my deepest gratitude to my parents, Clara and Ramón, and my siblings, Christian and Stefania. None of this would have been pos-

sible without their unwavering support. The years 2023 and 2024 were some of the most challenging in my life, as both my mother and brother were diagnosed with cancer. During this time, I had to summon immense mental and emotional strength to overcome these trials and tribulations, but it has also been a time of self-discovery throughout this journey. I would like to dedicate this doctoral thesis to both my family and my wife, for being the pillars of my life, especially in this milestone that I wish to achieve with this work.

I would like to express my heartfelt gratitude to my dearest friends, with whom I have shared decades of friendship. Though we are now separated by distance (Colombia, United States, and Spain), the bonds we have forged remain solid. They have witnessed my growth, my vulnerabilities, and my evolution, and have supported me throughout my journey toward a career in science.

Finally, to my wife Carmen, to whom this thesis is also dedicated, I would like to express my deepest gratitude. You have always been by my side during the most difficult times, offering your unconditional love. Most of all, I am grateful for your decision to share this journey with me, with its difficulties and its accomplishments, including the one I hope to achieve through this thesis.

Table of Contents

Chapter 1

Introduction

In recent years, representation learning has revolutionized the field of artificial intelligence, transforming many areas of scientific discovery. The life sciences, and in particular protein modelling, are no exception to this revolution. In the latter, the way we represent proteins provide key insights into the relationships between sequence, structure and function which can be used for a wide range of tasks including protein property prediction, protein and small molecule design [5], [6], [7].

The classical approach to protein representation has long been based on multiple sequence alignments (MSA), which have been considered the gold standard for describing the relationships between protein sequences and structures and for studying protein evolution [7], [6]. Today, MSA remain highly relevant, especially with the advancements in deep learning. These advancements have contributed to the development of algorithms that have transformed the field of biology and enabled groundbreaking progress of protein structure prediction by models like AlphaFold2. [1].

Beyond the traditional MSA approach, other players in the field of protein modelling have emerged and become very relevant in the past few years. The development of attention mechanisms alongside advances in natural language processing has led to the rise of Protein Language Models (pLMs) as a new paradigm for representing proteins [2]. Based on embedding spaces generated for transformer-based networks, pLM promises to be an alignment-free approach that can be used as a representation for many tasks related to prediction or even protein engineering applications [2], [3], [8]. Likewise, al-

ternative approaches have also been developed with the aim of exploiting the ability of deep learning to generate expressive representations in embedding-based spaces beyond just protein language models (pLMs). These methods include algorithms that capture the representation of protein structures, such as inverse folding, which aims to predict the sequence of proteins from their atomic coordinates, among many other algorithms that follows the same principle [9], [10]. An additional advantage is that embedding-based representations can be combined to create more expressive features for use in high-end applications. This approach has seen significant growth recently.

There is however no general consensus on which type of representation is superior, as this depends largely on the specific use case being addressed. This remains an open research question. The aim of this thesis is to explore the importance of representation spaces from the perspective of machine learning models applied to protein modelling. Specifically, the thesis focuses on algorithm design from two angles: the generalisation and inference of sequence alignments as a basis for representing molecules through deep generative models, and the induction of embedding-based representations for thermostability prediction. The purpose of this introduction is to provide the reader with the context as well as basic concepts necessary to understand the main contributions of this thesis, which are presented in chapters 3 and 4.

1.1 Representations through Multiple Sequence Alignment

Multiple sequence alignment (MSA) has long been a fundamental tool in computational biology, used to identify patterns of similarity between homologous biological sequences. Through the analysis of variation at the residue level, MSA has helped reveal evolutionary couplings between these residues, providing crucial insights into the underlying structure of proteins [6]. MSA has received considerable attention for their integration with deep learning models. This has led to advances in a wide range of biological tasks. In this section, I will give an overview of why MSA remains relevant today and how it has benefited from deep learning in several biological applications, and I will also discuss some introductory concepts in order to provide context to one of the contributions that led to this thesis, which will be explained in detail in Chapter 3.

1.1.1 Importance of Multiple Sequence Alignment

Many biological challenges and applications, such as the design of protein variants for industrial and pharmaceutical purposes, the development of vaccines and the prediction of protein folding, shares a common requirement: to understand biological sequences and their relationship between structure and function from an evolutionary perspective [7], [6]. A key motivation for using MSA is to allow the study of biological sequences by aligning them to identify similar patterns in homologous proteins or families [5], [6], [7]. This approach captures co-evolutionary signals between residues, leading to the identification of conserved regions that provide insights from three main perspectives:

Co-evolution

When we talk about co-evolution in proteins, we refers to the preservation of certain regions that remain relatively unchanged over time, maintaining essential functional interactions [6]. It specifically involves identifying correlated changes between pairs of residues within a sequence alignment. These correlations suggest that residues are structurally or functionally related, often through direct contact sites, and often provide important information

about activities such as protein stability, function, or even structural insights [11], [12], [13], [6]. The rationale behind co-evolution is that regions that have been preserved over time often correspond to structurally critical areas within proteins, so evolutionary pressures tend to preserve these regions because changes could affect protein function. As a result, conserved residues often indicate essential roles in maintaining structural integrity and facilitating functional activity [14], [15], [6]. co-evolution detected by MSA in pairs of residues without observable changes may suggest a functional relationship, which may aid in the identification of protein-protein interactions or binding sites [16].

Function Prediction

Another application of MSA is that by characterizing the conserved regions between sequences belonging to the same protein family, it is possible to predict the type of function that certain proteins have. If a reference protein has a known function, then compared to another protein that has the same conserved region but whose function is unknown, there is a high chance that this protein has a function very similar to that of the reference [17], [18]. At the functional level, MSA highlights conserved regions, known as motifs, which are often associated with key structural or functional roles in proteins. These regions frequently correspond to active sites, binding domains, or interactions with other peptides and protein domains, among other important functions [18], [6].

Structural Knowledge

MSA enables the inference of structural features in biological sequences, such as proteins and nucleic acids (DNA/RNA), by leveraging co-evolutionary information between aligned sequences. This method allows the identification of conserved residues and correlations between mutations, which can be used to generate contact maps essential for accurate prediction of protein structure [6]. Specifically, it has been shown that extracting information about variants found between aligned sequences provides enough information to determine proximity between residues to infer three-dimensional spacing and determine/predict how proteins fold [6],[5], [19], [20], [21], [22], [23], [24], [25], [26], [27]. There are well-established precedents that highlight the importance of using MSA information for protein structure inference. Before

AlphaFold2 became the state-of-the-art (SOTA), which also uses evolutionary data from MSA to infer protein structures (see Section 1.1.2), other techniques based on Direct Coupling Analysis (DCA) [19], such as EVFold (now integrated into EVcoupling) [6], [28], make use of the contact maps derived from protein families captured by DCA to predict protein structures [19], [6]. The general scheme of EVFold is presented in Figure 1.1, as illustrated below.

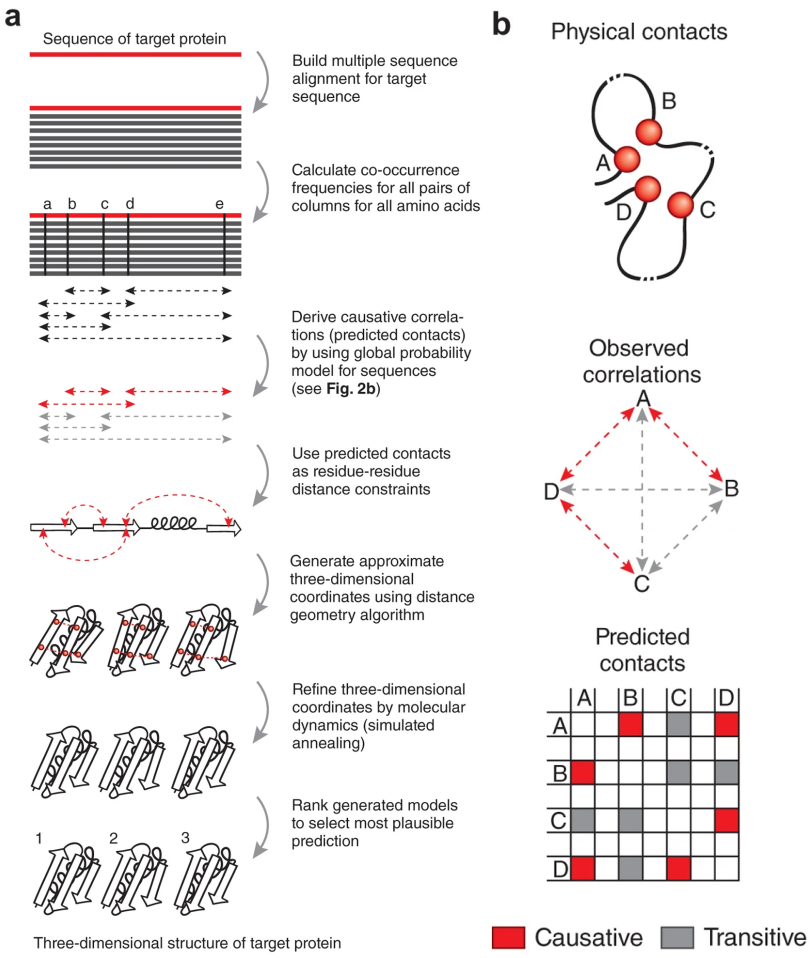

Figure 1.1: General workflow of EVfold: The figure, taken directly from [6], illustrates the process by which evolutionary information is used to derive contact maps, which are then used to predict the 3D structure of a protein sequence.

1.1.2 From Markov Models to Deep Learning: Transitioning strategies to MSA

In conventional biological sequence alignment algorithms, the theoretical basis typically follows the Markov assumption [29], [7] , a key concept in probabilistic modeling, which states that the probability of an event or state is determined only by its previous state, i.e:

$$P\left(x_n \mid x_1, x_2, \ldots, x_{n-1}\right) = P\left(x_n \mid x_{n-1}\right)$$

With this in mind, MSA uses this principle to identify conserved regions and similarities between sequences, providing valuable information on the relationships between them and their biological function [30], [31], [7]. Historically, multiple sequence alignment strategies have relied on probabilistic methods such as Hidden Markov Models (HMMs). These models evaluate the likelihood of each state, whether it is an insertion, match, or gap, given the symbols constituting their corresponding alphabet. This comprises amino acid residues for proteins, nucleotides for RNA and DNA, along with tokens representing gaps. To optimize these methods, it is usually necessary to use dynamic programming to tune the parameters[7], [30].

.

Now, in the current deep learning era, neural network-based approaches have shown extensive capabilities to derive meaningful representations for use in downstream tasks, as well as in many other applications related to protein engineering, among which protein sequence models have achieved outstanding results. In the latter (protein sequence models), deep learning has made it possible to capture with high expressiveness the evolutionary information encapsulated in the conserved regions of the MSA [32]. One of the most popular methods, DeepSequence, involves the use of deep latent variable models, specifically Variational Autoencoders (VAE), trained on MSA of protein families, using these alignments as pre-processed data for model induction. In contrast to the traditional Potts model, this approach captures more complex residue interactions indirectly and learns a more flexible representation of the sequence space, allowing the generative model to predict mutation effects and generate novel sequences [32].

Given the central role of MSA in protein sequence models, recent advances have introduced alternative sequence alignment approaches that leverage

deep learning to generate more informative representations, thereby improving the quality of alignments for use in downstream applications. For instance, [33] proposes the use of differentiable dynamic programming to adapt the Smith-Waterman algorithm into a differentiable framework for end-to-end learning of MSA using Random Markov fields for unsupervised contact prediction applications. However, several drawbacks appear in this regard: One of the disadvantages of this type of model, which is an end-to-end solution, is that the alignment process is already completely decoupled from the model, losing the ability to measure uncertainties in the alignments. In addition, as shown in [34], although MSA is a tool that can be easily integrated into generative models, its use as a preprocessing step can introduce statistical pathologies into such models. Likewise, as an end-to-end solution, it loses the ability to generate or sample biological sequences, which is highly desirable in fields such as protein engineering.

Other recent approaches include the use of attention mechanisms and transformers to learn MSA across protein families (MSAtransformers) [35], techniques that have gained prominence in recent years due to their influence on the development of protein structure prediction algorithms such as AlphaFold2 [1]. However, one of the main challenges of these methods is their high computational cost, and although they achieve high prediction accuracy, studies such as [36] have shown that small perturbations in input sequences can cause drastic changes in downstream tasks such as protein structure prediction. This phenomenon has been reported for both protein language models and MSAtransformers [36]. Approaches based on latent variable models have also emerged, such as the one proposed in [34], which uses latent alignment through a Hidden Markov Model (HMM). This model assumes block-structured emission and transition matrices that allow the alignment of sequences in latent space. Unlike conventional HMMs, this model is estimated by stochastic variational inference, estimating the ELBO gradients by automatic differentiation [34].

1.1.2.1 MSA as Spatial Transformation Problem

Building on the approaches discussed in the introductory section 1.1.2 regarding MSA and its significance in protein science, we formulated the following research questions:

- If it is possible to infer the MSA using generative models, what type of transformation would be appropriate to define such an alignment?
- Is it possible to generalize the alignment inferred by the generative model to new sequences not included in the density estimation?
- Can we reuse existing sequence alignments as a prior distribution to guide and generalize to new sequences, using this prior as a reference?

In this thesis, we aim to demonstrate, as a proof of concept, that the sequence alignment problem can be approached as a spatial transformation task, where the optimal alignment corresponds to the most suitable spatial transformation based on a defined set of parameters. To enhance the interpretability of this transformation from a probabilistic point of view, specifically, inferring the optimal transformation using probabilistic graphical models (PGM), it is ideal for the transformation to be diffeomorphic, i.e., to ensure the existence of a differentiable inverse transformation that approximates the original input space. This property is particularly important when employing probabilistic modeling through variational inference, the approach chosen for this framework. The core idea is illustrated in Figure 1.2. As this proof-of-concept treats MSA as a spatial transformation problem, it is essential to first introduce the notion of spatial transformation. The contribution of this thesis related to the MSA approach as a spatial transformation is explained in detail in Chapter 3, including its mathematical derivation via variational inference. However, the remaining sub-components related to Section 1.1 will introduce the necessary concepts to provide the reader with a basic knowledge to understand the work presented in Chapter 3.

Spatial Transformations for Modeling

Spatial transformation techniques have been essential in image processing and computer vision, enabling modifications such as rotation, scaling, and translation in the input space. These transformations aim to ensure invariance in the representation space, while maintaining such a representation stable despite changes in the input. This in turn improves both robustness and generalization, critical factors for real-world applications [37], [38].

Some approaches like Spatial Transformer Networks (STN) [38], apply direct transformations to data by parameterizing a neural network as localisation

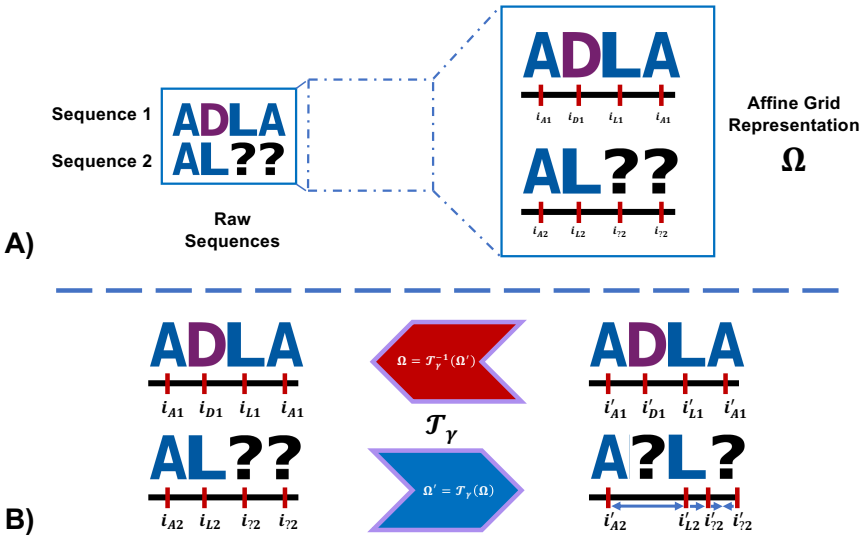

Figure 1.2: Proof of concept for sequence alignment through spatial transformations. In side **A**, each sequence is represented as an affine grid system, where residues are indexed accordingly. In Side **B**, a transformation scheme is introduced, applying a deformation to the affine grid space (indicated by blue small arrows) based on the transformation parameter γ. γ enables the shifting of indices required for alignment. The big red and blue arrows highlighted in purple illustrate the capability of performing both forward and inverse transformations on the input data, providing benefits for probabilistic modeling.

network. The localization network processes the raw input to extract feature maps, which serve as transformation parameters. These parameters are then fed into a grid generator which applies an affine transformation to warp a regular uniform grid based on the estimates provided by the localization network. The output from the grid generator, along with the original input, is used by a sampler to perform interpolation to produce the final transformed output. The attractiveness of STN lies in their ability to apply flexible, non-rigid deformations to signals, thereby enabling the extraction of invariant features for representation. This makes them useful in tasks such as image registration, among other applications [38], [39]. Although STN are very expressive, since one of the scientific questions is to obtain a transformation

that has the property of being invertible, smooth and differentiable in both directions without losing information, i.e. diffeomorphic, STNs do not inherently possess this property [40].

This work adopts an alternative spatial transformation method called Continuous Piecewise-Affine Based Transformations (CPAB) [39]. CPAB is a transformation that allows the parameterisation of non-rigid, smooth and differentiable deformations in the input space, while retaining the property of being fully diffeomorphic. The attractiveness of CPAB lies in its ability to provide highly expressive transformations at low computational cost, making it well suited to probabilistic modelling techniques such as variational inference, Markov Chain Monte Carlo, among others [39],[41]. CPAB has also been successfully incorporated as a structural element to enhance the expressiveness of existing STNs. For instance, Diffeomorphic Transformer Networks [40] replace traditional affine transformations with CPAB, improving expressiveness and performance in classification and regression tasks and the Probabilistic Spatial Transformer Networks [42] which provide a stochastic extension of conventional STNs.

Despite choosing CPAB as the base transformation for inferring alignments, two challenges need to be addressed. The first is how to infer such a transformation probabilistically, i.e. determining an optimal distribution of transformation parameters to statistically infer these transformations. This issue is introduced in Section 1.1.3 and is methodologically detailed in Chapter 3. The second challenge we face is that although CPAB is specifically designed for applications involving continuous signals, such as imaging and physiological data, it has limitations when applied to the estimation of discrete states, such as the alphabets in biological sequences (e.g. amino acids in proteins or nucleotides in DNA), precisely because of the way the transformation is mapped to the output space (linear interpolation). Yet, the CPAB transformation has very attractive properties that make it worth adapting for use in the discrete domain. The adaptation of this transformer, which constitutes part of the contributions, is described in detail in Chapter 3. In this approach, we treat the discrete spaces we want to model, i.e. protein sequences, as categorical distributions.

1.1.3 Generative Modelling through Deep Latent Variable Models

To address the primary problem of inferring transformation parameters for optimal alignment, we need A) a model capable of abstracting the distribution of the initial sequence set to infer these parameters, and B) a model that can scale and generalize the learned distribution across alignments to new sequences within the same family. This requires a flexible and interpretable inference model. Variational inference (VI) provides an efficient solution by treating the transformation parameters as latent variables that can be estimated by approximation. This method allows us to effectively model complex data in a probabilistic framework.

To provide some context for VI, it is a widely used technique in probabilistic machine learning that attempts to approximate complex probability densities through optimization. The approach involves assuming a family of densities to model the target distribution by estimating its reconstruction subject to regularization measures via Kullback-Leibler divergences to prevent the variance from being too small [43].

There are two conditions must be satisfied to perform VI: First, the family of approximate densities must be defined. In this project, we adopt the mean-field approximation, a commonly used approach in which the joint distribution is factorized into individual marginal distributions for each latent variable. Second, an appropriate prior distribution must be specified to impose structure on the latent variables. For this work, we assume that the latent variables follow normal distributions. Likewise, as an additional criterion, we must determine how we want to parameterize the distributions associated with each latent variable in the density. For this work, we chose Variational Autoencoders (VAE) precisely because, in addition to using VI for their construction, they offer versatility in modeling complex densities. Each VAE can be connected in a Direct Acyclic Graph (DAG) to create our final density for the task at hand. More details are provided in subsections 1.1.3.1, 1.1.3.2 and a more deep explanation for constructing the graphical model in Chapter 3.

1.1.3.1 Variational Autoencoder Framework

A common approach to parameterizing complex distributions involves the use of deep latent variable models, of which the variational autoencoder (VAE) is a well-known example. VAEs build on traditional Autoencoders (AE) that use deterministic mappings defined by neural networks to derive compact data representations [44]. Unlike standard autoencoders, VAEs learn probabilistic representations that capture the underlying structure of the data. They achieve this by mapping the input data to a distribution over latent variables, rather than compressing it to a single point. This probabilistic framework facilitates the generation of new data by sampling the latent space, thus increasing the flexibility of the model in handling complex high-dimensional datasets [45], [46].

Alternatively, VAE can be viewed from a probabilistic perspective as a natural extension of probabilistic PCA (pPCA) [46]. Whereas pPCA assumes that latent variables follow normal distributions linked by linear transformations, VAEs incorporate nonlinear transformations, allowing for more sophisticated modeling of complex data [45], [46]. Mathematically, the goal of a VAE is to maximize the likelihood of the observed data $\mathbf{x}$. The marginal likelihood of an input $\mathbf{x}$ is given by:

$$p(\mathbf{x}) = \int p(\mathbf{x}, \mathbf{z})\, d\mathbf{z} = \int p(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z})\, d\mathbf{z}$$

Since this integral is intractable, a variational approximation is required by introducing a latent variable, denoted as $q(\mathbf{z} \mid \mathbf{x})$. By incorporating this approximation, the marginal likelihood can be reformulated as follows.

$$\log p(\mathbf{x}) = \log \int p(\mathbf{x}, \mathbf{z})\, d\mathbf{z} = \log \int p(\mathbf{x} \mid \mathbf{z}) p(\mathbf{z}) \frac{q(\mathbf{z} \mid \mathbf{x})}{q(\mathbf{z} \mid \mathbf{x})}\, d\mathbf{z}$$

$$= \log \left(\mathbb{E}_{q(\mathbf{z}|\mathbf{x})} \left[p(\mathbf{x} \mid \mathbf{z}) \frac{p(\mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})} \right] \right)$$

Applying Jensen's inequality, we get a lower bound on the log likelihood, also known as the Evidence Lower Bound (ELBO):

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log\left(p(\mathbf{x} \mid \mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})}\right)\right]$$

Now, if we break the terms, we obtain:

$$\log p(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log\left(p(\mathbf{x} \mid \mathbf{z})\frac{p(\mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})}\right)\right]$$

$$\geq \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log p(\mathbf{x} \mid \mathbf{z})\right] + \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log\left(\frac{p(\mathbf{z})}{q(\mathbf{z} \mid \mathbf{x})}\right)\right]}_{\textbf{KL divergence}}$$

Now, the corresponding approximation to $p(\mathbf{x})$ is given by:

$$\log p(\mathbf{x}) \geq \underbrace{\mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log p(\mathbf{x} \mid \mathbf{z})\right]}_{\textbf{Reconstruction Loss}} - \underbrace{D_{\mathrm{KL}}\left(q(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})\right)}_{\textbf{KL regularization}}$$

Where the first component represents the reconstruction of the input data, while the second term corresponds to the Kullback-Leibler divergence between the variational distribution $q(\mathbf{z} \mid \mathbf{x})$ used for the approximation and the prior, represented by the latent variable $\mathbf{z}$. The Evidence Lower Bound (ELBO) can thus be expressed as:

$$\text{ELBO} = \mathbb{E}_{q(\mathbf{z}|\mathbf{x})}\left[\log p(\mathbf{x} \mid \mathbf{z})\right] - D_{\mathrm{KL}}\left(q(\mathbf{z} \mid \mathbf{x}) \parallel p(\mathbf{z})\right)$$

Maximizing this ELBO is equivalent to maximizing a lower bound on the log-likelihood $\log p(\mathbf{x})$. It is important to note that the Evidence Lower Bound (ELBO) serves as the objective function for training the densities modeled within this framework.

From an implementation perspective, Variational Autoencoders (VAEs) can be conceptualized as a mapping function composed of two network systems. The first network acts as the encoder, which encodes the input space into a latent space, where the representation is mapped to a probability distribution. The second network, the decoder, takes this latent representation and maps it back to the output space, effectively reconstructing the original input space [45]. Since VAEs are probabilistic and inherently stochastic, to

facilitate sampling from the distribution and allow for gradient propagation, a technique known as the reparameterization trick is employed. This trick introduces a stochastic transformation that enables sampling while preserving the differentiability of the model, thus allowing gradient flow during the learning phase [45].

1.1.3.2 Composing Graphical Models via VAE framework

Once the construction of VAEs and their ability to represent and parameterize distributions is well understood, a crucial question arises: can this framework be extended to model more complex densities?, in particular, can we construct graphical models with multiple latent variables using VAEs as the probabilistic basis?. Several papers have addressed this question, one of the most notable being presented in [47]. This approach uses ensembles of VAEs to model complex relationships between latent variables. Each latent variable is represented by a VAE, allowing the creation of more expressive graphical models for structured representations in probability densities via variational inference. For the purposes of this thesis, we propose using a graphical model to infer the optimal parameters for our spatial transformation, treating these parameters as latent variables within a VAE framework. A more detailed discussion of this approach will be provided in Chapter 3.

1.2 Embeddings as Protein Representation

In the last few years, a new way of modelling proteins has emerged as an alternative to traditional MSAs: embedding-based representations. In contrast to traditional MSA, these directly capture the representation of the protein sequence in a continuous vector space, while using the capabilities of deep learning to obtain a meaningful representation [2]. The popularity of this approach is due to the rise of disciplines such as NLP, which has led to the introduction of language models for protein representation, known as protein language models (pLM) [4], [3].

These models have gained considerable attention for their ability to capture complex relationships within protein sequences using methods adapted from NLP [2]. Building on the concept of embedding spaces as a form of protein representation, other algorithms have been developed that apply this ap-

proach through deep learning models to extract structural information from proteins. A prominent example is inverse folding, which aims to predict protein sequences from the atomic information provided by protein structures, which also use these embedding-based representations [9]. This method is an example of how deep learning can be used to improve the modelling of proteins by using embeddings to effectively infer structural properties [9].

In the following subsections, I will give an overview of how embedding spaces can generate highly expressive representations, along with some of the algorithms used to represent proteins. In addition, I will introduce contrastive representation learning, which will be used in combination with embedding spaces as an experimental framework to construct one of the contributions of this thesis, which will be explained in detail in chapter 4. Finally, I will present the motivation behind the problem at hand, which focuses on predicting thermostability - a task framed as a regression problem. I will also discuss the challenges and complexities involved in developing models to address this problem.

1.2.1 Representations Capacities of Embedding Spaces in Proteins

The revolution and adaptation of embedding spaces as a robust representation framework in proteins came with the development of attention mechanisms, leading to the Transformer architecture [48]. This architecture uses self-attention maps to focus on different segments of text sequences and their relevance, making it highly versatile and robust for dealing long-term dependencies. By doing so, it has effectively replaced recurrent neural networks, such as Long Short-Term Memory Networks (LSTM), which were once the dominant architectures for sequence processing tasks in NLP [49], [50]. The basic transformer algorithm is illustrated in Figure 1.3.

After the success of Transformers and its adoption in natural language processing tasks, several architectures were inspired by this framework, among them BERT (Bidirectional Encoder Representations from Transformers) [51]. Unlike conventional transformers, BERT captures the full context of words and uses a technique known as Masked Language Modelling (MLM), where a certain percentage of tokens are randomly masked to represent the words.

The model then predicts the missing tokens within the sentence based on the surrounding context, providing a more robust representation [51], [50]. Another significant advancement derived from transformers is the development of Generative Pre-trained Transformers (GPTs) [52]. This approach uses a semi-supervised methodology, where the model undergoes an initial unsupervised pre-training on a large volume of data, followed by supervised fine-tuning on a smaller scale for specific tasks [52]. [50].

It is due to the fact that transformers have been assimilated so well into the field of NLP that this type of algorithm has begun to attract the attention of other disciplines, including biology. One of the initial challenges faced by transformer-based architectures as they entered the field of biology was the difficulty of interpreting the representations generated by these models. However, [2] demonstrated that transformer-based models offer a powerful tool for protein science by providing new insights into complex biological data. Their work shows that attention mechanisms can capture high-level structural properties of proteins by establishing spatial relationships between sequence residues and structure, while also capturing higher-order representations that link structure to function [2].

The insight that protein residues could be treated analogously to word sequences in text, as is done in NLP, led to the development of pLMs [2]. This also has led to the emergence of state-of-the-art pLMs such as Evolutionary Scale Modeling (ESM) [3], [53], [54] and ProtBERT [4]. These models are used in downstream tasks, including protein function prediction, structure inference (ESM2Fold), and several applications related to protein engineering [54], [4]. However, while embedding techniques have significantly influenced the development of protein language models, this paradigm has also inspired other advances based on similar principles (embeddings as representations). One such example is Inverse Folding (IF) algorithms [9], which use transformers and graph neural networks to extract highly expressive representations that capture the underlying structural information in protein structures. In the following subsections, we will look more closely at the types of embedding sources that can be used to produce more meaningful representations. This includes those used for the application focus of this chapter, thermostability prediction, as well as the challenges and difficulties associated with predicting thermostability.

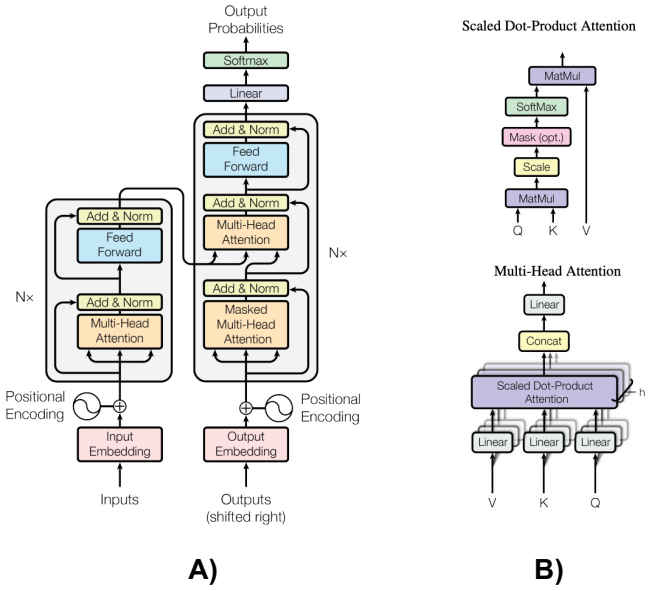

Figure 1.3: General Transformer Architecture: The figure, extracted and adapted directly from [48], depicts the foundational transformer model introduced in the paper Attention is All You Need. Initially developed as an autoregressive model, the transformer architecture has since evolved into various forms, including models like BERT [51] and GPT [52], among others.

1.2.2 Source of Embeddings in Protein Science

A very attractive property of embedding spaces is their ability to be combined or concatenated with other representations of different nature, thus being versatile to extend the richness of features to model many tasks, be it protein functions, protein-protein interactions and other related tasks. Studies such as [55] combine embeddings from protein language models (pLMs) with Gene Ontology (GO) embeddings to predict protein functions. Similarly, [56] integrates pLM embeddings with graph-based embeddings to improve protein-protein interaction prediction. In [57], adopt an approach that merges pLM embeddings with SMILES representations of chemical structures to predict drug-target interactions when there is a limited amount of data. In this subsection, I will describe the types of embedding-based representations used to address the prediction of protein thermostability. First, I

will explain the embeddings derived from protein language models and those based on Inverse Folding used to capture structural features. The rationale for combining these embeddings in thermostability prediction lies in the idea that a richer feature set will lead to a more appropriate representation space, improving prediction performance.

1.2.2.1 Embeddings from pLM - Evolutionary Scale Modeling

As discussed in Section 1.2.1, 1.2.2, protein language models have made significant contributions to the field of biology. This raises the question of which pLM is best suited for representation. In this thesis, for the second project, we selected Evolutionary Scale Modeling 2 (ESM2) as the foundational representation for pLMs. Developed by Meta AI, ESM2 [58], [59] is currently the state-of-the-art in protein language models and has had a substantial impact on bioinformatics and protein engineering.

ESM2, like other pLMs, is an alignment-free method for protein sequence analysis that captures evolutionary relationships by exploiting contextual information between residues without relying on multiple sequence alignment (MSA) data [2], [4], [58], [59]. In terms of training, ESM2 uses a masked language modelling (MLM) approach on a dataset of 250 million sequences. This extensive training allows the model to generalise effectively to large datasets due to the diversity inherent in the sequences used in the training set [3], [58], [59].

In the context of predicting thermostability, as motivated in Section 1.2.4 and discussed in detail in Chapter 4, several works supports the effectiveness of ESM2 as a featurization framework based on protein language models. Noteworthy examples include SaProt [60], FLIP [61], and other relevant works such as [62], [63], [64], and [65]. These studies collectively establish a strong foundation for using ESM2 to represent proteins in this modeling approach.

1.2.2.2 Inverse Folding

Despite the extensive success and versatility of protein language models in learning protein representations, there has been growing interest in extracting information from protein structures for use in areas such as de novo protein

design and protein modelling [9], [66]. As a result, algorithms such as inverse folding (IF) have come into play [9], [10], [66].

One of the best known inverse folding (IF) algorithms that has made use of deep learning as a framework has been presented by [9], by the name of (ESM-IF). This approach centers on learning representations from the atomic coordinates of protein structures, particularly the backbones, to predict corresponding protein sequences. The algorithm was developed in response to the scarcity of available protein structures in contrast to the vast amount of protein sequences available in databases like UniRef [67], [9]. To compensate for this scarcity, the authors augmented the data with predicted protein structures from AlphaFold2 on 12 million protein sequences from UniRef50 [9], [1]. From an algorithmic perspective, IF generates representations that capture structural features using graph neural network based algorithms to preserve equivariance. Specifically, the model incorporates a graph neural network called the Geometric Vector Perceptron (GVP) for structural characterisation, which is then integrated into an autoregressive transformer [9]. It is worth noting that there are different variants of GVPs, such as GVP-GNN [68] and GVP-Transformer [9]. The methodological framework of ESM-IF presented in [9] is shown in Figure 1.4.

Thanks to this strategy of capturing information from protein structures as a representation for modelling, significant efforts have been made in other works in this direction, such as ProteinMPNN [66] and PiFold [10]. In this work, we adopt PiFold as the standard algorithm for generating structure-related embeddings, which will be explained in the following subsection.

PiFold

PiFold is designed to improve residue representation and prediction accuracy by reconstructing features more effectively [10]. It uses a Featurizer to extract information from raw atomic data, which is then processed by a graph neural network framework called PiGNN. This framework captures multi-scale interactions and dependencies between the extracted features [10]. A key advantage of PiFold is its ability to avoid the autoregressive decoders commonly used in many inverse folding algorithms. It also achieves higher protein sequence recovery rates than current methods such as ProteinMPNN, STRUCTGNN and ESM-IF [10].

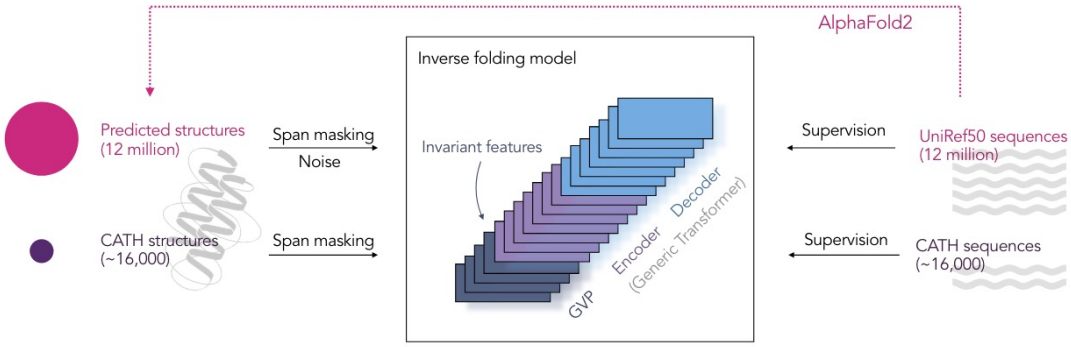

Figure 1.4: Original implementation of the inverse folding algorithm ESM-IF: The figure, taken directly from [9], illustrates the basic implementation of Inverse Folding algorithm made by Meta.

1.2.3 Constrastive Representation Learning

Contrastive methods have gained significant attention in the machine learning community due to their ability to build representations by assessing similarities between samples in the embedding space. By measuring the proximity between samples, those that are closer together are more likely to belong to the same category [69], [70].

Traditionally, this approach has two main variants. The first is a self-supervised method, which generates its own label signal in the absence of labeled data by applying data augmentation to identify relationships between samples. Several studies have explored this strategy [71], [69], [72], [73], [70], [74]. The second variant is the supervised approach, where positive pairs, i.e., examples with a certain degree of similarity, are contrasted against negative pairs. Notable examples of this approach include SupCon [75] and SimCLR [69]. However, these methods have not yet been scaled up to supervised learning tasks with labels of a continuous nature, i.e. regression problems. This has led to the development of algorithms like [71] to compensate for this limitation.

While contrastive methods have achieved significant success across various domains, their application in biological contexts, particularly for regression

tasks involving continuous labels, remains underexplored. Recent studies, such as those by [76] and [71], have started to investigate contrastive representations to tackle the challenges posed by highly imbalanced spaces in regression problems within other fields. However, these efforts offer valuable insights that could support broader adoption of contrastive representation learning in biological research, particularly in regression-oriented tasks, as addressed in this thesis. An overview of the Rank N Contrast Loss, the technique used in this thesis for contrastive representation learning in continuous label settings, will be presented in subsection 1.2.3. A more detailed exploration of the application of contrastive learning methods in the context of thermostability is provided in Chapter 4. To better understand the rationale behind the adoption of contrastive methods, it is recommended to review the motivation for the problem related to thermostability prediction, which is discussed in subsection 1.2.4, which also will be discussed on Chapter 4.

Rank N Contrastive Loss

Most regression methods rely on distance-based loss functions, such as L1 and MSE loss. However, model optimisation focuses mainly on constraining predictions, often ignoring the learned representation space [71]. This oversight often results in fragmented representations that hinder the accurate capture of relationships between data points in regression [71]. Within this framework, Rank N Contrast Loss stands out as a powerful loss function strategy. It contrasts samples based on their ranking within the target space of continuous labels. This approach preserves the relational integrity of each sample, thereby enhancing the continuity of the data geometry [71].

The structure of the implementation of Rank-N-Contrast is based on the decomposition of the problem into two main tasks: a feature encoder and a predictor. The encoder is responsible for the representation of the features, while the predictor maps the representation provided by the encoder to the prediction of the output variable, which in this case is a continuous label. It is important to highlight that the encoder and predictor are two independent components: the feature encoder, responsible for generating the representation, and the predictor operates on this representation in downstream tasks. The Rank-N-Contrast Loss is primarily used to train the feature encoder, whereas the predictor is optimized using a different loss function, such as mean squared error (MSE).

Using the same notation in [71], the $R_N C$ loss per sample is defined as:

$$l_{\text{RNC}}^{(i)} = \frac{1}{2N-1} \sum_{j=1, j\neq i}^{2N} -\log \underbrace{\frac{\exp(\text{sim}(v_i, v_j)/\tau)}{\sum_{v_k \in \mathcal{S}_{i,j}} \exp(\text{sim}(v_i, v_k)/\tau)}}_{\mathbb{P}(v_j | v_i, \mathcal{S}_{i,j})}$$

Where $\text{sim}(\cdot, \cdot)$ is the similarity measure that exists between a pair of embedded samples v_i and v_j, and τ is the temperature parameter. It is noteworthy that $\mathbb{P}(v_j | v_i, \mathcal{S}_{i,j})$ represents the likelihood of the embedded sample, which aims to be maximized. This likelihood determines whether the embedded sample outperforms other samples, with the set being previously ranked based on the [71] is defined as:

$$\mathcal{S}_{i,j} := \{v_k \mid k \neq i, d(\tilde{y}_i, \tilde{y}_k) \geq d(\tilde{y}_i, \tilde{y}_j)\}$$

Finally, the global Rank N Contrast loss would be the contribution of all the $R_N C$ loss calculated for each individual sample, expressed as:

$$\begin{aligned} \mathcal{L}_{\text{RNC}} &= \frac{1}{2N} \sum_{i=1}^{2N} l_{\text{RNC}}^{(i)} \\ &= \frac{1}{2N} \sum_{i=1}^{2N} \frac{1}{2N-1} \sum_{j=1, j\neq i}^{2N} -\log \frac{\exp(\text{sim}(v_i, v_j)/\tau)}{\sum_{v_k \in \mathcal{S}_{i,j}} \exp(\text{sim}(v_i, v_k)/\tau)} \end{aligned}$$

The rationale for this approach is to preserve the geometric continuity of the representation space and the sequential arrangement of the samples. This is achieved by organising samples within a batch based on the proximity of their labels to a reference point, called an anchor. Using this mechanism, the loss function identifies the closest pair of samples and ranks them according to their distances. This process is repeated iteratively until all ranks have been determined. For a more detailed explanation of Rank N Contrast Loss, see [71].

1.2.4 Motivation: Challenges in Thermostability Predictions

The thermal stability of proteins has long been an open question in the field of protein modelling and, in particular, protein engineering. Accurately pre-

dicting thermostability provides critical insight into the integrity of a protein under thermal stress, such as exposure to high temperatures. This knowledge is essential when working with enzymes and is crucial when designing proteins for specific purposes. Although it has long been an open problem, the use of deep learning algorithms to estimate protein thermostability has become very relevant in recent years [77]. Many of these strategies have focused on predicting changes in thermodynamic stability caused by mutations, known as $\Delta\Delta$G. Several approaches have emerged; some strategies based on classical deep learning have been implemented as a means of solving thermostability predictions: Some, such as RaSP, which uses convolutional neural networks to learn the representation of protein structures from their associated atomistic information, end up being used as a pre-trained model to induce a second downstream model, also based on convolutional neural networks, to estimate the changes in protein thermodynamic free energy ($\Delta\Delta$G). [78]. Another strategy, called thermoNet, involves a 3D convolutional neural network that uses features of the protein structure to predict changes in $\Delta\Delta$G. This approach primarily represents protein structures as 3D images by building multichannel voxel grids, each associated with specific biophysical properties such as hydrophobicity, aromatic features and occupancy. These voxel grids are then used as input features to train the neural networks [79]. Alternative approaches using protein language model and inverse folding representations for modelling have developed alongside conventional methods. A notable example is ThermoMPNN [80], which has achieved remarkable results in predicting $\Delta\Delta$G.

In addition to predicting $\Delta\Delta$G as an indicator of thermostability, an alternative approach focuses on the absolute prediction of thermostability, as discussed in studies such as [81]. This method is based on the forecasting of melting temperatures (Tm) which can be defined as the temperatures at which proteins lose structural integrity, going from their native and functional state to a more unfolded or denatured state [82], [83], serving as an indirect measure of thermostability. T_m offers very attractive advantages for consideration: One advantage is its reproducibility in high-throughput assays. Unlike other measures of protein stability, or those assessing stability under specific conditions, the melting temperature is directly comparable across different proteins. This makes it a valuable general optimization target, often correlating with stability under various stress factors [84].

In this respect, new benchmarks related to protein engineering have emerged, such as the Fitness Landscape Inference for Proteins (FLIP), which focuses on protein sequence fitness inference and provides several datasets curated for different tasks, including dataset partitions related to thermal proteome stability across the tree of life from Meltome Atlas, which allows the analysis of thermostability across the melting temperature of several species [61], [83].

However, the use of melting temperature databases is not without its own set of challenges: The melting temperature data available in large databases is aggregated from multiple assays and species. Since we know that different organisms have different optimal growth temperatures, we are likely to observe Simpson's-paradox like effects, where a global cross-species correlation overshadows intra-species correlations, and we might therefore see machine learning methods focus on the global trends, while we are typically interested in predicting the local changes in melting temperature within a species. These issues are addressed in detail in Chapter 4, which presents one of the main contributions of this thesis. This contribution focuses on overcoming the challenges of predicting thermostability using representation spaces derived from embeddings of different sources, including pLM, IF as well as the use of contrastive representation learning.

Chapter 2

Overview

This chapter provides a brief overview of the two papers that summarise my contributions to the field, which are explained in detail in chapter 3 and chapter 4. These works are the result of the time invested during my Ph.D. training, covering topics in machine learning oriented towards bioinformatics problems: The first problem to be solved was the inference and generalisation of MSA via spatial transformations and probabilistic graphical models, providing a density over alignments of protein families. The second problem was aimed at using embedding representations from different sources (pLM, IF) to address challenges in protein thermostability. A more concise outline of each project will follow.

Chapter 3: Probabilistic Multiple Sequence Alignment using Spatial Transformations

This work arose from the question of what would be the best strategy for generating multiple sequence alignments. The question was motivated by some work that has emerged in recent years, such as [34], which has shown that using MSA as pre-processed data to train protein models can lead to potential statistical artefacts that would affect the performance of these protein models. Initially, we were encouraged to find a different mechanism for aligning sequences compared to traditional methods. This led us to question whether this mechanism could perhaps be established by geometric means rather than relying on the Markov assumption as quoted on section 1.1.2 in Chapter 1. One way we thought of solving the problem was to think of a

geometric transformation that would allow us to somehow spatially shift the amino acids to achieve alignment, while making the output representation of the transformation invariant. In this aspect, we found work that inspired us, in the CPAB transformations presented in [39] and their versatility to be used in approaches of a probabilistic nature [41] to capture invariance. We found this to be very appealing, as we were able to formulate the alignment as an invariance problem by disentangling the alignment over the input data.

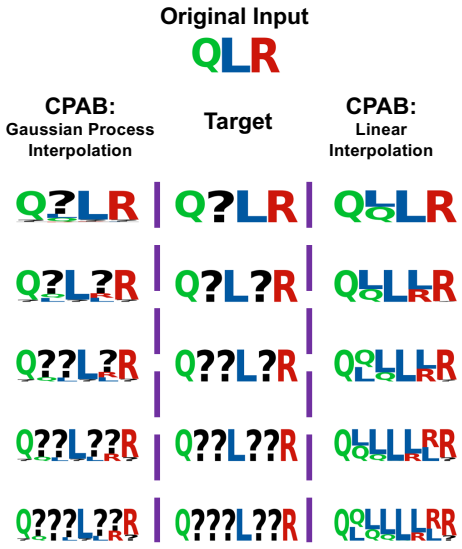

Figure 2.1: Experiment to determine expressivity of CPAB adaption in toy data: To provide context for the experiment, the sequence alphabet consisted of three characters (Q, L, R) and a symbol representing gaps (?). The aim was to find the optimal transformation parameters that allows the warping arround the raw sequences to make it match to the target or reference sequence via cross-entropy.

Adapting this approach to discrete spaces, such as protein sequences, introduced significant technical challenges. While CPAB was originally developed for continuous applications like images and physiological signals, it was not suited for handling categorical distributions. One of the key contributions of this work is the adaptation of this spatial transformation method for categorical distributions (see subsection 1.1.2.1 for an introduction and Chapter 3 for a detailed explanation of the methodology). The initial challenge was to determine whether the adapted CPAB transformations were enough ex-

pressive to function as sequence aligners. To address this, we designed an experiment in which we started with a raw sequence (i.e., a sequence without transformation) and a reference sequence. The objective was to transform the raw sequence to closely approximate the reference, with cross-entropy used to compare the target and transformed sequences. The results are presented in Figure 2.1. As illustrated in Figure 2.1, our proposed adaptation of CPAB transformation (left column) produced more expressive and target-like transformations (middle column). In contrast, the standard CPAB transformations (right column) produced values associated with gaps that converged to nearest neighbour-like values due to the inherent nature of its mapping procedure to produce the transformed output.

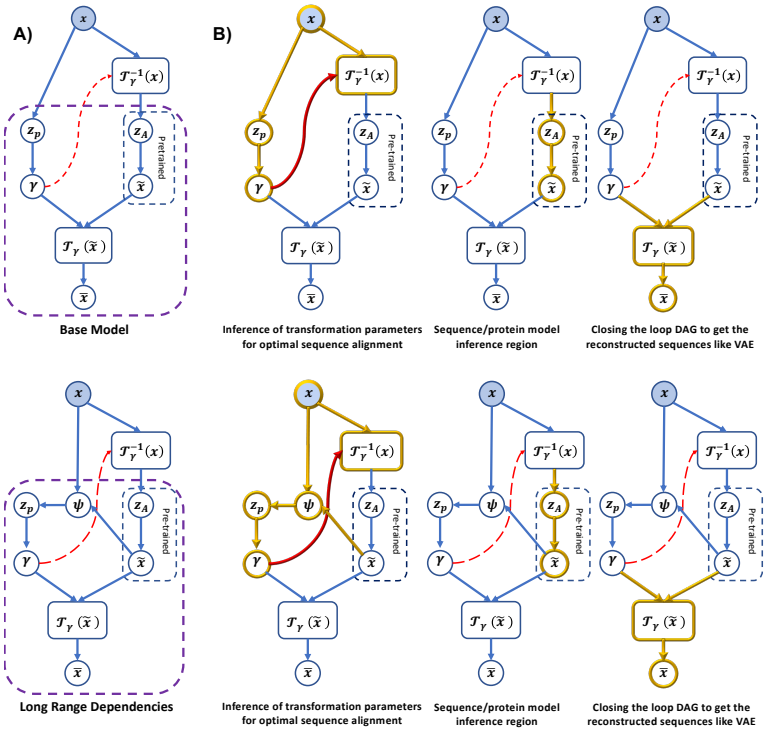

Figure 2.2: PGM scheme to align protein sequences using adapted CPAB transformations. Side A, two PGMs are presented: the first represents the baseline model, an adaptation of [41], while the second introduces our proposal to address long-range dependencies. In side B, The yellow-highlighted regions denote the specific function of each block within the PGM, with corresponding role at the bottom.

Now that we have demonstrated that CPAB transformations can effectively represent sequences, the next objective was to determine whether the optimal parameters for CPAB transformation could be learn using probabilistic methods. Figure 2.2 illustrates a brief workflow of how the PGM operates to infer MSA. However, the full contribution of this proof-of-concept for sequence alignment is presented in Chapter 3, which includes a detailed explanation of the mathematical model, experimental design, and results. This manuscript is currently being submitted to BiorXiv and is expected to be officially available soon.

Chapter 4: Cross-species vs species-specific models for protein melting temperature prediction

This second contribution, carried out in collaboration with Novonesis, addressed a problem of considerable interest to the company's enzyme research division. A widely used but indirect measure of protein thermostability, highly valued for its reproducibility and ability to estimate the denaturation point under thermal stress, is the melting temperature (T_m). Some of the most representative databases providing information on protein thermostability via melting temperatures are the Mealtime Atlas [83] and the Fitness Landscape Inference for Proteins (FLIP) [61], which contains melting temperatures for several species. However, many algorithms designed to predict T_m base their prediction performance on the use of global metrics over the entire datasets, without analysing the implications of this when such predictions are made locally per species. This has several drawbacks: There is usually a high degree of imbalance between the distributions per species, i.e. there are species with more samples than others, which can lead to a large bias in the predictions towards the better represented species. Thus, despite the relatively good performance in global terms, there is a very misleading result when compared in a local scenario per species. Given that the nature of the modelling problem is regression, regression analysis on unbalanced datasets is a new problem that has been addressed by [76] and [71]. However, these studies were on standard datasets and not on datasets related to experimental data, especially in a biological context. Thus, there is no related work in this area in the context of thermostability or in protein engineering related

tasks. Our contribution focuses on analysing this problem in the context of predicting melting temperature as an indicator of thermostability. We present an analysis of this problem and some strategies to compensate for it. The manuscript of this work has also been submitted to BiorXiv and we expect it to be officially available on the web soon.

Chapter 3

Probabilistic Multiple Sequence Alignment using Spatial Transformations

The work presented in this chapter has been submitted to BiorXiv, and is under screening on the platform before finally being made available to the public.

PROBABILISTIC MULTIPLE SEQUENCE ALIGNMENT USING SPATIAL TRANSFORMATIONS

Sebastián García López
DIKU
University of Copenhagen
sebastian.garcia.lopezs@di.ku.dk

Søren Hauberg
Section for Cognitive Systems
Technical University of Denmark
sohau@dtu.dk

Wouter Boomsma
DIKU
University of Copenhagen
wb@di.ku.dk

ABSTRACT

Multiple Sequence Alignment (MSA) has long been a prominent and critical tool in bioinformatics and computational biology. Its importance lies in its ability to provide valuable insights into the relationships between sequences and the evolutionary pressure leading to amino acid preferences at particular sites in a protein. Despite the recent advances in protein language models, MSAs remain critical in many applications, e.g. for state-of-the-art prediction of 3D structure and protein variant effects. Sequence alignment is typically considered a deterministic preprocessing step, leading to a single static MSA. Especially for low-similarity sequences, parts of an alignment will be subject to substantial uncertainty, which is disregarded when processing a static MSA. Earlier, HMM-based approaches handled this uncertainty by considering the full posterior ensemble over alignments. In this paper, we explore whether a similar approach is feasible within a modern deep learning approach, where we move beyond the Markovian restrictions of earlier models. In particular, we consider whether we can learn the alignment process as distribution over spatial transformations, in combination with a deep latent variable model of protein sequences. A proof-of-concept implementation of this work is available at `https://github.com/deltadedirac/Explicit_Disentanglement_Molecules`.

1 Introduction

For decades multiple sequence alignments (MSAs) have played a central role in the computational modeling of protein sequences and a long list of downstream prediction tasks, ranging from early work on protein secondary structure [1] to the recent breakthroughs in 3D structure prediction [2]. For such downstream applications, the alignment procedure is typically considered a *preprocessing* step, conducted once for a given set of sequences. However, there will generally be errors in the alignments caused by poor sequence coverage or approximations in the alignment algorithms. The effect of such errors on downstream performance is rarely analyzed.

From a probabilistic perspective, it would be desirable to model the *distribution* of possible alignments and "integrate out" the alignment, to take the relevant uncertainty into account. This goal was achieved many years ago in the profileHMM model [3], a hidden Markov model which directly modeled insertions, deletions and substitutions and thus constituted a statistical model over possible alignments, which exactly allowed for averaging over the posterior of alignments [4].

While profileHMMs have been highly impactful, and for many years constituted the de facto standard for describing protein families [5], they are no longer the most effective way to describe the amino acid preferences for individual sites within a family. The Markov-assumption underlying HMMs prevents direct modeling of correlations between sites that are distant in sequence but proximal in 3D space. Models such as Potts models and deep latent variable models (VAEs) have been shown to model such effects more reliably [6, 7]. A disadvantage with these methods is that the alignment process is no longer part of the model, and we lose the ability to probe the sensitivity to uncertainty in the alignments. Therefore, a natural question is whether we can combine the benefits of modern protein family models with a probabilistic description of the alignment process. This is the goal of the current manuscript.

Despite the fact that ProfileHMMs could in principle be trained on raw protein sequences, the corresponding optimization problem during training was known to be difficult, and ProfileHMMs were therefore typically built from pre-aligned sequences in practice [4]. In this manuscript, we will attempt to replicate this approach in the setting of deep latent variable models (VAEs) for protein families. In short, we will provide our models with an initial set of aligned sequences, but attempt to model the alignment process within the model, such that we can align sequences directly to the model, and "integrate out" the alignment by sampling alignments from the posterior.

Our approach is based on the framework of Disentanglement Representation Learning, in which we consider the sequence alignment as an invariant representation that can be disentangled from the raw sequence, and the alignment itself is modeled through a diffeomorphic spatial transformation. In this context, MSA is framed as a parametric spatial transformation problem, where the goal is to infer the optimal transformation by warping the input sequences to achieve the sequence alignment. Since parametric spatial transformations depend on transformation parameters, these can be inferred within a probabilistic graphical model via variational inference.

We summarize our contributions as follows:

- We propose that multiple sequence alignment can be approached as a spatial transformation problem, where the optimal transformation is derived via variational approximations. The concept behind this methodological approach is illustrated in Figure 1.
- We adapt Continuous Piecewise-Affine Based (CPAB) transformations to discrete applications.
- We construct a probabilistic graphical model that allows the generalization of alignments on sequences outside the training set.
- The graphical model enables uncertainty quantification for aligned sequences, which we anticipate can improve performance in downstream tasks.

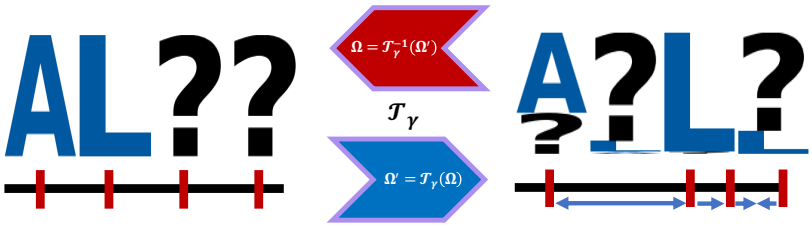

Figure 1: Proposed Methodological Approach: The core idea frames sequence alignment as a spatial transformation problem. Each sequence is represented as an affine grid system, with indices corresponding to individual residues. When passed through the spatial transformer, the system undergoes deformations (indicated by small blue arrows), adjusting the grid and displacing residues to achieve the desired transformation. The optimal estimation of transformation parameters will be addressed using probabilistic modeling, as detailed throughout this work. The transformation, being diffeomorphic, enables the transition between mappings, that is, between the original space of unaligned sequences and their transformed representation of aligned sequences.

2 Related work

Algorithms for multiple sequence alignment has been a topic of interest for more than 30 years. Early work established the foundations for efficient calculations of alignments using dynamic programming [8]. Later, hidden Markov models were used to provide a statistical model of protein sequences related by evolution, using discrete latent states to capture insertions, deletions and substitutions [3]. Once trained, these models produce a multiple sequence alignment of a set of sequences using a dynamic programming algorithm. These so-called Profile HMMs became a standard technique for describing protein families [5], and for homology search [4].

While Profile HMMs accurately capture the amino acid propensities at each site of a protein, their sequential nature makes them ill-suited for describing non-local correlations in a protein sequence. In 2011, several works demonstrated that such pairwise effects could be efficiently modeled using Potts models [6, 9], and that these correlations provided important signals for 3D structure prediction [9]. Higher-order effects were too numerous to model efficiently with the same technique, but later work showed that such correlations could be captured through the continuous latent variable of a variational autoencoder [7]. The likelihood of such models was later shown to correlate well with the pathology of clinical variants [10], and other variant effects [11].

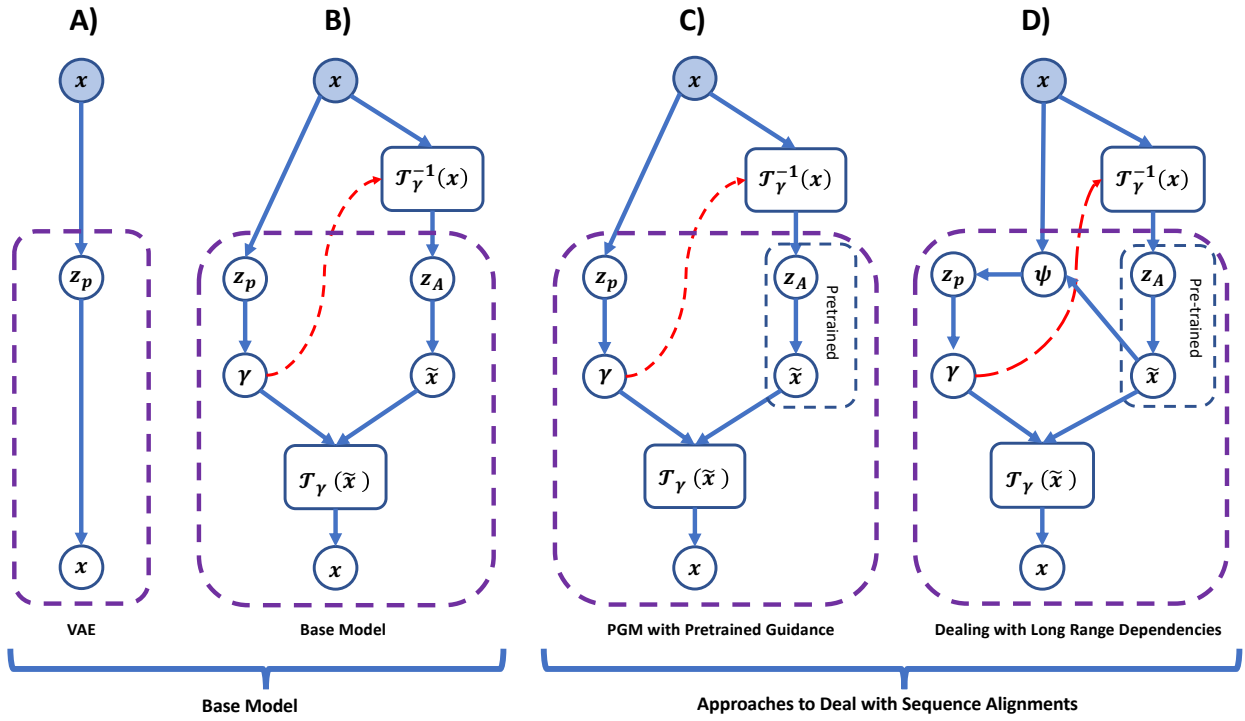

Figure 2: Proposed Framework: A) shows the graphical representation of a Variational Autoencoder (VAE) model, while **B)** shows the graphical representation of the Conditional Variationally Inferred Transformational Autoencoder (C-VITAE) [16]. These models have served as reference frameworks, inspiring the development of this work. On the right, **C)** illustrates the adaptation of the method from [16] for MSA and **D)** shows the model adaptation to deal with long-range dependencies. The proposed strategy partially adapts the methodology from [16] to demonstrate that MSA can be approached as a spatial transformation problem. Key distinctions from the original work in Part B include: 1) adaptation of the CPAB transformation, denoted as $\mathcal{T}_\gamma$, to handle categorical distributions (see Section [17]); 2) incorporation of a pretrained block, indicated by blue dashed lines, to guide the alignment process as an informative prior; and 3) integration of a graphical model that introduces a new latent variable (Ψ) aimed at capturing more accurate features, thereby mitigating flat optimization landscapes.

In the last few years, protein language models have emerged as powerful tools for protein analysis. While such models are potential alternatives to multiple sequence alignments, there are still many cases where they are outperformed by alignment-based methods, especially when many homologous sequences are available. As a consequence, hybrids of the two modeling approaches have been proposed to obtain more robust performance [12]. Other approaches have modelled the alignments themselves using language-model like approaches [13]. Finally, recent work has provided differentiable implementations of the alignment procedure, making it possible to differentiate through the alignment for downstream predictions [14].

Closest to our work is that of Weinstein and Marks [15], who propose a structured observation model that avoids the need for preprocessing sequences into an alignment, and was shown to formally generalize earlier work such as the profile HMM. Our work aims to provide a more efficient alternative by directly predicting the transformations rather than inferring them through a dynamic programming approach.

3 Methods

Our basic approach is illustrated in Figure 2. The purple regions represent the random variables, including latent variables, while x and $\bar{x}$ denote the input and output amino acid sequences, respectively. In Figure 2.C,D, the right branch of the models is a variational autoencoder (VAE) of aligned protein sequences, similar to the DeepSequence model [7], while the left branch is a variational autoencoder that outputs parameters for a spatial transformation. The goal is to infer the optimal transformation $\mathcal{T}_\gamma$ that allows enough deformation to spatially shift the residues to the proper alignment. Spatial transformations are parametric because they rely on transformation parameters (Section 3.1), and these parameters must be inferred (variationally) by inserting them into a graphical model as latent variables. Previous work has successfully inferred the parameters of diffeomorphic spatial transformations to capture invariant representation using graphical models. The model from Detlefsen et al. [16] that inspired our proposal, shown in Figure 2.B, is known as the Conditional Variationally Inferred Transformational Autoencoder (C-VITAE).

As the name suggests, C-VITAE is a generative model structured around a variational autoencoder framework (highlighted in purple in Figure 2.B), involving two latent variables, z_P and z_A. One of these variables, z_P, parameterizes the transformation parameters, γ, which induces the transformation $\mathcal{T}_\gamma^{-1}(x)$. This transformation captures the invariant representation in z_A, facilitating the disentanglement of unique sample attributes encoded in z_P, as well as the invariant features represented in z_A. The full model can be viewed as a large variational autoencoder, where the reconstruction of the original samples x is achieved through $\mathcal{T}\gamma(\tilde{x})$, using the same γ parameters computed for both $\mathcal{T}\gamma^{-1}$ and $\mathcal{T}_\gamma$. In practice, the graphical model is composed of 2 VAEs, i.e. the first one going from z_P to γ and the second one going from z_A to $\tilde{x}$, with two layers of spatial transformations ($\mathcal{T}\gamma^{-1}$ and $\mathcal{T}_\gamma$ respectively) sharing the same transformation parameter γ, which is inferred during training. This is one of the reasons why a fully diffeomorphic spatial transformation is required for modelling purposes, as it allows a return to the observation space for the reconstruction measurements necessary for the computation of the variational approximation. We use the Continuous Piecewise Affine-Based Transformations (CPAB) as a parametric spatial transformation (see [18] and Sections 3.1).

In the original work (see [16] and Figure 2.B), the graphical model evaluated its generative and representational capabilities on image datasets such as MNIST, SMPL, and CelebA, which were modeled as continuous variables. Since protein sequences are inherently categorical and spaced in discrete intervals, we cannot apply the original graphical model directly. Furthermore, we found that the original C-VITAE model struggled with longer-range dependencies. To address these limitations, we propose the schemes shown in Figure 2.C and Figure 2.D.The models use a pre-trained variational autoencoder (VAE) on an initial set of aligned sequences as a prior to guide the alignment process, and introduces a Gaussian process based smoothing approach to model the discrete signal (see section 3). In addition, we propose another graphical model (right side of figure 2.D) that incorporates an additional latent variable to get a richer featurization to achieve more expressive transformations, thus mitigating the problem of long-range dependencies. The corresponding mathematical derivations for its evidence lower bound (ELBO) appear in sections 3.1.3 and 3.1.4.

3.1 Spatial Transformation Layers

As our goal is to define multiple sequence alignment as a spatial transformation problem, it is crucial to first introduce a notion of what a spatial transformation is. Spatial transformation techniques have long been fundamental to image processing and computer vision, allowing modifications on the input space, i.e. rotation, scaling and translation with the purpose of achieving invariance in the representation space. This ensures that the representations remain stable under various transformations or changes in the input, thereby enhancing robustness and generalisation [19], [17].

Spatial Transformer Networks (STN) [17] parameterize a neural network as a localisation network that processes the raw input to extract feature maps that will work as transformation parameters. These transformation parameters are fed into a grid generator which performs an affine transformation to warp a regular uniform grid given the transformation parameters estimated by the localization net. The output from the grid generator, along with the original input, is used by a sampler to perform interpolation to produce the final transformed output. The attractiveness of STN lies in their ability to apply flexible, non-rigid deformations to signals, thereby enabling the extraction of invariant features for representation. This makes them useful in tasks such as image registration, among other applications [17].

For our purposes, we require invertible spatial transformations. We therefore adopt on a special type of spatial transformation method called Continuous PiecewiseAffine Based Transformations (CPAB) [18]. CPAB is a transformation that allows the parameterisation of non-rigid, smooth and differentiable deformations in the input space, while retaining the property of being fully diffeomorphic. The attractiveness of CPAB lies in its ability to provide highly expressive transformations at low computational cost, making it well suited to probabilistic modelling techniques such as variational inference, Markov Chain Monte Carlo, among others [16], [18]. CPAB has also been successfully incorporated as a structural element to enhance the expressiveness of existing STNs. For instance, Diffeomorphic Transformer Networks [20] replace traditional affine transformations with CPAB, improving expressiveness and performance in classification and regression tasks and the Probabilistic Spatial Transformer Networks [21] which provide a stochastic extension of conventional STNs. Since CPAB is a parametric transformation, the optimal transformation parameters can be estimated as a natural part of the parameters in the variational autoencoder.

3.1.1 CPAB Transformation — Fundamentals

Continuous Piecewise-Affine Based Transformations (CPAB) [18] are inspired by conventional affine transformations. Affine transformations are mathematical functions that allow mapping in a way that preserves the spatial representation under geometric changes such as scaling, rotation and translation. Despite their widespread use in image processing, affine transformations are limited by their linear nature, making them unsuitable for modelling non-linear deformations due to their inability to capture complex spatial relationships [22], [17]. Beneficially, affine transformations are, however, diffeomorphic and differentiable over their entire domain.

To overcome the rigidity of conventional affine transformations, the input space is partitioned into multiple regions and different affine transformations are applied to each region. This approach is also known as Continuous Piecewise Affine Transformations (CPA). While CPA increases flexibility, it introduces non-differentiability at the region boundaries, resulting in smooth transitions within regions but not across boundaries [18]. CPAB introduces two changes compared to the CPA approach. First, linear constraints are applied to ensure both continuity and differentiability across the entire transformation space, effectively linking each partitioned region to maintain global continuity. Second, the deformations are produced through the integration of CPA vector fields [18].

The CPAB transformation process starts by partitioning space into small regions using tessellation cells. This creates a grid system that is used in the deformation process. Then, affine functions are applied to construct CPA vector fields for each region, and linear constraints are imposed to ensure continuity across the regions. Each associated affine matrix generates a vector field that facilitates mapping of the initial point to its new position using the transformation parameters v^{ζ}. Finally, the integration of trajectories generated by the vector fields, followed by interpolation, yields the final transformed output produced by CPAB [18, 21]. The transformation is defined by the trajectory of the vector fields, which is determined by solving the following differential equation:

$$\underbrace{\phi^{\zeta}(x,t) = x + \int_0^t v^{\zeta}\Big(\phi^{\zeta}(x,\tau)\Big)\, d\tau}_{\textbf{Integral equation}} \qquad \text{or} \qquad \underbrace{\frac{d\phi^{\zeta}(x,t)}{dt} = v^{\zeta}\Big(\phi^{\zeta}(x,t)\Big)}_{\textbf{ODE equivalent}}$$

where v^{ζ} represents the transformed grid point within the set of affine matrices $\{A_{\zeta_c}\}_{c\subseteq\Omega}$ and Ω is the transformation domain [18]. These transformations define the diffeomorphic map $T^{\zeta} : x \mapsto \phi^{\zeta}(x, 1)$.

3.1.2 CPAB Transformation — Adaptation to Discrete Domain

The standard implementation of CPAB transformations relies on linear interpolation between the deformed affine grid, adjusted by the transformation parameters, and the initial input space to produce the final transformed output. While this approach efficiently transforms continuous signals, it faces challenges when quantifying uncertainty within discrete states. Fortunately, the interpolation in CPAB is decoupled from the rest of the components that enable the affine grid deformations. Since the deformations on the affine grids take place first (the transformation component), to then translate these changes into the output space by mapping, we can think of making these transitions by probabilistic means.

In the context of proteins, each sequence can be viewed as a set of categorical distributions, where the highest probability is assigned to a given residue occurring. Thus, each protein can be represented as a sequence of one-hot encoding vectors. We convert this into a continuous representation by placing the residue observations to occur at unit intervals on the real line, and applying a smoothing operation. For our purpose, we require a smoothing operation, that distributes probability mass locally around the discrete observations, gradually dropping off to a prior probability corresponding to a uniform distribution of the 20 amino acids. We employ Gaussian Processes (GP) for this purpose. Formally, this would require a multi-output GP (MO-GP), that takes into account that the probabilities for the twenty amino acids are coupled by a requirement to sum to one. However, for simplicity, we model the development for each amino acid propensity independently, and renormalize post-hoc. The strategy is visualized in Figure 3.

The interpolation method with MO-GP is defined as follows. The index space is composed of two different grid types: one modified by CPAB deformations and the other uniform. An index is assigned to each vertex that defines a part of the tessellation cell. Each index is associated with a residue and its position in the protein, ensuring that the order of the amino acids in the sequence is preserved. Likewise, each index is assigned a one-hot encoding for the corresponding residue, based on a predefined alphabet. The MO-GP prior is defined using the index space Ω' corresponding to the grids deformed by CPAB and linked to their respective one-hot encoded values Y, as shown in Figure 3 (side A). After defining the MO-GP, an affine uniform lattice is used to map the transformation to the output space. The shift relative to the information present in the MO-GP prior is determined by kriging (Figure 3 - side B). It is important to emphasize that MO-GP operates strictly as an interpolator, with no training, optimization or parameter inclusion in further training schemes. However, the choice of an appropriate length scale is crucial as it governs the influence of nearby data points. Specifically, the length scale defines the distance within the index space at which function values are no longer correlated, thus determining the strength of correlation between data points as a function of their distance.

3.1.3 Derivation of ELBO of Basic Framework from Base Model

The initial inference of optimal CPAB transformation parameters for alignment induction and density estimation is based on the Conditional Variationally Inferred Transformational Autoencoders (C-VITAE) [16] (see Figure 2.B). We

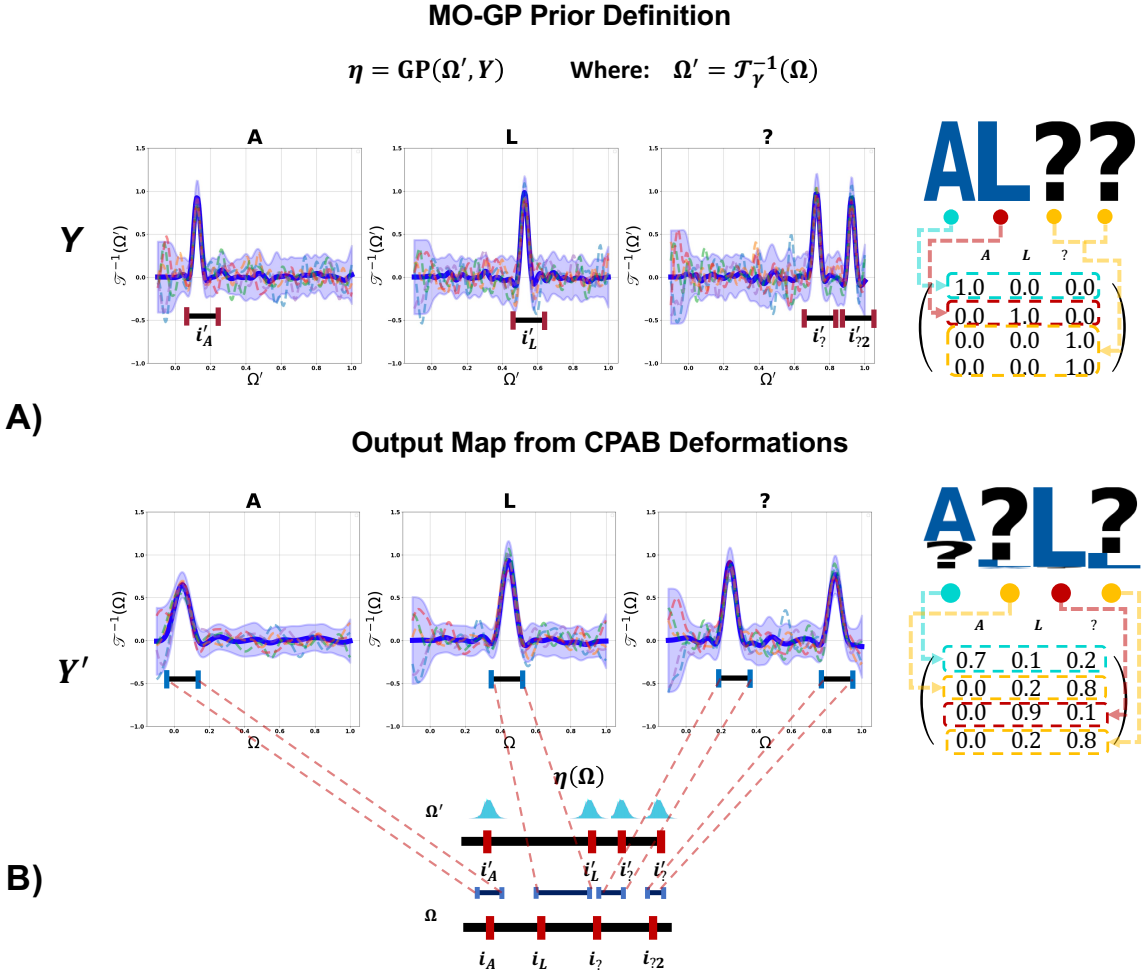

Figure 3: Side A illustrates the preparation of the MO-GP prior, where the deformed vertices of the affine grid Ω', denoted by $\mathbf{i}$, are generated by the CPAB transformation. These serve as input values, while the output values correspond to the one-hot representation Y of the original input sequence. Side B shows how the output mapping would be performed. Both the deformed affine grid with CPAB and the uniform affine grid are required to determine the displacement relative to a reference. This allows interpolation via kriging, resulting in the deformed output space Y'.

derive the evidence lower bound by considering the likelihood

$$p(x) = \int\int p(x|z_A, z_P)p(z_A)p(z_P)dz_A dz_P$$

and defining approximate posteriors for the two latent variables

$$p(z_P|x) \approx q_P(z_P|x); \;\; p(z_A|x) \approx q_A(z_A|x, z_P).$$

We can derive the marginal likelihood:

$$\begin{aligned}
\log p(x) &= \log\left(\int\int p(x|z_A, z_P)p(z_A)p(z_P)dz_A dz_P\right) \\
&= \log\left(\int\int p(x|z_A, z_P)p(z_A)p(z_P)\frac{q_A(z_A|z_P, x)}{q_A(z_A|z_P, x)}\frac{q_P(z_P|x)}{q_P(z_P|x)}dz_A dz_P\right) \\
&= \log\left(\int \mathbb{E}_{q_A(z_A|z_P,x)}\left[\frac{p(x|z_P, z_A)p(z_A)}{q_A(z_A|z_P, x)}\right]p(z_p)\frac{q_P(z_P|x)}{q_P(z_P|x)}dz_P\right) \\
&= \log\left(\mathbb{E}_{q_P(z_P|x)}\left[\mathbb{E}_{q_A(z_A|z_P,x)}\left[\frac{p(x|z_P, z_A)p(z_A)}{q_A(z_A|z_P, x)}\right]\frac{p(z_P)}{q_P(z_P|x)}\right]\right).
\end{aligned}$$

Using Jensen's inequality to change the order of the external expectation with the logarithm gives

$$\begin{aligned}
\log p(x) &\geq \mathbb{E}_{q_P(z_P|x)}\left[\log\left(\mathbb{E}_{q_A(z_A|z_P,x)}\left[\frac{p(x|z_P,z_A)p(z_A)}{q_A(z_A|z_P,x)}\right]\right)+\log\left(\frac{p(z_P)}{q_P(z_P|x)}\right)\right] \\
&\geq \mathbb{E}_{q_P(z_P|x)}\left[\log\left(\mathbb{E}_{q_A(z_A|z_P,x)}\left[\frac{p(x|z_P,z_A)p(z_A)}{q_A(z_A|z_P,x)}\right]\right)\right]-D_{KL}\Big(q_P(z_P|x)\|p(z_P)\Big) \\
&\geq \mathbb{E}_{q_P(z_P|x)}\left[\mathbb{E}_{q_A(z_A|z_P,x)}\left[\log p(x|z_P,z_A)+\log\left(\frac{p(z_A)}{q_A(z_A|z_P,x)}\right)\right]\right]-D_{KL}\Big(q_P(z_P|x)\|p(z_P)\Big) \\
&\geq \mathbb{E}_{q_P(z_P|x)}\left[\mathbb{E}_{q_A(z_A|z_P,x)}\Big[\log p(x|z_P,z_A)\Big]\right]-\mathbb{E}_{q_P(z_P|x)}\left[D_{KL}\Big(q_A(z_A|z_P,x)\|p(z_A)\Big)\right] \\
&\qquad -D_{KL}\Big(q_P(z_P|x)\|p(z_P)\Big).
\end{aligned}$$

The Evidence Lower Bound (ELBO) can now be written as

$$\begin{aligned}
\log p(x) \geq \mathbb{E}_{q_P(z_P|x)}\left[\mathbb{E}_{q_A(z_A|z_P,x)}\Big[\log p(x|z_P,z_A)\Big]\right]-\mathbb{E}_{q_P(z_P|x)}\left[D_{KL}\Big(q_A(z_A|z_P,x)\|p(z_A)\Big)\right] \\
-D_{KL}\Big(q_P(z_P|x)\|p(z_P)\Big).
\end{aligned}$$

This expression represents the ELBO introduced in [16], which is used as a loss function to optimise the density and obtain the optimal values for the adapted CPAB transformation. This ELBO will also serve as a reference for the derivation of an ELBO specifically designed to address long-range dependencies (see Figure 2.D and Section 3.1.4).

3.1.4 ELBO Derivation from the Model to Deal with Long Range Dependencies with Prior

The primary limitation of the base model (Figure 2.B) is its susceptibility to converge in local minima, particularly when dealing with long-range dependencies in large sequences. This likely arises from an insufficient capacity to capture node-specific information in the graphical model, leading to flat optimization landscapes. To mitigate this, we introduce an additional latent variable as a feature extractor, increasing the expressiveness of z_P and refining the CPAB transformation parameters. The updated model (Figure 2.D) includes a new latent variable, Ψ, which uses samples from z_A to inform the distribution over the transformation latent space z_p.

Ψ depends on z_A through $\tilde{x}$, and we can thus write its conditional probability as

$$p(\Psi|x,z_A)=\int p(\Psi|x,\tilde{x})p(\tilde{x}|z_A)d\tilde{x}=\mathbb{E}_{p(\tilde{x}|z_A)}\left[p(\Psi|x,\tilde{x})\right]$$

Although this expectation will generally be expensive to evaluate, we only require a rough estimate, and approximate the expectation with a single Monte Carlo sample. In the following, we choose $p(\Psi|x,\tilde{x})$ to be a Gaussian parameterized by a light attention block.

$$p(\Psi|x,\tilde{x})=\mathcal{N}(\mathrm{LA}_{\theta_1}(x,\tilde{x}),\mathrm{LA}_{\theta_2}(x,\tilde{x}))$$

In practice, we found it to work well to set the variance of this distribution to zero, thus effectively using a delta function.

Since Z_A belongs to the prior distribution and remains independent, that is, it is not affected by other latent variables in the graphical model, the model parameters associated with this prior (blue dashed region of Figure 2.D) are fixed. The graphical model thus gives rise to the following factorization

$$p\left(x,\Psi,z_A,z_P\right)=p(x|z_A,z_p)p(z_p|\Psi)p(\Psi|z_A)p(z_A).$$

Likewise, the evidence can be defined as

$$p(x)=\int_A\int_p\int_\Psi p(x|z_A,z_p)p(z_p|\Psi)p(\Psi|z_A)p(z_A)dz_Adz_pd_\Psi.$$

To derive $\log p(x)$, we get the following

$$\begin{aligned}
\log(x) &= \log\left(\int_A\int_p\int_\Psi p(x|z_A,z_p)p(z_p|\Psi)p(\Psi|z_A)p(z_A)dz_Adz_pd_\Psi\right)\\
&= \log\left(\int_A\int_p\int_\Psi p(x|z_A,z_p)p(z_p|\Psi)p(\Psi|z_A)p(z_A)\frac{q_A(z_A|x)}{q_A(z_A|x)}\frac{q_p(z_p|\Psi)}{q_p(z_p|\Psi)}\frac{q_\Psi(\Psi|x,z_A)}{q_\Psi(\Psi|x,z_A)}dz_Adz_pd_\Psi\right)\\
&= \log\left(\int_A p(z_A)\frac{q_A(z_A|x)}{q_A(z_A|x)}\int_\Psi p(\Psi|z_A)\frac{q_\Psi(\Psi|x,z_A)}{q_\Psi(\Psi|x,z_A)}\int_p p(x|z_A,z_p)p(z_p|\Psi)\frac{q_p(z_p|\Psi)}{q_p(z_p|\Psi)}dz_pd_\Psi dz_A\right)\\
&= \log\left(\mathbb{E}_{q_A(z_A|x)}\left[\frac{p(z_A)}{q_A(z_A|x)}\mathbb{E}_{q_\Psi(\Psi|x,z_A)}\left[\frac{p(\Psi|z_A)}{q_\Psi(\Psi|x,z_A)}\mathbb{E}_{q_p(z_p|\Psi)}\left[p(x|z_A,z_p)\frac{p(z_p|\Psi)}{q_p(z_p|\Psi)}\right]\right]\right]\right).
\end{aligned}$$

By using Jensen inequality, we get

$$\log(x) \geq \mathbb{E}_{q_A(z_A|x)}\left[\log\left(\frac{p(z_A)}{q_A(z_A|x)}\mathbb{E}_{q_\Psi(\Psi|x,z_A)}\left[\frac{p(\Psi|z_A)}{q_\Psi(\Psi|x,z_A)}\mathbb{E}_{q_p(z_p|\Psi)}\left[p(x|z_A,z_p)\frac{p(z_p|\Psi)}{q_p(z_p|\Psi)}\right]\right]\right)\right].$$

At this point, a small substitution can be introduced to facilitate the algebraic manipulation required to derive the final expression, as follows

$$a(x,z_A) = \tfrac{p(z_A)}{q_A(z_A|x)}, \quad b(x,\Psi,z_A) = \tfrac{p(\Psi|z_A)}{q_\Psi(\Psi|x,z_A)}, \quad c(x,\Psi,z_A,z_p) = p(x|z_A,z_p)\tfrac{p(z_p|\Psi)}{q_p(z_p|\Psi)}. \tag{1}$$

Then, we get

$$\begin{aligned}
\log(x) &\geq \mathbb{E}_{q_A(z_A|x)}\left[\log\left(a(x,z_A)\mathbb{E}_{q_\Psi(\Psi|x,z_A)}\left[b(x,\Psi,z_A)\mathbb{E}_{q_p(z_p|\Psi)}\left[c(x,\Psi,z_A,z_p)\right]\right]\right)\right]\\
&\geq \mathbb{E}_{q_A(z_A|x)}\left[\mathbb{E}_{q_\Psi(\Psi|x,z_A)}\left[\log\left(b(x,\Psi,z_A)\right)\right]\right] + \mathbb{E}_{q_A(z_A|x)}\left[\mathbb{E}_{q_\Psi(\Psi|x,z_A)}\left[\mathbb{E}_{q_p(z_p|\Psi)}\left[\log\left(c(x,\Psi,z_A,z_p)\right)\right]\right]\right]\\
&\qquad + \mathbb{E}_{q_A(z_A|x)}\left[\log\left(a(x,z_A)\right)\right].
\end{aligned}$$

where

$$\begin{aligned}
&\mathbb{E}_{q_A(z_A|x)}\left[\log a(x,z_A)\right] = -D_{KL}\left(q_A(z_A|x)\|p(z_A)\right)\\
&\mathbb{E}_{q_\Psi(\Psi|x,z_A)}\left[\log b(x,\Psi,z_A)\right] = -D_{KL}\left(q_\Psi(\Psi|x,z_A)\|p(\Psi|z_A)\right) \qquad .\\
&\mathbb{E}_{q_p(z_p|\Psi)}\left[\log c(x,\Psi,z_A,z_p)\right] = \mathbb{E}_{q_p(z_p|\Psi)}\left[\log p(x|z_A,z_p)\right] - D_{KL}\left(q_p(z_p|\Psi)||p(z_p|\Psi)\right)
\end{aligned}$$

Expanding $a(\cdot)$, $b(\cdot)$, $c(\cdot)$, we obtain our final expression:

$$\begin{aligned}
\log(x) \geq\ & \mathbb{E}_{q_A(z_A|x)}\left[\mathbb{E}_{q_\Psi(\Psi|x,z_A)}\left[\mathbb{E}_{q_p(z_p|\Psi)}\left[\log p(x|z_A,z_p)\right]\right]\right] - \mathbb{E}_{q_A(z_A|x)}\left[\mathbb{E}_{q_\Psi(\Psi|x,z_A)}\left[D_{KL}\left(q_p(z_p|\Psi)||p(z_p|\Psi)\right)\right]\right]\\
&- \mathbb{E}_{q_A(z_A|x)}\left[D_{KL}\left(q_\Psi(\Psi|x,z_A)\|p(\Psi|z_A)\right)\right] - D_{KL}\left(q_A(z_A|x)\|p(z_A)\right).
\end{aligned}$$

This last expression represents the ELBO that will be used to make the density estimation in the presence of long-range dependencies on Figure 2.D

4 Experimental Setup

4.1 Experimental Design

To evaluate the effectiveness of our proof-of-concept framework for sequence alignment, two types of experiments were conducted.

Experiment 1 - MSA Feasibility on Synthetic Data The primary objective of this experiment is to assess whether spatial transformations, using the modified CPAB transformations for discrete domains, are expressive and robust enough to produce consistent sequence alignments. To this end, a pre-trained model is built using a Variational Autoencoder with four synthetic sequences, serving as an informative prior to guide the sequence alignment process. The same sequences used to pre-train the prior are also used to train the graphical model, to infer the optimal transformation parameters to achieve the best alignment. The goal is to determine whether, by (over)fitting the density on these sequences, the model can converge to an appropriate alignment and demonstrate robustness as a method for aligning real protein sequences.

Experiment 2 - Capacity of Transformations to Generalize Alignments to New Sequences The second experiment aims to demonstrate that training our model using a pre-trained prior with a very small amount of proteins, can effectively infer the alignment of both sequences employed for density estimation and new sequences to be aligned. It is important to emphasize that the proteins used to train the prior, those for density estimation, and the test set are entirely different, with no overlap among these three groups. Another notable distinction in this second experiment is the use of a limited amount of data to induce the pre-trained prior as well as for density estimation (see Sec. 4.2). The intention behind using relatively small training sets for both the pre-trained and graphical models is to empirically demonstrate the effectiveness of this approach in performing alignments and its ability to generalize such alignments to new sequences beyond the base knowledge

4.2 Datasets

For the first experiment, a small synthetic dataset was created, consisting of four synthetic proteins based on a five-character alphabet, including three amino acids and two gap symbols. The pre-trained model used the gap symbol $-$, while the symbol ? was used for padding in the density estimation model. The rationale behind this is to prevent flat landscapes during the evaluation of the loss function. The same gap scheme has been applied for both experiments 1 and 2. For the second experiment, we used real protein data, specifically those associated with the WW domain (RSP5 domain) [23]. Protein sequences were obtained from InterPro, with manually curated seed alignments. From this dataset, 224 proteins were selected as a pre-trained prior, while 60 proteins were used to train the graphical model. An additional 15 proteins formed the test set. The 224 proteins were previously aligned using Clustal Omega [24], and the pre-trained prior was trained on aligned sequences using variational autoencoders according to the DeepSequence methodology [7].

4.3 Details about architectures and parameterization of the model

The construction of the graphical models is based on a composition of interconnected VAEs designed for density estimation [25]. The base model, following [16], consists of a pre-trained VAE model from Z_A to $\tilde{x}$, and a block from Z_p to γ, which estimates transformation parameters for the spatial transform layers (Figure 2.C). The VAE configuration consists of 3 layers for both the encoder and decoder. The output generated by γ in the graphical model varies depending on the tessellation cells selected by the spatial transformation layer, with an 8-cell tessellation used for the first experiment and 1750 cells for the second. A higher number of partitions improves the integration region in the CPAB vector field. Additionally, a length scale of 0.5 was applied to the MO-GP to capture covariance between grid components more effectively, enabling more flexible data transformations. For the light attention layer represented by Ψ (Figure 2.D) in the second experiment, the parameterization follows [26], except for modifications in the input structure, as described in Section A.2. Furthermore, the number of channels in the light attention layer was set to match the alphabet used to represent sequences, i.e., 22 channels.

4.4 Optimization Setup

AdamW was used as the main optimiser. The learning rate (lr) was adjusted according to the type of experiment: In the first experiment, the pre-trained model used 1000 epochs with a lr $= 1 \times 10^{-5}$, while the density was trained with a lr $= 1 \times 10^{-4}$ for 600 epochs. In the second experiment, the pre-trained model was trained for 2000 epochs with the same lr $= 1 \times 10^{-5}$, and the density was again trained with a lr $= 1 \times 10^{-4}$ for 600 epochs.

5 Results and Discussion

5.1 MSA Feasibility on Synthetic Data

Figure 4 illustrates the results of the first experiment. A closer examination of the sequence alignment generated (Figure 4.B) reveals that the synthetic amino acids were correctly aligned across columns, indicating that the transformation

inferred by the graphical model is robust enough to estimate the transformation parameters accurately and induce proper alignments. These results suggest several findings regarding the use of the adapted transformation $\mathcal{T}_{\gamma}^{-1}(x)$ in discrete domains, among them: we demonstrate that the approach based on spatial transformations can indeed successfully align sets of biological sequences in batches. This approach differs from traditional methods based on Markov assumptions, such as HMMs, in that it does not assume alignment based on previous states of the sequences to align residues, but instead uses vertex shifts between residue positions in aligned columns. In addition, it provides the ability to quantify uncertainty in alignments using MO-GP mappings built into the CPAB modification.

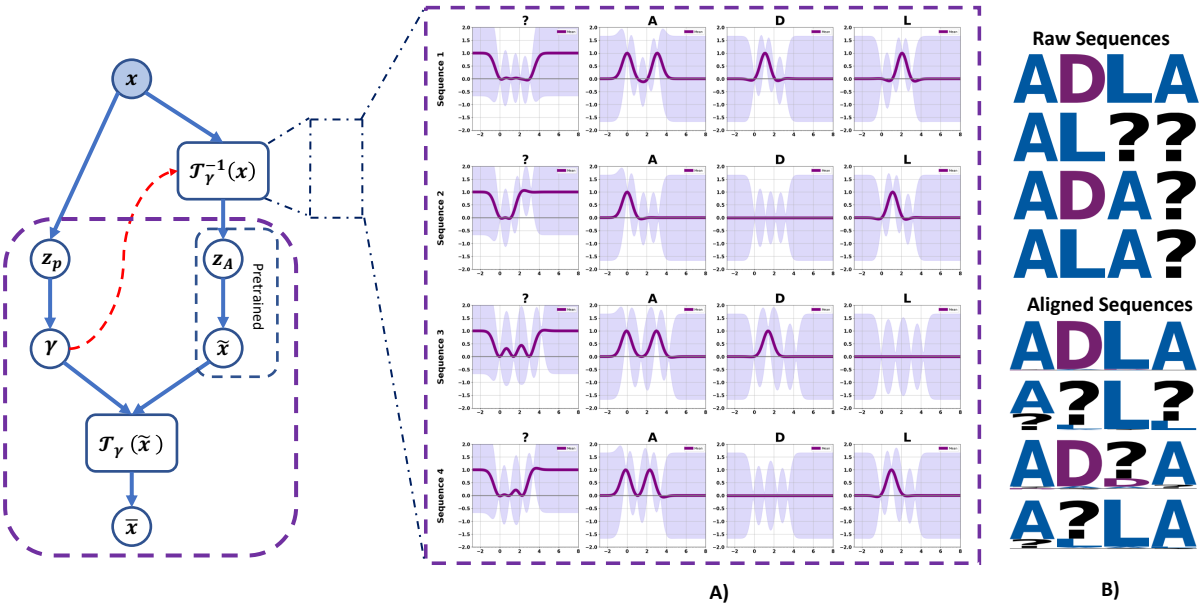

Figure 4: Results of alignments on synthetic data: Side **A)** describes four sequences, each accompanied by their designated names, arranged in rows, and four channels identified by alphabetical symbols. In the GP interpolation plots for each channel, the x-axis represents the index space containing the continuous values of the affine grid and their offset, while the y-axis indicates the probability value of the given state or channel component. In particular, the Multi-Output Gaussian Process (MO-GP) includes four channels or states, represented by the symbols ?, A, D and L, which form the alphabet used for interpolation between discrete states. In addition, the MO-GP prediction used to fit the CPAB transform to the modified CPAB sampler is the mean of the posterior distribution over the channel, with the uncertainty of the prediction represented by a fluctuation region (shown as a purple-shaded area). Side **B)** illustrates the final transformation inferred by the graphical model, where the upper part shows the initial input or its original appearance, while the lower part shows the final result of the sequence alignment as inferred by the spatial transformation layer $\mathcal{T}_{\gamma}^{-1}(x)$.

On the other hand, Figure 4.A shows the interpolations by the MO-GPs within the CPAB modification to estimate the likelihood of each residue given its position in the deformed affine grids. An important observation in the deformed affine grids is that the peaks align with the axes corresponding to the vertices. This alignment is significant because it allows for a geometric interpretation: within each sequence per channel, there should be a positional point of convergence. For instance, if a given column contains A or D, these elements should converge spatially to a similar point along the transformed grids for each sequence. In general, the Spatial Transformer is expressive enough to perform probabilistic alignments consistently. However, it must be acknowledged that this evaluation is based on an overfitting of the graphical model using identical sequences for both input and training of the pre-trained model (aligned sequences for the prior and unaligned for global model). It remains to be determined whether this observed performance would be maintained if the sequences used for pre-training the model were completely different from those used for training the graphical model, as well as for the new sequences to be evaluated given the trained density. In addition, the ability of the model to generalise the alignment to novel sequences is an important consideration. These issues are explored in the next section.

5.2 Capacity of MSA generalization to new sequences

The second experiment use real-world data from the WW domain (Figure 5) to assess the ability of the graphical model to generalise sequence alignments for novel sequences and to deal with long-range dependencies in large sequences (see Figure 2.D). In particular, none of the protein sets used for pretraining, density estimation, or testing had any common protein, ensuring robust alignment for a completely novel sequence.

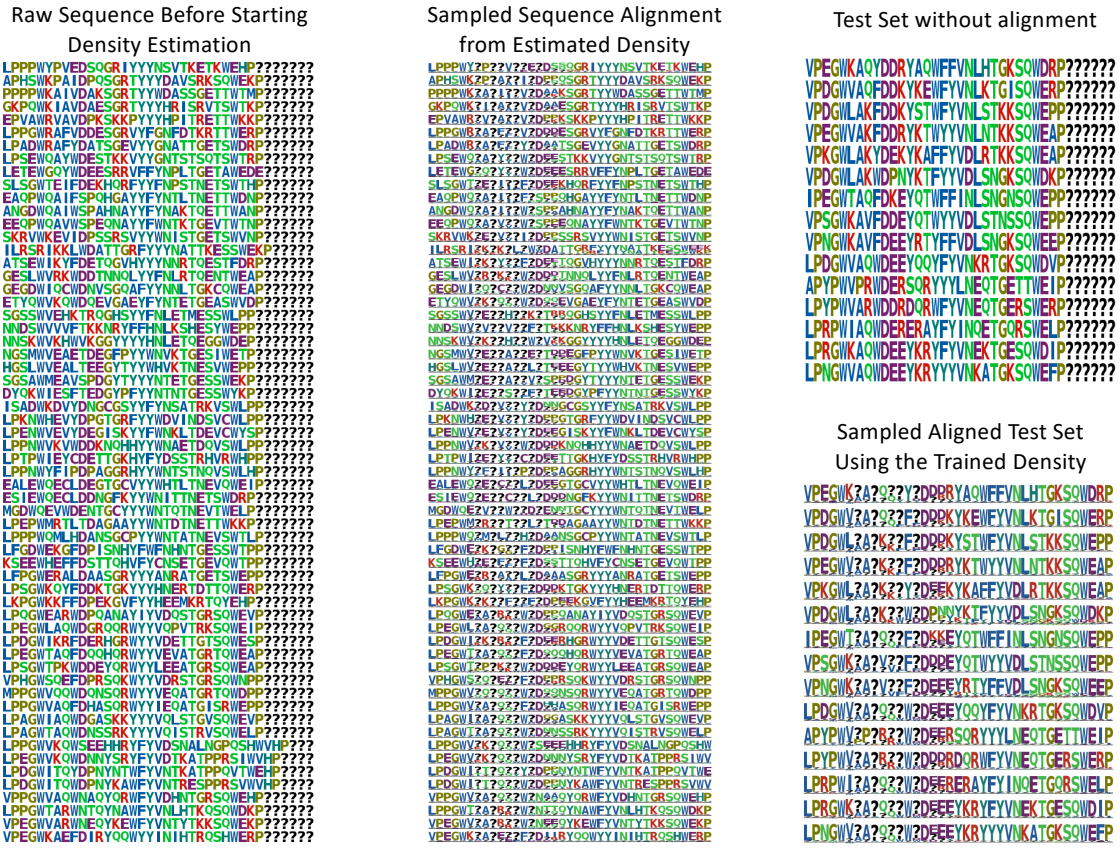

Figure 5: Generalization of sequence alignment on real data: The figure displays four plots showing the status of proteins associated with the WW domain. The plots show the unaligned sequences used to train the graphical model, the alignment of these sequences after density estimation, and the unaligned test sequences alongside their inferred alignments.

As shown in Figure 5, the model demonstrates robustness in both alignment and generalisation to novel, long sequences. This suggests that the inclusion of the stochastic network, modelled as a latent variable in the ELBO derivation, effectively captures the expressive features necessary for accurate alignment and representation.

It should be noted that the prior used was a VAE trained on 224 aligned sequences from the WW domain. On the basis of the results, it can be suggested that the model uses this prior as basic guidance to infer the optimal transformation to produce alignments that are similar to the reference information of the prior. This opens up the possibility of using VAEs on pre-existing sequence alignments as informative priors to recycle them and generate alignments based on a reference framework. However, the effectiveness of this alignment guide will depend on the quality of the training over the prior. Nevertheless, this approach has significant potential for the development of reference-guided alignments.

An important observation regarding Figures 2.D and 6.B is that, based on the results, it can be suggested that the light attention block Ψ is extracting sufficiently high-quality features for z_P to learn the transformation parameters, leading to accurate and precise alignments for this particular example. Furthermore, the generalization of this alignment to new sequences indicates that the distribution learned by z_P preserves the geometric transformation profiles specific to this protein family. By sampling from this distribution, the geometric structure of the features can be conserved across the family. This pattern recognition is analogous to the profile models used in Hidden Markov Models (HMM), such as HMMER [27], and may offer potential for future modelling approaches.

Conclusion and Outlooks

As a proof of concept, the model yields promising results in multiple sequence alignment and generalisation through deep generative modelling. However, further testing and validation in specific applications such as mutation effect prediction and other protein engineering tasks are essential. In addition, it is important to assess the behaviour of

the graphical model when it is estimated from scratch, as the current approach relies on an informative prior and introduces a new latent variable to improve feature extraction and inference of transformation parameters for long-range dependencies. While the model itself remains acyclic in this setting, the full end-to-end estimation procedure will have cyclic dependencies, which complicates parameter estimation in the model. We leave a further exploration of this matter for future work.

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 801199, the Novo Nordisk Foundation through the MLSS Center (Basic Machine Learning Research in Life Science, NNF20OC0062606), and the Pioneer Centre for AI (DNRF grant number P1).

References

[1] Burkhard Rost and Chris Sander. Prediction of protein secondary structure at better than 70% accuracy. *Journal of molecular biology*, 232(2):584–599, 1993.

[2] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[3] Anders Krogh, Michael Brown, I Saira Mian, Kimmen Sjölander, and David Haussler. Hidden markov models in computational biology: Applications to protein modeling. *Journal of molecular biology*, 235(5):1501–1531, 1994.

[4] Robert D Finn, Jody Clements, and Sean R Eddy. Hmmer web server: interactive sequence similarity searching. *Nucleic acids research*, 39(suppl_2):W29–W37, 2011.

[5] Jaina Mistry, Sara Chuguransky, Lowri Williams, Matloob Qureshi, Gustavo A Salazar, Erik LL Sonnhammer, Silvio CE Tosatto, Lisanna Paladin, Shriya Raj, Lorna J Richardson, et al. Pfam: The protein families database in 2021. *Nucleic acids research*, 49(D1):D412–D419, 2021.

[6] Sivaraman Balakrishnan, Hetunandan Kamisetty, Jaime G Carbonell, Su-In Lee, and Christopher James Langmead. Learning generative models for protein fold families. *Proteins: Structure, Function, and Bioinformatics*, 79(4):1061–1078, 2011.

[7] Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

[8] David J Lipman, Stephen F Altschul, and John D Kececioglu. A tool for multiple sequence alignment. *Proceedings of the National Academy of Sciences*, 86(12):4412–4415, 1989.

[9] Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3d structure computed from evolutionary sequence variation. *PloS one*, 6(12):e28766, 2011.

[10] Jonathan Frazer, Pascal Notin, Mafalda Dias, Aidan Gomez, Joseph K Min, Kelly Brock, Yarin Gal, and Debora S Marks. Disease variant prediction with deep generative models of evolutionary data. *Nature*, 599(7883):91–95, 2021.

[11] Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. Proteingym: Large-scale benchmarks for protein fitness prediction and design. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 64331–64379. Curran Associates, Inc., 2023.

[12] Pascal Notin, Lood Van Niekerk, Aaron W Kollasch, Daniel Ritter, Yarin Gal, and Debora S Marks. Trancepteve: Combining family-specific and family-agnostic models of protein sequences for improved fitness prediction. *bioRxiv*, pages 2022–12, 2022.

[13] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8844–8856. PMLR, 18–24 Jul 2021.

[14] Samantha Petti, Nicholas Bhattacharya, Roshan Rao, Justas Dauparas, Neil Thomas, Juannan Zhou, Alexander M Rush, Peter Koo, and Sergey Ovchinnikov. End-to-end learning of multiple sequence alignments with differentiable smith–waterman. *Bioinformatics*, 39(1):btac724, 2023.

[15] Eli N Weinstein and Debora Marks. A structured observation distribution for generative biological sequence prediction and forecasting. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11068–11079. PMLR, 18–24 Jul 2021.

[16] Nicki Skafte and Søren Hauberg. Explicit disentanglement of appearance and perspective in generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[17] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[18] Oren Freifeld, Søren Hauberg, Kayhan Batmanghelich, and Jonn W Fisher. Transformations based on continuous piecewise-affine velocity fields. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2496–2509, 2017.

[19] Pola Elisabeth Schwöbel. *Learned Data Augmentation for Bias Correction*. PhD thesis, Technical University of Denmark, 2022.

[20] Nicki Skafte Detlefsen, Oren Freifeld, and Søren Hauberg. Deep diffeomorphic transformer networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4403–4412, 2018.

[21] Pola Schwöbel, Frederik Rahbæk Warburg, Martin Jørgensen, Kristoffer Hougaard Madsen, and Søren Hauberg. Probabilistic spatial transformer networks. In *Uncertainty in Artificial Intelligence*, pages 1749–1759. PMLR, 2022.

[22] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000, 2003.

[23] Kay Hofmann and Philipp Bucher. The rsp5-domain is shared by proteins of diverse functions. *FEBS letters*, 358(2):153–157, 1995.

[24] Fabian Sievers, Andreas Wilm, David Dineen, Toby J Gibson, Kevin Karplus, Weizhong Li, Rodrigo Lopez, Hamish McWilliam, Michael Remmert, Johannes Söding, et al. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular systems biology*, 7(1):539, 2011.

[25] Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. *Advances in neural information processing systems*, 29, 2016.

[26] Hannes Stärk, Christian Dallago, Michael Heinzinger, and Burkhard Rost. Light attention predicts protein location from the language of life. *Bioinformatics Advances*, 1(1):vbab035, 2021.

[27] Zemin Zhang and William I Wood. A profile hidden markov model for signal peptides generated by hmmer. *Bioinformatics*, 19(2):307–308, 2003.

[28] Edwin V Bonilla, Kian Chai, and Christopher Williams. Multi-task gaussian process prediction. *Advances in neural information processing systems*, 20, 2007.

[29] Invenia Blog. Gaussian processes: from one to many outputs, 2021.

A Appendix A

A.1 Multi-output Gaussian Processes

A standard Gaussian process (GP) is a stochastic process where any finite set of random variables follows a multivariate normal distribution. This tool is very versatile and can be used for tasks related to non-linear regression. Formally, a GP is defined as:

$$f(\mathbf{x}) \sim \mathcal{GP}\left(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\right)$$

Where $\mathbf{x}$ is the input vector, $f(\mathbf{x})$ is the random variable associated with each input $\mathbf{x}$, $\mathbf{m}\left(\mathbf{x}\right) = \mathbb{E}[f(\mathbf{x})]$ is the mean function, representing the expected value of the process at $\mathbf{x}$ and $k(\mathbf{x}, \mathbf{x}') = \mathbb{E}[(f(\mathbf{x}) - \mathbf{m}\left(\mathbf{x}\right))(f(\mathbf{x}') - \mathbf{m}\left(\mathbf{x}'\right))]$ is the covariance function defined in terms of kernel function, which measures the covariance between the function values at $\mathbf{x}$. Likewise, For any set of inputs $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, as well as the association with their corresponding output values $\{f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)\}$ follow a joint multivariate normal distribution

$$\mathbf{f} = \left(f(\mathbf{x}_1), f(\mathbf{x}_2), \ldots, f(\mathbf{x}_n)\right)^\top \sim \mathcal{N}\left(\mathbf{m}, \mathbf{K}\right)$$

In view of the above, we can say that Multi-Output Gaussian Process (MO-GP) is a generalization of the standard GP to model a number of correlated outputs, with the aim of predicting several outputs at the same time. We can think of MO-GPs as a collection of GPs, each aiming to model a different output. We start from the assumption that each GP uses the same type of kernel function, but with different parameters focused on each task. Following the definition and notation outlined by [28], we can define the MO-GP as follows:

$$\langle f_l(\mathbf{x}), f_k(\mathbf{x}')\rangle = K^f_{lk} k^x(\mathbf{x}, \mathbf{x}') \quad y_{il} = \mathcal{N}\left(f_l(\mathbf{x}_i), \sigma_l^2\right),$$

Here, K^f denotes a positive semidefinite (PSD) matrix representing the similarity between tasks, k^x is the covariance function over inputs, and σ_l^2 is the noise variance for the l^{th} task/channel. Similarly, for inference:

$$\tilde{f}_l(x_*) = (k_l^f \otimes k_*^x)^T \Sigma^{-1} y \qquad \Sigma = K^f \otimes K^x + D \otimes I$$

where $\otimes$ is the Kronecker product, k_l^f is the cofactor that selects the column l^{th} of K^f, k_*^x is the vector of covariances between the test sample x_* and the training points, K^x is the covariance matrix between all pairs of training points, D is a diagonal matrix in which the element $(l,l)^{th}$ is the noise variance on that channel, and Σ denotes the global covariance matrix [28]. The simplest approach to extending MO-GPs is to model each output independently with single-output GPs, i.e. independent sets of GPs. This approach requires that the joint normal distribution over the output vector y is block diagonal with respect to the tasks or channels. Such a configuration prevents observations from one task or channel from influencing predictions for other tasks or channels [28], [29].

A.2 Light Attention

Light Attention is a deep neural network architecture inspired by attention mechanisms in transformer networks. However, unlike traditional attention methods, it captures regions of interest through operations within convolutional networks, specifically using the Hadamard product. This architecture has primarily been employed to enhance representation and feature extraction in protein language models, such as ESM and ProtTrans, functioning as a pooling operator for downstream tasks, and has shown success in applications like subcellular localization prediction [26].

Originally designed as a pooling mechanism for protein language models, the expressive capabilities of Light Attention can be leveraged to address issues related to long-range dependencies, such as flat optimization landscapes when modeling long sequences. In contrast to the original implementation, where both convolutional layers process the same signal (see Figure 6.A), we made a slight modification to the approach by using two input sources instead: one convolutional layer handles the raw input, while the other extracts features from a distribution in the graphical model (see 6.B). This approach simulates what a cross-attention mechanism does on a smaller scale while preserving computational efficiency and speed. The light attention layer is denoted as Ψ_{x,z_A} in the graphical model (refer to section A.2.1 and Figure 2.B), with the methodological scheme illustrated in Figure 6.

A.2.1 ELBO Derivation from the Model to Deal with Long Range Dependencies without Prior

In the main paper, we use an informative prior to guide the generalisation of sequence alignments. However, this raises the question of the implications of not having such a prior and starting the process from scratch. To explore this, we can attempt to derive the expression for ELBO under the following initial conditions:

$$\Psi_{x,z_A} = f\left(x, MC(p(x|z_A))\right), \;\; \tilde{x} \sim p(x|z_A), \;\; \gamma \sim p(x|z_p), \;\; \text{latent variables: } \Psi, z_A, z_p$$

The joint distribution is defined as:

$$p\left(x, \Psi, z_A, z_P\right) = p(x|z_A, z_p)p(z_p|\Psi)p(\Psi|z_A)p(z_A|z_p)$$

The evidence is defined as follows.

$$p(x) = \int_A \int_p \int_\Psi p(x|z_A, z_p)p(z_p|\Psi)p(\Psi|z_A)p(z_A|z_p)dz_A dz_p d_\Psi$$

To derive $\log p(x)$, we get the following:

$$\begin{aligned}
\log(x) &= \log\left(\int_A \int_p \int_\Psi p(x|z_A, z_p)p(z_p|\Psi)p(\Psi|z_A)p(z_A|z_p)dz_A dz_p d_\Psi\right) \\
&= \log\left(\int_A \int_p \int_\Psi p(x|z_A, z_p)p(z_p|\Psi)p(\Psi|z_A)p(z_A|z_p)\frac{q_A(z_A|x,z_p)}{q_A(z_A|x,z_p)}\frac{q_p(z_p|\Psi)}{q_p(z_p|\Psi)}\frac{q_\Psi(\Psi|x,z_A)}{q_\Psi(\Psi|x,z_A)}dz_A dz_p d_\Psi\right) \\
&= \log\left(\int_A \int_\Psi \underbrace{p(\Psi|z_A)\frac{q_\Psi(\Psi|x,z_A)}{q_\Psi(\Psi|x,z_A)}\left\{\int_p p(z_p|\Psi)p(x|z_A,z_p)p(z_A|z_p)\frac{q_A(z_A|x,z_p)}{q_A(z_A|x,z_p)}\frac{q_p(z_p|\Psi)}{q_p(z_p|\Psi)}dz_p\right\}d_\Psi}_{\textbf{Not possible a proper factorization}} dz_A\right)
\end{aligned}$$

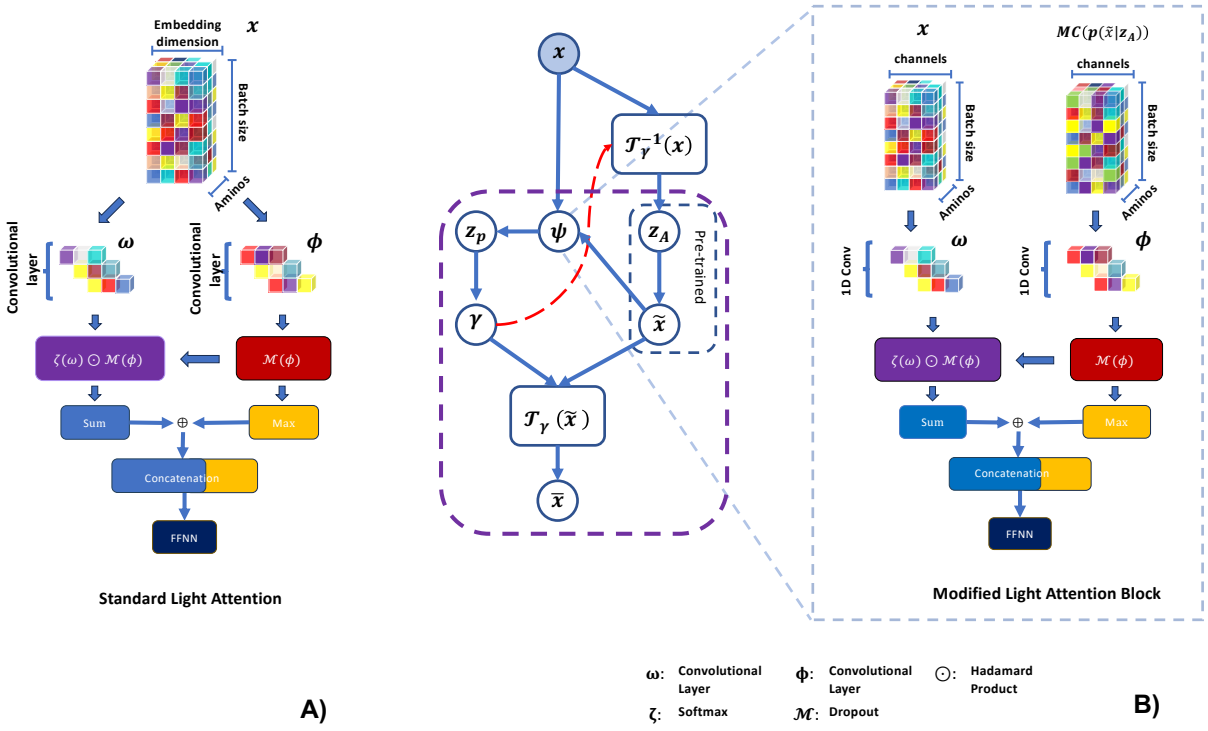

Figure 6: Light Attention building blocks: A illustrates the structure and functional blocks of light attention. First, given an input or embedding x, it passes through the convolutional layers ω and Φ to extract their respective features. The features obtained from Φ undergo a dropout, followed by a pointwise multiplication of the softmax of the features from ω and the resulting dropout from Φ. Additionally, pooling is applied to both the pointwise multiplication and the dropout of Φ. Finally, the features are concatenated and passed to a multilayer perceptron (MLP) to obtain the final features. **B** illustrates the integration of light attention as a latent variable, modelled as a normal distribution. Unlike the original light attention framework, this modification uses two inputs, each connected to different convolutional layers: x is connected to ω, while the Monte Carlo sample from the prior, defined as the pre-trained model of z_A, is connected to $\tilde{x}$ and then to Φ. The process then follows the same procedure as in the original implementation.

A detailed examination reveals that the inclusion of Ψ_{x,z_A} in the graphical model introduces a significant trade-off: the presence of a cycle within what was originally a Directed Acyclic Graph (DAG). This modification does not allow the derivation of a closed form expression for the ELBO which make it non-tractable. This limitation can be potentially be overcome using Markov Chain Monte Carlo (MCMC) techniques such as Gibbs sampling or Hamiltonian Monte Carlo, but we leave such considerations for future work.

Chapter 4

Cross-species vs species-specific models for protein melting temperature prediction

The work presented in this chapter has been submitted to BiorXiv, and is under screening on the platform before finally being made available to the public.

CROSS-SPECIES VS SPECIES-SPECIFIC MODELS FOR PROTEIN MELTING TEMPERATURE PREDICTION

A PREPRINT

Sebastián García López
Department of Computer Science - DIKU
University of Copenhagen
Sebastian.Garcia.Lopezs@di.ku.dk

Jesper Salomon
Novonesis
JRSX@novonesis.com

Wouter Boomsma
Department of Computer Science - DIKU
University of Copenhagen
wb@di.ku.dk

October 10, 2024

ABSTRACT

Protein melting temperatures are important proxies for stability, and frequently probed in protein engineering campaigns, including enzyme discovery, protein optimization, and de novo protein design. With the emergence of large datasets of melting temperatures for diverse natural proteins, it has become possible to train models to predict this quantity, and the literature has reported impressive performance values in terms of Spearman rho. The high correlation scores suggest that it should be possible to reliably predict melting temperature changes in engineered variants, to design de novo thermostable proteins, and identifying naturally thermostable proteins. However, in practice, results in this setting are often disappointing. In this paper, we explore the discrepancy between these two settings. We show that Spearman rho over cross-species data gives an overly optimistic impression of prediction performance, and that models trained on species-specific data often outperform larger cross-species models. Finally, we explore a number of strategies for improving performance, demonstrating a performance boost of 1.5 degree RMSE with fairly simple means.

Introduction

Reliable prediction of protein thermal stability is a long-standing challenge in protein engineering, particularly for enzyme optimization, where it is a strong indicator for functional integrity under thermal stress. Special focus has been on the prediction of the *changes* in stability induced by mutations, also referred to as $\Delta\Delta G$. Over the last decades, a long list of algorithms have been developed for this purpose, ranging from early models such as Delgado et al. [2019] and Rosetta Sora et al. [2023], to deep learning approaches based on convolutional and graph neural networks (Boomsma and Frellsen [2017], Li et al. [2020], Blaabjerg et al. [2023]), and most recently approaches building on large pre-trained models Dieckhaus et al. [2023].

Rather than considering *differences* in thermostability, other work has focused on predicting absolute stability Cagiada et al. [2024]. In this setting, the melting temperature, T_m of a protein is a useful proxy. It has the advantage that it can be measured reproducibly in high-throughput assays. Additionally, unlike other measures of protein stability or those that assess stability under specific conditions, the melting temperature is directly comparable across proteins, making it a valuable general optimization target, frequently correlating with stability to complex stress factors Sanchez-Ruiz [2010].

A recent large-scale dataset of melting temperature of 48,000 proteins across 13 species Jarzab et al. [2020], combined with standardized splits Dallago et al. [2021], has made the T_m prediction problem accessible to the machine learning community. Several recent studies have reported impressive prediction performance, with Spearman correlation coefficients above 0.7 Dallago et al. [2021], Su et al. [2023], Sułek et al. [2024], Rodella et al. [2024], Pudžiuvelytė et al. [2024], Li et al. [2023], Yang et al. [2022].

Although progress in this area are highly welcome, the high performance values reported in the literature are somewhat at odds with our expectation about the difficulty of this prediction task. Generally, absolute stability prediction has been

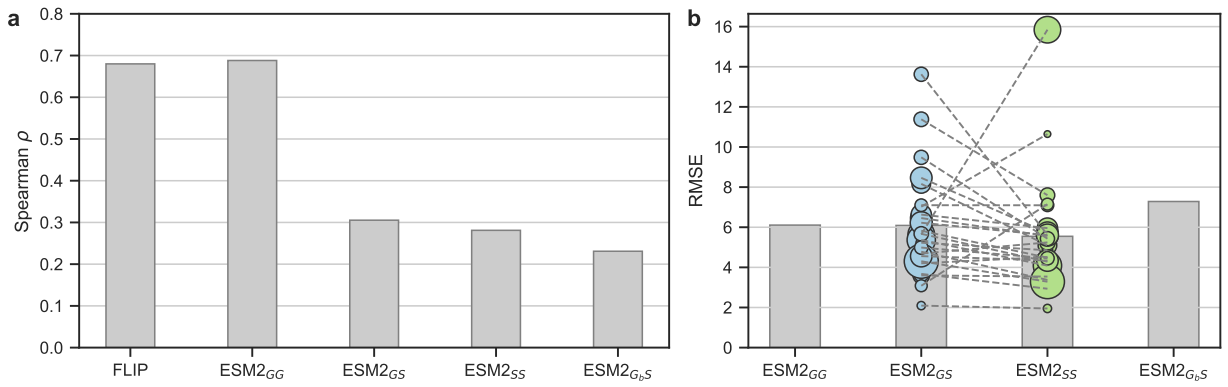

Figure 1: Performance of sequence based embeddings to predict melting temperature across species. a) Performance measured in terms of Spearman rank correlation, b) Performance measured in terms of RMSE, showing the individual performances per species as circles, scaled by the size of the dataset. The G and S subscripts denote *global*, and *specifies-specific*, respectively, with the first denoting the training scenario and the second denoting the testing scenario. For instance *GS* denotes a model trained on all data, while being evaluated on each species. G_B denotes a model that where the dataset was balanced by species during training (see Method). Note how much of the correlation arises simply from the fact that the correlation is measured across all species.

considered to be more challenging that relative stability prediction. In addition, it conflicts with our own prior experience with melting temperature prediction in the protein engineering setting, where we have typically seen substantially lower performances than those reported Notin et al. [2023]. In this paper, we investigate the origin of this apparent discrepancy as a stepping stone towards a better understanding of the status of prediction performance in this domain. We start by identifying the basis for the performance discrepancy – and show that much of the reported Spearman correlation arises as the consequence of the difference in melting temperature between species, rather than an ability to predict melting temperatures for individual variants. We thus confirm that species-specific melting temperature remains a challenging problem, with current methods displaying RMS errors of about 6 degrees, which are substantial, given the inner-species variance. Next, we ask the question whether greater performance can be obtained by training species-specific models, rather than a single large cross-species model. We then pursue three potential strategies for increasing performance further, using 1) a richer input embedding, 2) a representation obtained from a constrastive training objective, and 3) a loss that explicitly anchors the results to the optimal growth temperature of the species. Ultimately, our results show that the species-specific models maintain their edge over the cross-species models in these different settings, and that prediction errors can be reduced by about 1.5 degree RMSE with fairly simple means.

Results

We start by establishing a simple model architecture as a baseline for our subsequent analyses. The model takes ESM-2 embeddings Lin et al. [2022] as input, and produces melting temperature values as output. For simplicity, we use a classic transfer-learning setup with frozen embeddings (no fine-tuning). To aggregate the per-position embeddings into a single output we use a light attention layer, as this is known to outperform simple averaging over the length of the protein Stärk et al. [2021], Detlefsen et al. [2022]. Running on the standard splits of the Meltome dataset, our baseline method marginally outperform the results reported in the original FLIP paper (Figure 1a, two left-most bars), demonstrating that our baseline implementation serves as a reasonable representative of the current state of the art. See *Methods* for details.

Discrepancy explained: correlation is a poor metric

The third bar in Figure 1a (ESM2_{GS}) is equivalent to the second, but evaluates the Spearman correlation for each specifies individually and reports the average. The dramatic difference of this result compared to the two first bars explains the discrepancy discussed above. Clearly, much of the observed correlation found in the first two bars is due simply to global melting temperature differences between species, rather than the ability to determine melting temperature differences for different proteins within a species. This suggests that basing conclusions solely on the correlation of predictions with the ground truth may lead to misleading conclusions, and highlights the need to include

additional metrics in the assessment of model performance. To illustrate, if we instead measure our performance in terms of root mean square error (RMSE), we find no notable discrepancy between the cross-species and per-species assessment of our model (1b). In the remainder of this paper, we will therefore rely on RMSE for our conclusions.

Species-specific models potentially outperform cross-species models

We have established that evaluating performance per-species aligns more closely with the objective of interest, namely assessing the impact of variation within species. A natural question is whether *training* separate models on each species is also beneficial. Although cross-species models are convenient, there are situations where species-specific models would be equally useful. One important use-case is for the surrogate function in a Bayesian optimization framework. In this setting, we typically have observed data for the specific system, and wish to generalize to unseen variants, potentially updating the model when new experimental data is observed. The training of cross-species vs species-specific models represents a trade-off. On one hand, one might expect that cross-species models could benefit from larger datasets and could provide greater generalization capabilities; on the other hand, a specialized species-specific model might obtain a better fit with fewer parameters.

Assessing this question empirically on the FLIP dataset, we see that training separate models on each species (ESM2$_{SS}$, 1a, fourth column) provides comparable performance to the global model, suggesting that training larger cross-species models is not beneficial in this setting, perhaps even showing slight detrimental effects in terms of RMSE. We note that the lack of performance of the global model is not related to the imbalance of data set sizes between species. In fact, rebalancing the databaset during training leads to worse results (ESM2$_{G_bS}$). Instead, the problem is likely related to the same behavior that led to the exaggerated performance when measured with Spearman correlation: a Simpson-paradox like issue, where the model focuses on the global trends and disregard the local intra-species effects.

When tracking the performance differences on the individual data sets (Figure 1b, middle two columns), we see that there is actually a general trend towards better performance when training species-specific models, but that these models show greater variance across species (see also Appendix A, C, D). Interestingly, while the outliers for the global model are typically the smaller datasets as one would expect, the species-specific models sometimes fail even on species with abundant data. This suggests that there is potential for performance improvement in the species-specific models if we can find ways to mitigate the variance issues. In the following, we will pursue three different strategies towards this goal.

Strategy 1: Richer featurizations

We hypothesize that one way to reduce variance and improve the reliability of predictions is to enrich the input featurization by incorporating information derived from protein structures. For this reason, in addition to using sequence-based embeddings (ESM2), we included embeddings generated by the PiFold inverse folding model (Figure 2C and *Methods*). The idea is that concatenation of sequence and protein structure embeddings provides a more expressive representation, potentially allowing for improved prediction.

Figure 4a demonstrates the performance for global and species-specific models using the two input embeddings, for clarity allocating the y-axis for the individual species. The combined ESM2+Pifold embedding is indeed observed to reduce the variance of the species specific models, avoiding the critical outliers observed for the ESM2 embedding and generally displaying improved performance across species. Interestingly, the global model produces *more* outliers with the combined embedding, although this effect seems to be limited to the species for which we have very little training data.

Strategy 2: Contrastive learning

The results above suggest that the global model focuses on the differences in optimal growth temperature (OGT) between species, rather than the effect of individual variants. We therefore considered whether we could choose alternative training objectives that better reflected our desired outcomes.

One intuitively appealing approach is *contrastive* learning, where a model is trained to enforce pairwise relationships between samples, for instance ensuring that proximal neighbors belong to the same class. Often, such methods will use data augmentation strategies to produce local variations that are considered proximal, and can then be contrasted with all other points in terms of the predicted output values. In the field of image processing, data augmentation often consists of various transformation techniques, such as cropping, resizing, Gaussian noise addition and rotations Chen et al. [2020]. The effectiveness of this approach depends on the expressiveness of such transformations. Overly conservative, trivial augmentation might result in lack of generalization, while overly broad augmentation produces false negative samples Chuang et al. [2020],Khosla et al. [2020].

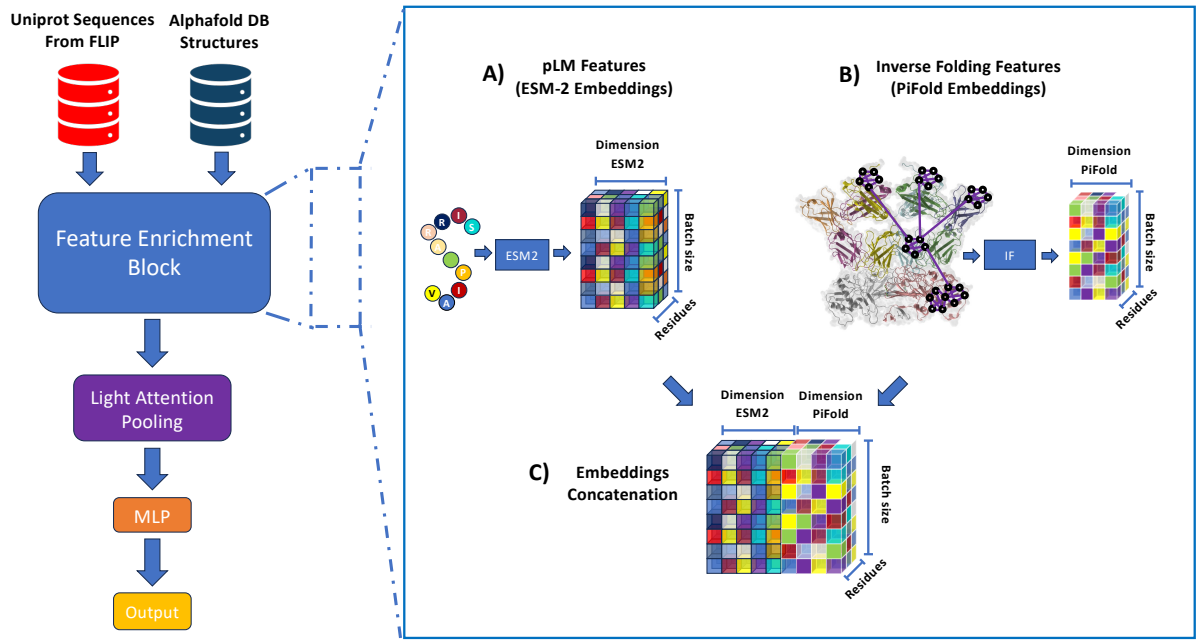

Figure 2: The methodological approach in this study utilizes embeddings derived from protein language models, specifically ESM2 and PiFold, or a combination thereof (see Figure 2). Subsequently, a Light Attention (LA) block is implemented to perform appropriate pooling of these embeddings. Given the variability in sequence length among proteins in the dataset across organisms, the LA block is particularly suitable. Unlike conventional pooling methods, the LA block processes embeddings by channel using convolutional neural networks (CNNs), rather than based on embedding length. This block performs learned pooling, capturing features extracted by the embedding and generating a condensed representation of the initial set of embeddings for each protein. This condensed representation then serves as input features for a multilayer perceptron (MLP) regressor to predict melting temperatures (t_m).

For the purpose of T_m prediction, there are two challenges in applying contrastive learning: 1) there is no trivial data augmentation procedure available, 2) we are in a regression-setting rather than a classification setting. A recent approach, rank-N-contrast Zha et al. [2024], presents a solution to both problems, by using a contrastive learning objective to learn a representation of the inputs, which can subsequently be used as input for a downstream regression task. This method was shown to be insensitive to the choice of data augmentation (it can be applied even without it), and the two-step procedure makes is amenable for regression. To learn the representation, the rank-N-contrast method contrasts samples based on their ranking within the target space of continuous labels. For example, if two samples in a batch during training have the highest similarity in terms of melting temperature, they will be enforced to have the shortest distance in representation space. Given the beneficial role of data augmentation in the original paper, we add a simple data augmentation step in the form of a small Gaussian perturbation of the embedding input vector. Once the representations are learned, they are used directly as input to the multilayer perceptron regressor (MLP), replacing the feature enrichment and light attention blocks in Figure 2. See *Method* for details.

The results for the contrastive learing experiments are report in Figure 4b. While the contrastive method produces reasonable results, they are clearly inferior to our initial MSE strategy.

Strategy 3: Anchoring to optimal growth temperature

A simpler strategy towards an improved focus on in-species differences is to use centroids to condition learning over each species or mode in the distributions. Each centroid represents the average melting temperature (μ_{t_m}) of the proteins belonging to a particular species. The purpose of these centroids is to prevent the represented proteins from deviating from their corresponding distributions. In practical terms, the average t_m of each species would serve as the Optimal Growth Temperature (OGT) for the proteins of that species. In the context of thermostability, optimal growth temperature (OGT) is an indirect measure of the ability of proteins and biomolecules to maintain their essential functions and structural integrity (i.e. remain stable) in high temperature environments. This measure is reflected in the behaviour of organisms (micro-organisms) when exposed to elevated temperatures just prior to denaturation.

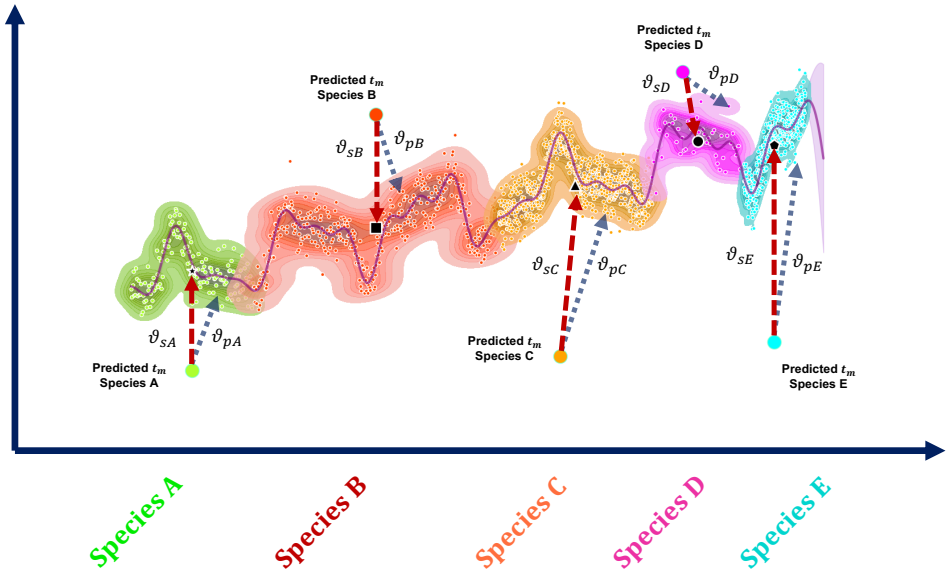

Figure 3: Mechanism of action of the dual loss function. In the graph, the densities represent the space of continuous labels corresponding to the t_m of each protein within the training set. Each density is indicated by a specific colour, which indicates the species to which the points within that density belong. The purple line through all densities depicts the trend of t_m over the continuous label space. Points outside the densities are the t_m predicted by the model and are indicated by colours indicating their membership of a particular species, where the predicted t_m is a linear combination of the predicted OGT and its bias (see equation 1). The dashed blue arrows represent the MSE between the predicted sample and its ground truth t_m, while the red arrows correspond to the MSE between the predicted OGT and the centroid in the respective density, which acts as the estimated OGT or true value for practical purposes. In addition, it is important to consider practical issues, e.g. that μ_{t_m} is only needed during the training stage, so that this information is not needed during inference. Furthermore, the calculation of μ_{t_m} is done beforehand by averaging the t_m per species in the training and validation sets, and is then used during the training process. Therefore, during the induction process, it is necessary to have information about the species to which the protein belongs and its associated t_m.

Figure 3 illustrates the operation of the loss function. This mechanism is based on conditioning the t_m predictions of each protein to values within the ranges determined by the Optimal Growth Temperature (OGT) of each species. This is achieved by contrasting the predictions with the centroids calculated previously by averaging the t_m per species used as estimated OGTs. This procedure can be considered as a regularisation mechanism during optimisation, as it attempts to avoid predictions that deviate significantly from the expected values according to the OGTs. The loss function is defined as:

$$\mathcal{L}_{\text{dual}} = \vartheta_p(\varphi_{t_m}, y_{t_m}) + \vartheta_s(\tilde{y}_{OGT}, \mu_{t_m}) \quad \text{where} \quad \varphi_{t_m} = \tilde{y}_{OGT} + \tilde{y}_{bias} \tag{1}$$

Here, φ_{t_m} the predicted t_m output from the model (see Figure 2), y_{t_m} denotes the ground truth melting temperature, $\tilde{y}_{OGT}$ is the predicted OGT for each sample, μ_{t_m} is the average t_m per species, ϑ_p is the mean square error (MSE) between the predicted φ_{t_m} and the ground truth t_m, and ϑ_s is the MSE between the predicted OGT and the average t_m per species. A visual illustration of the procedure is presented in figure 3. During training, values for the average and the deviation are specified separately, but this training scheme maintains the advantage that no information about OGT (or average melting temperature) is necessary at test time.

Our experiments on the FLIP dataset are reported in Figure 4c. The dual-loss approach provides some improvement for the global model for the structurally enriched ESM+PiFold input representation, but it does not fully close the gap between the global and species-specific models, with the latter still displaying a slight edge. Perhaps surprisingly at first sight, we note that the species-specific model also benefits from the dual-loss strategy, although in this case, the average loss is simply a constant value for all data points in the training set. The explanation for this phenomenon is found in the fact that the dual loss implicitly standardizes the output values (from global melting temperatures to deviations from an average). The dual loss strategy can thus be viewed as a meaningful standardization technique that can be applied in the global setting, where standard standardization techniques would fail (since a global mean is not a meaningful anchor), while corresponding to a standard substraction of the mean in the single-species case.

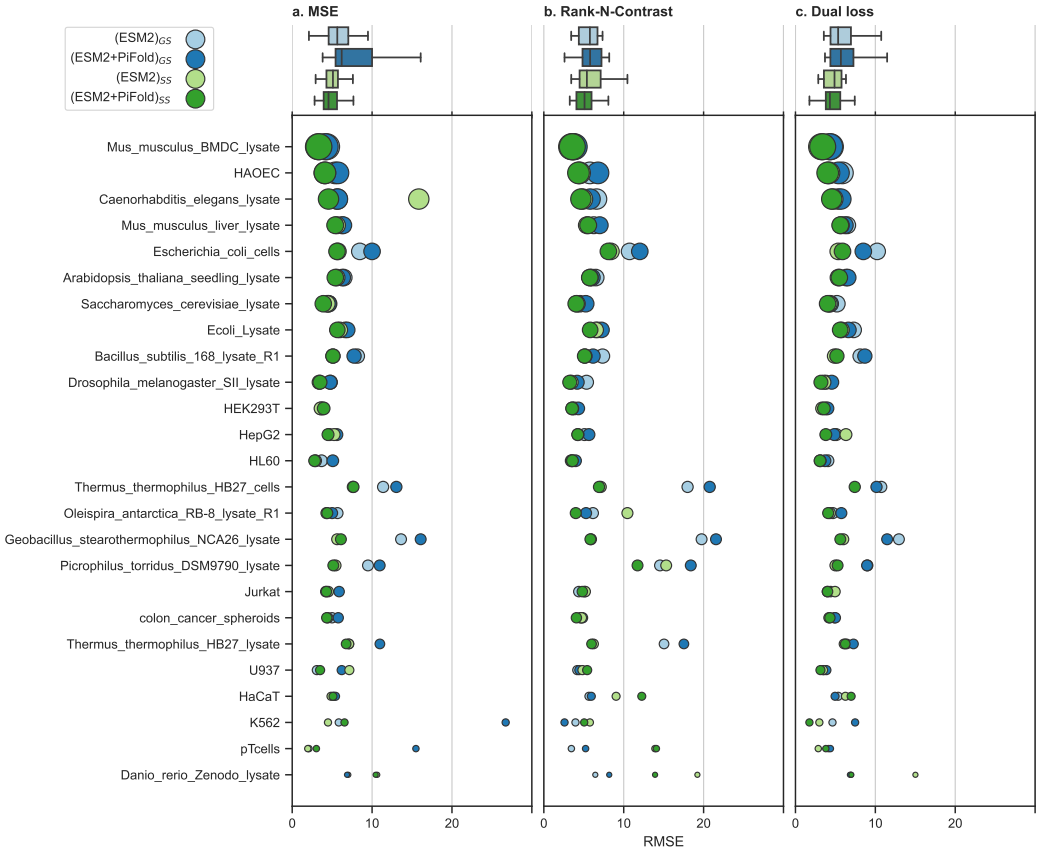

Figure 4: Results on the FLIP Meltome dataset for the three strategies discussed in the main text. For each strategy, the figure contrasts the performance of sequence-based embeddings (light colors) vs. the combination of sequence-based embeddings and protein structurederived embeddings (full colors) in terms of RMSE.

	Global		**Species-specific**	
	ESM_{gs}	**$(ESM + PiFold)_{gs}$**	**ESM_{ss}**	**$(ESM + PiFold)_{ss}$**
MSE	5.90	6.34	5.99	4.59
Rank-N-contrast	6.78	7.21	5.63	5.08
Dual loss	5.95	5.86	7.21	**4.53**

Table 1: Overview of results in terms of an average over RMSE for the individual datasets, weighted by test-set size. **ss** denotes individual models per species, while **gs** represents global models applied to each species. The methods compared include traditional approaches based on loss functions such as Mean Squared Error (MSE) or sample balancing per batch using the same loss function, as compared to contrastive methods such as Rank N Contrast Loss (Rank N) and the dual loss (biasg) proposed in this study.

Discussion

A summary of the results can be found in Table 1. In short, we find that species-specific models generally outperform globally trained models. Species-specific models suffer from larger variance on small datasets, but it appears that this effect can be mitigated by expanding the input embedding to include structural information, presumably serving as a regularizer to guard against overfitting. Somewhat unexpectedly, the inclusion of structural information was not beneficial in the global training scheme.

The Rank-N-contrast and Dual loss approaches can both be viewed as examples of contrastive learning methods. The former learns a representation by contrasting inputs within a small perturbation region with the rest, while the latter is a simple strategy where predictions are anchored to the species mean. In our hands, the rank-N-contrast method did not provide any useful performance gains, but this could be due to the choice of perturbation mechanism used. When combined with the enriched featurization, the dual loss ultimately produces the best results, but they differ only marginally from the results obtained with MSE on the (ESM + PiFold) input embedding. As discussed earlier, we stress that when employed in the species-specific setting, the dual loss merely corresponds to a mean standardization.

The overall conclusion is that the performance of thermal stability prediction can be improved by almost 1.5 degrees RMSE using fairly simple techniques. In particular, the choice of training species-specific models rather than global models had an important effect, but only when also combined with a enriched input featurization that incorporates structural information.

Methods

Embedding types Our model employs two different embedding types: a sequence-based embedding from the ESM2 (Evolutionary Scale Modeling) model Lin et al. [2022] and an embedding obtained from the inverse folding model PiFold Hsu et al. [2022]. The former is an unsupervised language model based on the transformer architecture, trained on 250 million sequences from the UniRef dataset, which has been shown to capture and learn relevant information such as biochemical properties of amino acids, residue contact mapping, and homology detection, making it well-suited for various downstream tasks Rives et al. [2021], Lin et al. [2022], Michael et al. [2024]. As an inverse folding method, PiFold aims to predict protein sequences based on the atomic coordinates of protein backbone structures, and thus more directly captures the structural environment in its embeddings.

Model design The fundamental structure of the model is illustrated in Figure 2. The sequence and structural embeddings produce per-residue vector representations of 1280 and 128 dimensions respectively. These were either used independently or concatenated per residue, and passed through a light attention block (see Figure 2). As the name suggests, light attention is a light-weight mechanism to approximate a traditional attention mechanism, and is commonly used as an alternative to simpler aggregation options such as calculating a mean over the sequence length Stärk et al. [2021]. This approach allows a unified and efficient representation of protein features, regardless of sequence length. While the Feature Enrichment Block performs the embeddings pre-processing during the induction/inference phase of the model, the Light Attention Block acts as an encoder, learning a compact representation of the features in the embedding space. This condensed representation is then fed into a single dense neural network which performs regression to predict melting temperature values.

The light attention block was implemented using the default parameters as described in Stärk et al. [2021], although dimensionality of the input embeddings was adjusted based on the type of embedding used. For prediction, a multilayer perceptron (MLP) was designed, comprising three hidden layers with an architecture of (128, 64, n), where n represents the number of output components, which varies depending on the specific experiment. This configuration allows for model adaptability to different predictive tasks while maintaining a consistent base structure.

Rank-N-Contrast The Rank-N-Contrast Zha et al. [2024] procedure consists of 3 components:

Data augmentation: A batch of input-label pairs undergoes data augmentation in the original embedding space using a Gaussian perturbation. While the original description of the method proposed constructing a two-view batch consisting of two types of augmentations, we deviate slightly from this by creating a two-view batch which comprises the original input and a transformed version of the same input. In our case, the transformation was implemented as a Gaussian perturbation in the embedding space.

Rank-N-Contrast Loss: A regression-aware loss function is constructed to learn a representation of the input data such that relative distances reflect differences in their continuous target values (i.e. melting temperatures). Following the original publication, the $\mathrm{R}_N\mathrm{C}$ loss per sample is defined as:

$$l_{\text{RNC}}^{(i)} = \frac{1}{2N-1} \sum_{j=1, j\neq i}^{2N} -\log \mathbb{P}(v_j|v_i, \mathcal{S}_{i,j}) \qquad \text{where} \qquad \mathbb{P}(v_j|v_i, \mathcal{S}_{i,j}) = \frac{\exp(\text{sim}(v_i, v_j)/\tau)}{\sum_{v_k \in \mathcal{S}_{i,j}} \exp(\text{sim}(v_i, v_k)/\tau)}$$

Here, $\text{sim}(\cdot, \cdot)$ is a similarity in representation-space between samples v_i and v_j, and τ is a temperature parameter. $\mathbb{P}(v_j|v_i, \mathcal{S}_{i,j})$ represents the likelihood of a sample v_j in the context of anchor v_i, and the set of samples $\mathcal{S}_{i,j}$) that relative to v_i have a worse rank than v_j in terms of the *output* values.

$$\mathcal{S}_{i,j} := \{v_k \mid k \neq i, d(\tilde{y}_i, \tilde{y}_k) \geq d(\tilde{y}_i, \tilde{y}_j)\}$$

As an example, if v_j is the sample that is closest to v_i in terms of melting temperature, the likelihood would be optimized when v_j had the highest similarity to v_i in representation space. The full likelihood is simply an average over all samples in the batch.

$$\mathcal{L}_{\text{RNC}} = \frac{1}{2N} \sum_{i=1}^{2N} l_{\text{RNC}}^{(i)}$$

Representation Learning: The model learns an encoder by optimizing $\mathcal{L}_{\text{RNC}}$. Once the representation is learned, the encoder is frozen, and a separate predictor is trained using a standard loss. In our case, the learned Rank-N-Contrast representations are used as direct input to the light attention block in figure 2.

Data All analyses were done on the Meltome partition of the FLIP dataset Dallago et al. [2021]. The dataset comprises 15 partitions designed to address various tasks in the field of protein design. For this study, we specifically focused on the partition related to thermostability, based on proteins with their corresponding melting temperatures (t_m) derived from the Meltome Atlas database. The 'Mix' partition was selected, characterized by a high diversity of proteins originating from multiple species. Proteins less than 50 residues in length were discarded for this work. The amount of proteins corresponding to each species in the splits of the FLIP dataset is presented in the Table 2

	Sample Size		
Species	**Train**	**Val**	**Test**
Mus musculus BMDC lysate	3108	369	227
HAOEC	2157	242	488
Caenorhabditis elegans lysate	1900	195	350
Mus musculus liver lysate	1308	139	71
Escherichia coli cells	1254	133	49
Arabidopsis thaliana seedling lysate	1252	139	203
Saccharomyces cerevisiae lysate	1245	123	226
Ecoli Lysate	1108	137	212
Bacillus subtilis 168 lysate R1	931	91	158
Drosophila melanogaster SII lysate	895	103	152
HEK293T	735	76	123
HepG2	647	71	85
HL60	628	79	82
Thermus thermophilus HB27 cells	573	59	58
Oleispira antarctica RB 8 lysate R1	555	77	125
Geobacillus stearothermophilus NCA26 lysate	541	68	36
Picrophilus torridus DSM9790 lysate	539	53	138
Jurkat	499	53	49
colon cancer spheroids	486	50	74
Thermus thermophilus HB27 lysate	426	47	105
U937	380	29	16
HaCaT	293	42	33
K562	237	28	2
pTcells	185	24	3
Danio rerio Zenodo lysate	119	11	11

Table 2: Sample distribution table by species. This table was constructed based on the distribution provided by the flip partition in relation to the Meltome Atlas, using the crossspecies partition or "mix".

Protein sequences from FLIP split were used to compute embeddings using ESM2, which were then stored in files for efficient loading during model induction/inference. To map structural features, the UniProtIDs of each protein were used as a query in AlphaFoldDB Varadi et al. [2022] to obtain predicted structures. This approach was adopted to ensure the use of the closest predicted protein structure for each sequence, while avoiding the complexity associated with multi-chain protein structures. Once the protein structures were obtained, PiFold was used to generate graph embeddings, which were then stored in files. Notably, only the encoder component of the algorithm was used, with the decoder omitted for practical reasons.

Training The training approach follows the principle of classic transfer learning, where the embeddings were pre-generated and not fine-tuned on the thermostability task. This strategy was chosen for its versatility and scalability to larger datasets, in addition to the fact that fine-tuning does not always improve performance and can be counterproductive Dieckhaus et al. [2023].

AdamW was used as the main optimiser. The learning rate (lr) was adjusted according to the type of experiment: for the calibration of models using the full FLIP training partition in the global model (all species in the dataset), $\text{lr} = 1 \times 10^{-4}$ was set, while for the training of species-specific models (individual models trained with proteins belonging to the same species), $\text{lr} = 1 \times 10^{-3}$ was used.

Due to its particular architecture, the Rank N Contrast model followed a different procedure.. This method requires a previously trained encoder to generate the representation (in this case, the light attention module) and a predictor (the multilayer dense neural network). The two components were optimised with different learning rates, denoted as lr_e (encoder) and lr_p (decoder). For global model calibration, $\text{lr}_e = 1 \times 10^{-5}$ and $\text{lr}_p = 1 \times 10^{-4}$, while for species-specific models, $\text{lr}_e = 1 \times 10^{-5}$ and $\text{lr}_p = 1 \times 10^{-3}$.

Acknowledgement

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 801199, the Novo Nordisk Foundation through the MLSS Center (Basic Machine Learning Research in Life Science, NNF20OC0062606), and the Pioneer Centre for AI (DNRF grant number P1). The project was carried out in collaboration with the Enzyme Research Division of Novonesis A/S, located in Kongens Lyngby, Denmark.

References

Javier Delgado, Leandro G Radusky, Damiano Cianferoni, and Luis Serrano. FoldX 5.0: working with RNA, small molecules and a new graphical interface. *Bioinformatics*, 35(20):4168–4169, 03 2019. ISSN 1367-4803. doi:10.1093/bioinformatics/btz184. URL https://doi.org/10.1093/bioinformatics/btz184.

Valentina Sora, Adrian Otamendi Laspiur, Kristine Degn, Matteo Arnaudi, Mattia Utichi, Ludovica Beltrame, Dayana De Menezes, Matteo Orlandi, Ulrik Kristoffer Stoltze, Olga Rigina, Peter Wad Sackett, Karin Wadt, Kjeld Schmiegelow, Matteo Tiberti, and Elena Papaleo. Rosettaddgprediction for high-throughput mutational scans: From stability to binding. *Protein Science*, 32(1):e4527, 2023. doi:https://doi.org/10.1002/pro.4527. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/pro.4527.

Wouter Boomsma and Jes Frellsen. Spherical convolutions and their application in molecular modelling. *Advances in neural information processing systems*, 30, 2017.

Bian Li, Yucheng T Yang, John A Capra, and Mark B Gerstein. Predicting changes in protein thermostability upon point mutation with deep 3d convolutional neural networks. *bioRxiv*, 2020.

Lasse M Blaabjerg, Maher M Kassem, Lydia L Good, Nicolas Jonsson, Matteo Cagiada, Kristoffer E Johansson, Wouter Boomsma, Amelie Stein, and Kresten Lindorff-Larsen. Rapid protein stability prediction using deep learning representations. *Elife*, 12:e82593, 2023.

Henry Dieckhaus, Michael Brocidiacono, Nicholas Randolph, and Brian Kuhlman. Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *bioRxiv*, 2023.

Matteo Cagiada, Sergey Ovchinnikov, and Kresten Lindorff-Larsen. Predicting absolute protein folding stability using generative models. *bioRxiv*, pages 2024–03, 2024.

Jose M Sanchez-Ruiz. Protein kinetic stability. *Biophysical chemistry*, 148(1-3):1–15, 2010.

Anna Jarzab, Nils Kurzawa, Thomas Hopf, Matthias Moerch, Jana Zecha, Niels Leijten, Yangyang Bian, Eva Musiol, Melanie Maschberger, Gabriele Stoehr, et al. Meltome atlas—thermal proteome stability across the tree of life. *Nature methods*, 17(5):495–503, 2020.

Christian Dallago, Jody Mou, Kadina E Johnston, Bruce J Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pages 2021–11, 2021.

Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pages 2023–10, 2023.

Adam Sułek, Jakub Jończyk, Patryk Orzechowski, Ahmed Abdeen Hamed, and Marek Wodziński. Esmtemp - transfer learning approach for predicting protein thermostability. In *Computational Science – ICCS 2024: 24th International Conference, Malaga, Spain, July 2–4, 2024, Proceedings, Part III*, page 187–194, Berlin, Heidelberg, 2024. Springer-Verlag. ISBN 978-3-031-63758-2. doi:10.1007/978-3-031-63759-9_23. URL https://doi.org/10.1007/978-3-031-63759-9_23.

Chiara Rodella, Symela Lazaridi, and Thomas Lemmin. TemBERTure: advancing protein thermostability prediction with deep learning and attention mechanisms. *Bioinformatics Advances*, 4(1):vbae103, 07 2024. ISSN 2635-0041. doi:10.1093/bioadv/vbae103. URL https://doi.org/10.1093/bioadv/vbae103.

Ieva Pudžiuvelytė, Kliment Olechnovič, Egle Godliauskaite, Kristupas Sermokas, Tomas Urbaitis, Giedrius Gasiunas, and Darius Kazlauskas. TemStaPro: protein thermostability prediction using sequence representations from protein language models. *Bioinformatics*, 40(4):btae157, April 2024. ISSN 1367-4811. doi:10.1093/bioinformatics/btae157. URL https://doi.org/10.1093/bioinformatics/btae157.

Mengyu Li, Hongzhao Wang, Zhenwu Yang, Longgui Zhang, and Yushan Zhu. Deeptm: A deep learning algorithm for prediction of melting temperature of thermophilic proteins directly from sequences. *Computational and Structural Biotechnology Journal*, 21:5544–5560, 2023. ISSN 2001-0370. doi:https://doi.org/10.1016/j.csbj.2023.11.006. URL https://www.sciencedirect.com/science/article/pii/S2001037023004221.

Yang Yang, Jianjun Zhao, Lianjie Zeng, and Mauno Vihinen. Protstab2 for prediction of protein thermal stabilities. *International Journal of Molecular Sciences*, 23(18), September 2022. ISSN 1661-6596. doi:10.3390/ijms231810798.

Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. Proteingym: Large-scale benchmarks for protein fitness prediction and design. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 64331–64379. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/cac723e5ff29f65e3fcbb0739ae91bee-Paper-Datasets_and_Benchmarks.pdf.

Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *BioRxiv*, 2022:500902, 2022.

Hannes Stärk, Christian Dallago, Michael Heinzinger, and Burkhard Rost. Light attention predicts protein location from the language of life. *Bioinformatics Advances*, 1(1):vbab035, 2021.

Nicki Skafte Detlefsen, Søren Hauberg, and Wouter Boomsma. Learning meaningful representations of protein sequences. *Nature communications*, 13(1):1914, 2022.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33: 18661–18673, 2020.

Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: learning continuous representations for regression. *Advances in Neural Information Processing Systems*, 36, 2024.

Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International Conference on Machine Learning*, pages 8946–8970. PMLR, 2022.

Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. 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.

Richard Michael, Jacob Kæstel-Hansen, Peter Mørch Groth, Simon Bartels, Jesper Salomon, Pengfei Tian, Nikos S Hatzakis, and Wouter Boomsma. A systematic analysis of regression models for protein engineering. *PLOS Computational Biology*, 20(5):e1012061, 2024.

Mihaly Varadi, Stephen Anyango, Mandar Deshpande, Sreenath Nair, Cindy Natassia, Galabina Yordanova, David Yuan, Oana Stroe, Gemma Wood, Agata Laydon, et al. Alphafold protein structure database: massively expanding the structural coverage of protein-sequence space with high-accuracy models. *Nucleic acids research*, 50(D1): D439–D444, 2022.

Suplementary Material

The supplementary material presented below includes supporting results for all experiments conducted in this study. This material includes violin plots and box plots for all methods and performance measures analysed in the paper, as well as tables with the corresponding results from the cross-species analysis.

A Violin and Box Plots of Predictions Across Species

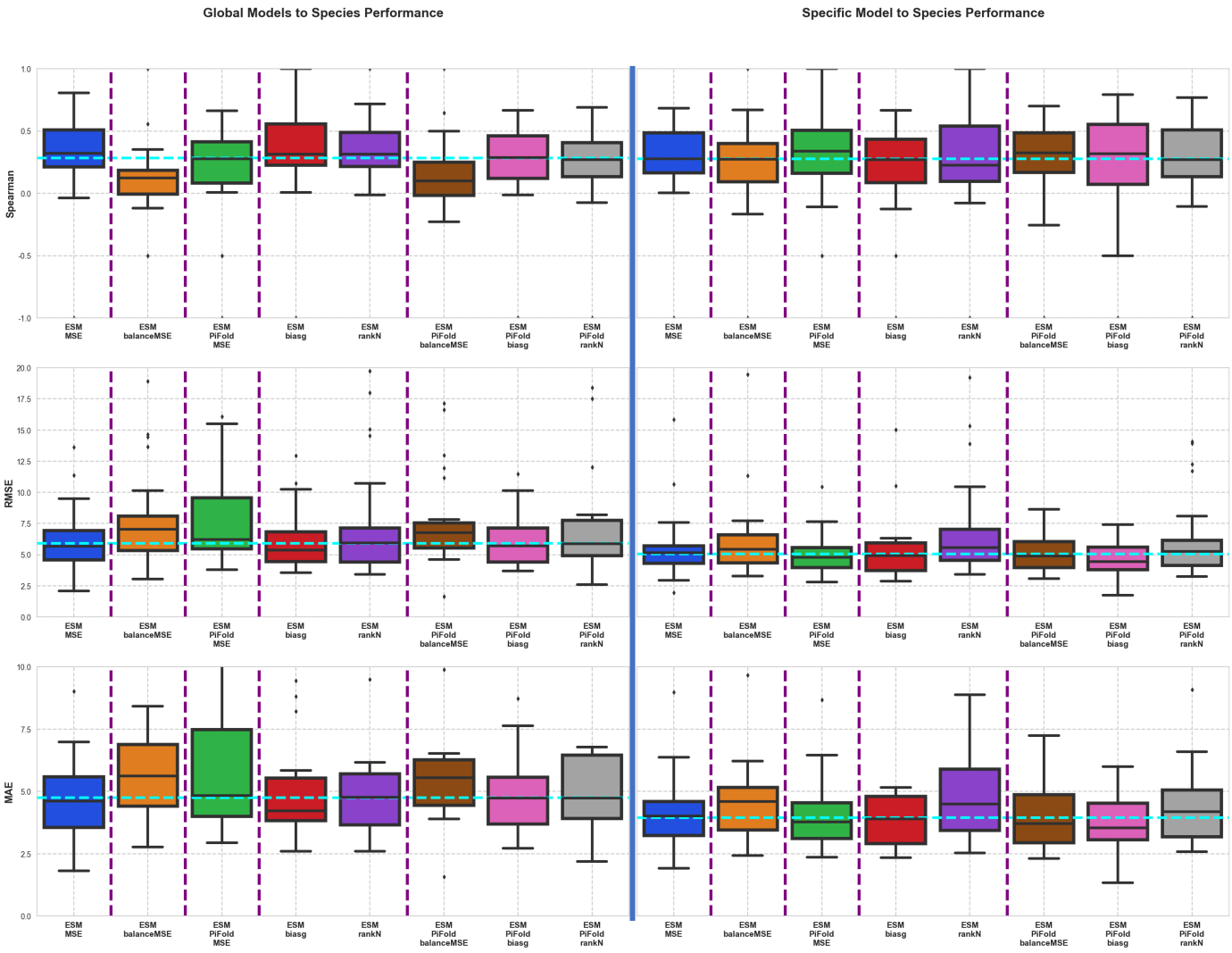

Figure 5: Box plots of the strategies used to predict thermostability. The blue line separates the use of global models applied to each species (right side) from the use of individual models for each species (left side). In addition, each region has subdivisions marked in purple, indicating the order in which the ideas were first tested: First, the prediction method using sequence embeddings was implemented. Next, species-by-species sampling was used in batches. Then the combination of sequence embeddings with inverse folding was used. Next, contrastive methods using sequence embeddings alone were applied. Finally, sequence embeddings were combined with contrastive methods. The cyan line crossing the box plots represents the average of the medians of all box plots. The points at the top and bottom of the box plots correspond to the outliers present in the predictions. Compared to the Boxplot shown in the main paper, this presents the boxplots related to all the performance metrics considered along the work, including: Spearman Correlation, RMSE and MAE respectively.

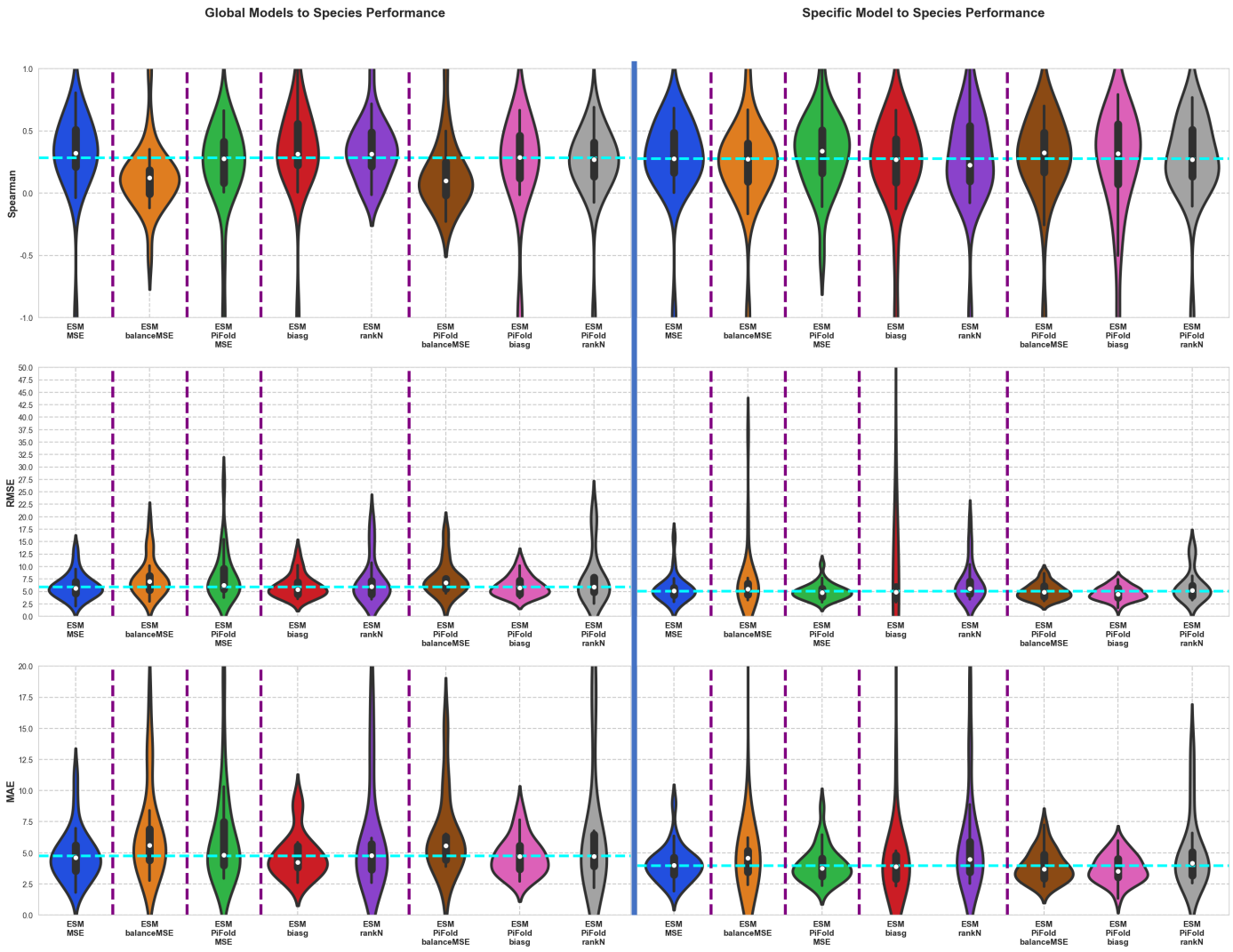

Figure 6: Violin plots of the strategies used to predict thermostability. The blue line separates the use of global models applied to each species (right side) from the use of individual models for each species (left side). In addition, each region has subdivisions marked in purple, indicating the order in which the ideas were first tested: First, the prediction method using sequence embeddings was implemented. Next, species-by-species sampling was used in batches. Then the combination of sequence embeddings with inverse folding was used. Next, contrastive methods using sequence embeddings alone were applied. Finally, sequence embeddings were combined with contrastive methods. The cyan line crossing the box plots represents the average of the medians of all violin plots. This plot show the variance and density estimation to all the performance metrics considered along the work, including: Spearman Correlation, RMSE and MAE respectively.

B Tables of Results for the Full Flip Split

Methods	global			
	Spearman	MSE	RMSE	MAE
MSE	0.688	37.303	6.1076	4.5268
balanceMSE	0.415	61.8778	7.8662	6.0651
biasg	0.6951	37.8031	6.1484	4.492
rank_l2_l1_MSE_001_t2	0.6356	53.1513	7.2905	5.3756

Table 3: Results of methods using sequence-based embeddings (ESM2). Performance evaluation using the full dataset of FLIP splits related to thermostability.

Methods	global			
	Spearman	MSE	RMSE	MAE
MSE	0.3763	87.9824	9.3799	6.788
balanceMSE	0.3563	100.758	10.0378	7.1735
biasg	0.3961	88.4589	9.4053	6.6275
rank_l2_l1_MSE_001_t2	0.309	98.5775	9.9286	7.0107

Table 4: Results of methods using embeddings from protein structures using Inverse Folding Algorithms (PiFold). Performance evaluation using the full dataset of FLIP splits related to thermostability.

Methods	global			
	Spearman	MSE	RMSE	MAE
MSE	0.679	42.8623	6.5469	4.7677
balanceMSE	0.5018	52.7803	7.265	5.676
biasg	0.6912	36.452	6.0376	4.481
rank_l2_l1_MSE_001_t2	0.6219	65.2838	8.0798	5.7973

Table 5: Results of methods using the concatenation of embeddings from ESM2 and PiFold. Performance evaluation using the full dataset of FLIP splits related to thermostability.

C Tables of Results for Global Models Applied to Individual Species

Methods	balanceMSE				biasg			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	-0.0988	56.3553	7.507	5.9166	0.1737	40.3051	6.3486	4.6455
Bacillus_subtilis_168_lysate_R1	-0.1196	102.93	10.1454	8.4139	0.0487	65.4492	8.0901	5.8427
Caenorhabditis_elegans_lysate	0.1683	61.1264	7.8183	6.6306	0.2485	28.5346	5.3418	4.4649
Danio_rerio_Zenodo_lysate	0.0818	84.3226	9.1827	8.0521	0.7636	48.2345	6.9451	5.6205
Drosophila_melanogaster_SII_lysate	0.0362	49.3716	7.0265	5.8863	0.3217	14.5884	3.8195	3.0671
Ecoli_Lysate	0.1727	51.7253	7.192	5.8725	0.2293	53.0866	7.2861	5.7959
Escherichia_coli_cells	0.5557	51.5221	7.1779	6.3236	0.4178	104.673	10.231	8.8138
Geobacillus_stearothermophilus_NCA26_lysate	0.3519	186.93	13.6722	12.2979	0.2649	167.706	12.9501	9.4482
HAOEC	0.0601	49.3427	7.0244	5.1104	0.5448	34.5538	5.8782	4.3843
HEK293T	0.0989	21.5179	4.6387	3.5816	0.5893	14.4708	3.804	2.9393
HL60	0.2614	15.497	3.9366	3.0002	0.5686	16.5823	4.0721	2.725
HaCaT	0.1179	39.1102	6.2538	4.7219	0.5411	28.7117	5.3583	4.04
HepG2	-0.0364	42.8814	6.5484	5.2415	0.6338	25.9643	5.0955	3.9699
Jurkat	0.2568	27.7194	5.2649	4.383	0.562	19.2444	4.3868	3.6055
K562	1	9.2415	3.04	2.7739	-1	21.4709	4.6337	3.8368
Mus_musculus_BMDC_lysate	-0.0792	18.0147	4.2444	3.1469	0.2714	19.0719	4.3671	2.96
Mus_musculus_liver_lysate	0.1486	44.4799	6.6693	5.3624	0.1226	41.9219	6.4747	5.245
Oleispira_antarctica_RB-8_lysate_R1	0.1053	67.0532	8.1886	6.9769	0.2248	21.8788	4.6775	3.9502
Picrophilus_torridus_DSM9790_lysate	0.1438	208.776	14.4491	13.3615	0.2341	81.201	9.0112	5.7109
Saccharomyces_cerevisiae_lysate	0.1898	21.3975	4.6257	3.5893	0.3053	26.9074	5.1872	3.9288
Thermus_thermophilus_HB27_cells	0.1255	357.955	18.9197	17.3433	0.2166	115.26	10.7359	8.2072
Thermus_thermophilus_HB27_lysate	0.3065	214.4	14.6424	13.2952	0.3608	41.2089	6.4194	5.1294
U937	-0.0889	32.4743	5.6986	4.4815	0.6765	12.6116	3.5513	2.6062
colon_cancer_spheroids	-0.0208	24.8119	4.9812	3.9806	0.0078	23.8056	4.8791	3.8172
pTcells	-0.5	29.8796	5.4662	4.889	1	17.0161	4.1251	4.0651
MEANS	0.1295	74.7534	7.77256	6.5853	0.333108	43.3783	6.14678	4.75278

Methods	MSE				rankN			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	0.1713	41.6821	6.4562	4.8311	0.0795	42.2389	6.4991	4.9576
Bacillus_subtilis_168_lysate_R1	0.0586	66.5909	8.1603	6.0094	-0.012	54.0174	7.3496	5.681
Caenorhabditis_elegans_lysate	0.34	31.8574	5.6442	4.7308	0.2828	43.6121	6.6039	5.7153
Danio_rerio_Zenodo_lysate	0.3636	49.33	7.0235	5.6581	0.7182	41.5056	6.4425	4.9886
Drosophila_melanogaster_SII_lysate	0.287	22.8456	4.7797	3.9267	0.2742	28.5485	5.3431	4.566
Ecoli_Lysate	0.2485	43.9449	6.6291	5.3792	0.2318	45.1501	6.7194	5.314
Escherichia_coli_cells	0.4553	71.6161	8.4626	6.9848	0.418	115.169	10.7317	9.4907
Geobacillus_stearothermophilus_NCA26_lysate	0.304	185.641	13.625	11.2403	0.3045	390.007	19.7486	18.7062
HAOEC	0.5905	28.7833	5.365	3.963	0.5018	33.2474	5.7661	4.2156
HEK293T	0.5569	12.9611	3.6001	2.818	0.4592	14.6691	3.83	3.0526
HL60	0.6305	13.4951	3.6736	2.5821	0.591	11.7356	3.4257	2.5935
HaCaT	0.5127	27.3408	5.2288	4.1923	0.378	31.8441	5.6431	4.5837
HepG2	0.6174	21.9505	4.6851	3.8385	0.5724	25.8588	5.0852	4.028
Jurkat	0.611	17.6694	4.2035	3.409	0.5221	19.1855	4.3801	3.6597
K562	-1	33.9431	5.8261	4.5849	1	15.6871	3.9607	3.8239
Mus_musculus_BMDC_lysate	0.264	18.5152	4.3029	3.0061	0.2366	13.5068	3.6752	2.6633
Mus_musculus_liver_lysate	0.0522	38.733	6.2236	5.0271	-0.0144	39.0666	6.2503	5.151
Oleispira_antarctica_RB-8_lysate_R1	0.2023	32.1654	5.6715	4.637	0.2087	37.8013	6.1483	5.2765
Picrophilus_torridus_DSM9790_lysate	0.1184	89.9399	9.4837	6.5683	0.0982	211.472	14.5421	12.2113
Saccharomyces_cerevisiae_lysate	0.3373	20.797	4.5604	3.4606	0.3222	20.4986	4.5275	3.4832
Thermus_thermophilus_HB27_cells	0.2481	129.526	11.3809	9.0225	0.1148	323.852	17.9959	16.0593
Thermus_thermophilus_HB27_lysate	0.3968	50.3902	7.0986	5.839	0.3073	226.428	15.0475	13.5913
U937	0.8059	9.5045	3.0829	2.3054	0.3971	17.5221	4.1859	3.3232
colon_cancer_spheroids	-0.0378	24.876	4.9876	3.8913	0.0026	20.7494	4.5552	3.6538
pTcells	0.5	4.3833	2.0936	1.8125	0.5	11.8894	3.4481	3.4415
MEANS	0.30538	43.5393	6.08994	4.78872	0.339784	73.4105	7.27619	6.16923

Table 6: Evaluation of methods using sequence-based embeddings (ESM2): Performance evaluation was performed using the full training and validation sets of FLIP partitions related to thermostability to induce the model. Testing was performed on individual species within the FLIP test set. The table shows the evaluation based on four primary metrics: Spearman correlation, MSE, RMSE and MAE, for the four methods analysed in this study (balanceMSE, biasg, MSE and RankN).

Methods	balanceMSE				biasg			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	0.1978	43.6081	6.6036	5.1771	0.0682	47.5744	6.8974	5.3945
Bacillus_subtilis_168_lysate_R1	0.1186	64.2171	8.0136	6.2432	0.1013	90.2633	9.5007	7.0436
Caenorhabditis_elegans_lysate	0.1135	77.9945	8.8314	7.4103	0.1488	67.6063	8.2223	6.8556
Danio_rerio_Zenodo_lysate	0.1545	67.2587	8.2011	5.6998	0.4818	85.5648	9.2501	7.8557
Drosophila_melanogaster_SII_lysate	0.1883	42.6002	6.5269	5.436	0.0446	51.1427	7.1514	5.9902
Ecoli_Lysate	0.0712	61.3955	7.8355	6.194	0.101	72.936	8.5403	6.6497
Escherichia_coli_cells	0.2221	135.125	11.6243	10.0279	0.0373	176.004	13.2666	11.8437
Geobacillus_stearothermophilus_NCA26_lysate	0.088	498.251	22.3215	21.2174	0.0049	478.348	21.8712	19.6788
HAOEC	0.3888	49.0695	7.005	5.4388	0.3905	41.4828	6.4407	4.8899
HEK293T	0.3737	24.4279	4.9425	3.768	0.3315	21.264	4.6113	3.5963
HL60	0.2711	23.8911	4.8879	3.6236	0.3725	17.2556	4.154	3.2397
HaCaT	0.2216	45.1755	6.7213	4.7603	0.242	37.8045	6.1485	4.8963
HepG2	0.3273	38.7255	6.223	4.6599	0.4269	34.835	5.9021	4.8028
Jurkat	0.1976	34.0146	5.8322	4.6808	0.4273	25.7759	5.077	4.1718
K562	-1	4.0299	2.0075	1.5151	-1	18.3807	4.2873	4.287
Mus_musculus_BMDC_lysate	0.1254	26.6945	5.1667	3.6526	0.1506	27.9881	5.2904	3.6731
Mus_musculus_liver_lysate	0.1326	46.4017	6.8119	5.5285	0.1947	46.4456	6.8151	5.6925
Oleispira_antarctica_RB-8_lysate_R1	0.1121	59.6047	7.7204	6.3074	0.0632	61.3767	7.8343	6.0074
Picrophilus_torridus_DSM9790_lysate	-0.041	410.771	20.2675	18.9768	0.126	305.723	17.4849	14.515
Saccharomyces_cerevisiae_lysate	0.053	32.4263	5.6944	4.5244	0.3205	29.4371	5.4256	4.2492
Thermus_thermophilus_HB27_cells	-0.1055	838.375	28.9547	26.1275	-0.0737	642.825	25.354	20.9251
Thermus_thermophilus_HB27_lysate	0.0468	582.313	24.1312	21.609	-0.0441	483.995	21.9999	17.1478
U937	0.3706	25.0477	5.0048	4.2869	0.1	30.4037	5.514	4.1933
colon_cancer_spheroids	0.0613	26.1402	5.1127	4.1275	0.2479	38.4994	6.2048	4.5201
pTcells	0.5	34.794	5.8986	5.8971	-0.5	50.3598	7.0965	6.875
MEANS	0.127576	131.694	9.29361	7.8756	0.110548	119.332	9.21362	7.55976

Methods	MSE				rankN			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	0.0615	49.9119	7.0648	5.4569	0.04	47.601	6.8993	5.3863
Bacillus_subtilis_168_lysate_R1	0.0577	77.1768	8.785	6.8443	-0.0649	67.5899	8.2213	6.6734
Caenorhabditis_elegans_lysate	0.2483	88.5995	9.4127	8.0938	-0.0366	76.9103	8.7699	7.6532
Danio_rerio_Zenodo_lysate	-0.0182	63.064	7.9413	5.9462	0.6545	53.9727	7.3466	5.702
Drosophila_melanogaster_SII_lysate	0.2405	56.1757	7.495	6.493	0.3073	40.6112	6.3727	5.6224
Ecoli_Lysate	0.0873	66.5738	8.1593	6.4321	-0.0549	59.2102	7.6948	6.0148
Escherichia_coli_cells	0.0935	141.508	11.8957	10.15	0.0313	108.28	10.4057	9.1288
Geobacillus_stearothermophilus_NCA26_lysate	-0.0762	402.561	20.0639	18.2502	-0.2718	397.585	19.9395	18.0143
HAOEC	0.389	37.326	6.1095	4.579	0.3736	47.5098	6.8927	5.0855
HEK293T	0.3936	23.1623	4.8127	3.5236	0.3369	23.2582	4.8227	3.5999
HL60	0.3305	19.1996	4.3817	3.4348	0.354	19.2183	4.3839	3.1957
HaCaT	0.0785	40.7701	6.3851	5.359	0.3414	39.0351	6.2478	4.8033
HepG2	0.5175	25.7657	5.076	4.0447	0.4026	34.4372	5.8683	4.7732
Jurkat	0.5916	17.522	4.1859	3.4371	0.2134	31.7117	5.6313	4.5822
K562	1	5.6946	2.3863	2.1568	0	5.059	2.2492	1.8709
Mus_musculus_BMDC_lysate	0.21	20.926	4.5745	3.4033	0.1203	16.1929	4.024	2.9002
Mus_musculus_liver_lysate	-0.0108	46.809	6.8417	5.7137	-0.0597	48.1737	6.9407	5.7201
Oleispira_antarctica_RB-8_lysate_R1	0.0712	74.8146	8.6495	7.3418	0.1333	72.2167	8.498	7.3424
Picrophilus_torridus_DSM9790_lysate	0.1537	314.616	17.7374	15.7521	-0.0237	393.507	19.837	18.6257
Saccharomyces_cerevisiae_lysate	0.1838	21.5273	4.6398	3.6465	0.1058	21.5062	4.6375	3.6898
Thermus_thermophilus_HB27_cells	-0.1581	815.522	28.5573	25.5008	-0.013	860.323	29.3313	26.241
Thermus_thermophilus_HB27_lysate	0.0875	492.982	22.2032	19.2164	0.0335	609.46	24.6873	22.0347
U937	0.4765	22.0209	4.6926	3.7639	0.3316	28.9023	5.3761	4.2117
colon_cancer_spheroids	0.3813	23.1553	4.812	3.6682	0.27	18.2909	4.2768	3.5334
pTcells	0.5	46.0152	6.7835	6.4249	-0.5	36.0084	6.0007	5.9248
MEANS	0.235608	119.736	8.94586	7.54532	0.120996	126.263	9.0142	7.69319

Table 7: Evaluation of methods using embeddings from protein structures via inverse folding (PiFold): Performance evaluation was performed using the full training and validation sets of FLIP partitions related to thermostability to induce the model. Testing was performed on individual species within the FLIP test set. The table shows the evaluation based on four primary metrics: Spearman correlation, MSE, RMSE and MAE, for the four methods analysed in this study (balanceMSE, biasg, MSE and RankN).

Methods	balanceMSE				biasg			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	-0.0328	44.9936	6.7077	5.1725	0.1137	42.7874	6.5412	5.0204
Bacillus_subtilis_168_lysate_R1	0.0251	48.0577	6.9324	5.6365	-0.0123	75.4061	8.6837	6.5255
Caenorhabditis_elegans_lysate	0.1034	45.9781	6.7807	5.8325	0.3285	32.324	5.6854	4.6984
Danio_rerio_Zenodo_lysate	0.6455	61.3115	7.8302	6.5264	0.6455	46.4755	6.8173	5.611
Drosophila_melanogaster_SII_lysate	0.1124	25.6896	5.0685	4.1189	0.3215	20.6017	4.5389	3.6341
Ecoli_Lysate	0.0858	50.1222	7.0797	5.7697	0.2931	43.929	6.6279	5.4187
Escherichia_coli_cells	0.3961	124.554	11.1604	9.8918	0.3531	72.1184	8.4923	7.3403
Geobacillus_stearothermophilus_NCA26_lysate	0.2636	293.461	17.1307	15.6566	0.2785	131.69	11.4756	8.7325
HAOEC	-0.0117	52.6367	7.2551	5.5859	0.5974	28.9445	5.38	3.9445
HEK293T	-0.0166	30.0877	5.4852	4.3884	0.4751	16.0743	4.0093	3.0436
HL60	-0.026	27.6587	5.2592	4.0565	0.5124	13.6517	3.6948	2.713
HaCaT	-0.2289	55.3317	7.4385	5.8553	0.6654	24.5686	4.9567	4.0428
HepG2	-0.0209	47.0074	6.8562	5.5277	0.5767	23.547	4.8525	3.8697
Jurkat	-0.2288	45.9317	6.7773	5.9716	0.5776	16.0434	4.0054	3.3378
K562	1	2.7174	1.6485	1.5705	-1	55.7615	7.4674	5.3456
Mus_musculus_BMDC_lysate	-0.0049	23.717	4.87	3.8992	0.2541	18.3161	4.2797	2.9783
Mus_musculus_liver_lysate	0.1471	43.8143	6.6192	5.4036	0.1062	38.0584	6.1692	4.8434
Oleispira_antarctica_RB-8_lysate_R1	0.1304	32.0531	5.6616	4.8655	0.1592	32.9704	5.742	4.7691
Picrophilus_torridus_DSM9790_lysate	0.0946	168.576	12.9837	10.7069	-0.0145	80.3387	8.9632	6.2426
Saccharomyces_cerevisiae_lysate	0.0423	30.4126	5.5148	4.3921	0.3267	18.9499	4.3531	3.2769
Thermus_thermophilus_HB27_cells	0.2346	276.846	16.6387	14.5587	0.1349	102.891	10.1435	7.6395
Thermus_thermophilus_HB27_lysate	0.3538	143.249	11.9687	10.2945	0.2823	52.6987	7.2594	5.6441
U937	0.2529	33.3963	5.779	4.7876	0.4147	14.9665	3.8687	3.0603
colon_cancer_spheroids	-0.1639	31.1899	5.5848	4.6067	0.0389	24.9092	4.9909	3.8851
pTcells	0.5	21.1659	4.6006	4.2683	-1	19.0944	4.3697	3.8958
MEANS	0.146124	70.3984	7.58526	6.37376	0.217148	41.8846	6.13471	4.78052

Methods	MSE				rankN			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	0.109	39.2262	6.2631	4.8214	0.1274	36.2154	6.0179	4.5516
Bacillus_subtilis_168_lysate_R1	0.0245	59.9784	7.7446	5.7714	0.0426	37.9457	6.16	4.8815
Caenorhabditis_elegans_lysate	0.3299	32.1148	5.667	4.8481	0.2677	33.5518	5.7924	4.8652
Danio_rerio_Zenodo_lysate	0.6636	47.1145	6.864	5.3625	0.6909	67.0075	8.1858	6.6925
Drosophila_melanogaster_SII_lysate	0.2991	22.454	4.7386	3.8411	0.381	17.4374	4.1758	3.4377
Ecoli_Lysate	0.2094	47.4049	6.8851	5.5404	0.2948	52.1178	7.2193	5.7509
Escherichia_coli_cells	0.3589	99.9341	9.9967	8.4899	0.2228	144.307	12.0128	10.624
Geobacillus_stearothermophilus_NCA26_lysate	0.2144	258.657	16.0828	13.9394	0.271	464.474	21.5517	20.7134
HAOEC	0.5121	32.7875	5.726	4.1267	0.4289	45.9515	6.7788	5.0711
HEK293T	0.4918	14.5158	3.81	2.9418	0.3636	18.5763	4.31	3.312
HL60	0.5576	25.8021	5.0796	3.0008	0.4153	15.7443	3.9679	2.8836
HaCaT	0.4221	29.4583	5.4275	4.4413	0.2096	35.3295	5.9439	4.6046
HepG2	0.5828	31.2366	5.589	4.0038	0.5321	31.8896	5.6471	4.3951
Jurkat	0.5608	34.4767	5.8717	4.0023	0.426	25.9512	5.0942	4.2632
K562	-1	714.098	26.7226	19.1323	-1	6.7103	2.5904	2.1838
Mus_musculus_BMDC_lysate	0.266	16.9931	4.1223	3.0564	0.2084	14.4295	3.7986	2.7389
Mus_musculus_liver_lysate	0.0062	40.8254	6.3895	5.0758	0.0941	48.9111	6.9936	5.7595
Oleispira_antarctica_RB-8_lysate_R1	0.1337	24.9606	4.9961	4.1284	0.1489	27.9463	5.2864	4.5097
Picrophilus_torridus_DSM9790_lysate	0.0185	119.918	10.9507	8.2336	0.0208	338.396	18.3955	16.5375
Saccharomyces_cerevisiae_lysate	0.3261	20.3761	4.514	3.4657	0.2844	27.5199	5.2459	4.2572
Thermus_thermophilus_HB27_cells	0.073	170.008	13.0387	10.0611	0.0676	431.125	20.7636	18.904
Thermus_thermophilus_HB27_lysate	0.2913	120.471	10.9759	7.906	0.2938	307.381	17.5323	16.1797
U937	0.3941	38.377	6.1949	3.9725	0.6	19.7639	4.4457	3.4367
colon_cancer_spheroids	0.0681	33.1918	5.7612	4.0455	-0.0761	23.4299	4.8404	3.8038
pTcells	-0.5	239.848	15.487	10.3164	0.5	27.3404	5.2288	5.0902
MEANS	0.21652	92.5691	8.19594	6.18098	0.232624	91.9781	7.91915	6.7779

Table 8: Evaluation of methods using concatenation of embeddings from ESM2 and PiFold: Performance evaluation was performed using the full training and validation sets of FLIP partitions related to thermostability to induce the model. Testing was performed on individual species within the FLIP test set. The table shows the evaluation based on four primary metrics: Spearman correlation, MSE, RMSE and MAE, for the four methods analysed in this study (balanceMSE, biasg, MSE and RankN).

D Tables of Results for Individual Model per Species

Methods	balanceMSE				biasg			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	0.2852	35.1803	5.9313	4.571	0.3915	28.9075	5.3766	4.0307
Bacillus_subtilis_168_lysate_R1	0.2912	24.1663	4.9159	3.7388	0.3267	23.8897	4.8877	3.8329
Caenorhabditis_elegans_lysate	0.3766	379.356	19.4771	4.6509	0.45	21.898	4.6795	3.5624
Danio_rerio_Zenodo_lysate	0.0909	128.475	11.3347	9.6606	-0.1273	225.52	15.0173	12.0062
Drosophila_melanogaster_SII_lysate	0.1526	15.3985	3.9241	3.0623	0.2276	12.6765	3.5604	2.8362
Ecoli_Lysate	0.407	36.5361	6.0445	4.8506	0.4971	33.2729	5.7683	4.626
Escherichia_coli_cells	0.5233	45.1256	6.7176	5.6226	0.6143	28.6546	5.353	4.3872
Geobacillus_stearothermophilus_NCA26_lysate	0.0901	39.4851	6.2837	5.2636	0.0798	35.8998	5.9916	4.8417
HAOEC	0.6683	18.0296	4.2461	3.3848	0.6677	17.5784	4.1927	3.1416
HEK293T	0.3765	16.5299	4.0657	3.1876	0.6461	11.0681	3.3269	2.6517
HL60	0.4633	11.4203	3.3794	2.7111	0.6226	9.6282	3.1029	2.5825
HaCaT	0.2289	44.3847	6.6622	4.8588	0.2811	39.1072	6.2536	4.9183
HepG2	0.428	31.5882	5.6203	4.3371	0.05	39.9588	6.3213	5.1588
Jurkat	0.4148	24.0002	4.899	3.8913	0.3752	24.4667	4.9464	4.2259
K562	-1	23.8188	4.8804	4.6064	-1	8.9454	2.9909	2.624
Mus_musculus_BMDC_lysate	0.3239	10.8574	3.2951	2.4241	0.3697	10.7146	3.2733	2.3409
Mus_musculus_liver_lysate	-0.1671	36.438	6.0364	4.8889	0.0819	32.2965	5.683	4.659
Oleispira_antarctica_RB-8_lysate_R1	0.0454	26.8454	5.1813	3.8702	0.0335	18.6379	4.3172	3.2358
Picrophilus_torridus_DSM9790_lysate	0.165	41.0108	6.404	4.9941	0.2599	24.9528	4.9953	3.9412
Saccharomyces_cerevisiae_lysate	0.2978	17.0026	4.1234	3.1669	0.3283	17.5996	4.1952	3.1873
Thermus_thermophilus_HB27_cells	0.0117	1354.8	36.8076	29.4928	0.1002	20153.3	141.962	25.2994
Thermus_thermophilus_HB27_lysate	0.1008	59.3835	7.7061	6.2203	0.2217	36.9322	6.0772	4.8796
U937	0.2618	18.036	4.2469	3.2158	0.6588	11.882	3.447	2.6935
colon_cancer_spheroids	-0.0616	21.4252	4.6287	3.6292	0.1363	17.5956	4.1947	3.4264
pTcells	1	27.8208	5.2745	5.2123	-0.5	8.1613	2.8568	2.5818
MEANS	0.230976	99.4846	7.28344	5.42048	0.231708	835.74	10.5108	4.86684

Methods	MSE				rankN			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	0.2757	31.0004	5.5678	4.3043	0.1801	33.4706	5.7854	4.4929
Bacillus_subtilis_168_lysate_R1	0.2119	26.0025	5.0993	3.944	-0.0782	26.5269	5.1504	3.9686
Caenorhabditis_elegans_lysate	0.4443	251.104	15.8463	4.3104	0.4126	23.4481	4.8423	3.6256
Danio_rerio_Zenodo_lysate	0.1636	113.312	10.6448	8.979	0.4818	369.521	19.2229	18.4969
Drosophila_melanogaster_SII_lysate	0.3183	11.3899	3.3749	2.6573	0.1157	11.7702	3.4308	2.7413
Ecoli_Lysate	0.4357	35.479	5.9564	4.8229	0.4253	42.1144	6.4896	5.3407
Escherichia_coli_cells	0.5606	32.7802	5.7254	4.8546	0.3895	70.6396	8.4047	7.2346
Geobacillus_stearothermophilus_NCA26_lysate	0.2564	31.6178	5.623	4.5296	0.1179	34.9162	5.909	4.8523
HAOEC	0.6832	16.7097	4.0877	3.2149	0.6621	20.0385	4.4764	3.2012
HEK293T	0.5759	12.533	3.5402	2.7081	0.6298	12.7439	3.5699	2.8206
HL60	0.6247	8.6127	2.9347	2.4174	0.6328	12.4881	3.5338	2.8032
HaCaT	0.6781	23.5524	4.8531	3.6007	0.2961	81.834	9.0462	7.3093
HepG2	0.4166	27.5064	5.2447	4.198	0.7306	18.2769	4.2751	3.3763
Jurkat	0.5497	20.3893	4.5155	3.6091	0.56	26.8047	5.1773	4.1779
K562	-1	20.0878	4.4819	3.5169	-1	33.2189	5.7636	5.5281
Mus_musculus_BMDC_lysate	0.3265	10.8	3.2863	2.3875	0.0678	12.1015	3.4787	2.5353
Mus_musculus_liver_lysate	0.094	31.4523	5.6082	4.6244	0.0873	29.042	5.3891	4.4974
Oleispira_antarctica_RB-8_lysate_R1	0.0911	18.3544	4.2842	3.2036	-0.0215	109.501	10.4643	8.8855
Picrophilus_torridus_DSM9790_lysate	0.222	29.4879	5.4303	4.3721	0.0355	234.615	15.3172	13.448
Saccharomyces_cerevisiae_lysate	0.2468	18.84	4.3405	3.2997	0.1777	17.2934	4.1585	3.3133
Thermus_thermophilus_HB27_cells	0.1717	57.7222	7.5975	6.3657	0.1418	50.6287	7.1154	5.9142
Thermus_thermophilus_HB27_lysate	0.0034	50.3409	7.0951	5.8207	0.1577	38.7479	6.2248	5.1955
U937	0.0412	51.165	7.153	4.7931	0.5824	23.3845	4.8357	3.8638
colon_cancer_spheroids	0.1359	19.8405	4.4543	3.6356	-0.02	22.0879	4.6998	3.6528
pTcells	0.5	3.8066	1.9511	1.9092	1	193.474	13.9095	13.5656
MEANS	0.281092	38.1555	5.54785	4.08315	0.270592	61.9475	6.82682	5.79364

Table 9: Assessment of methodologies employing sequence-based embeddings from ESM2: The performance of individual models was assessed for each species and subsequently tested on that target species inside the FLIP partition. Spearman correlation, mean square error (MSE), root mean square error (RMSE) and mean absolute error (MAE). These metrics were used to analyse the effectiveness of the four methods investigated in this study: balanceMSE, biasg, MSE and RankN.

Methods	balanceMSE				biasg			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	0.1901	33.1926	5.7613	4.416	0.2502	31.8047	5.6396	4.36
Bacillus_subtilis_168_lysate_R1	0.235	26.0891	5.1078	3.8833	0.2005	26.9053	5.187	4.0578
Caenorhabditis_elegans_lysate	0.2814	23.3374	4.8309	3.8388	0.2717	24.0761	4.9067	3.7928
Danio_rerio_Zenodo_lysate	0.0545	55.0791	7.4215	5.9855	-0.1909	85.0409	9.2218	6.6883
Drosophila_melanogaster_SII_lysate	0.332	11.2	3.3466	2.6635	0.2971	11.8738	3.4458	2.6509
Ecoli_Lysate	0.2438	49.9904	7.0704	5.551	0.3893	38.8821	6.2355	5.0328
Escherichia_coli_cells	0.2479	62.1012	7.8804	6.5516	0.1184	53.963	7.346	6.2021
Geobacillus_stearothermophilus_NCA26_lysate	0.0345	43.1849	6.5715	5.3951	0.0842	41.817	6.4666	5.1216
HAOEC	0.5553	21.1289	4.5966	3.6297	0.5702	21.5063	4.6375	3.5582
HEK293T	0.3274	18.916	4.3493	3.4813	0.3553	16.8959	4.1105	3.3724
HL60	0.5285	12.7037	3.5642	2.8803	0.5094	12.7012	3.5639	2.8331
HaCaT	0.0953	46.8168	6.8423	5.0379	0.0528	57.0223	7.5513	5.5844
HepG2	0.4819	27.8663	5.2789	4.0159	0.5148	25.9791	5.097	4.1233
Jurkat	0.3438	27.167	5.2122	4.2192	0.0998	29.5164	5.4329	4.4984
K562	-1	20.4271	4.5196	4.0903	-1	14.7407	3.8394	3.6112
Mus_musculus_BMDC_lysate	0.1828	12.4816	3.5329	2.5956	0.2007	12.7979	3.5774	2.5749
Mus_musculus_liver_lysate	0.1398	29.6782	5.4478	4.4864	0.1861	139.129	11.7953	5.8553
Oleispira_antarctica_RB-8_lysate_R1	-0.0207	24.0207	4.9011	3.762	0.0777	17.7828	4.217	3.183
Picrophilus_torridus_DSM9790_lysate	0.0646	37.0564	6.0874	4.8446	0.113	28.2721	5.3171	4.3029
Saccharomyces_cerevisiae_lysate	0.0481	19.6548	4.4334	3.4708	0.1198	18.515	4.3029	3.395
Thermus_thermophilus_HB27_cells	-0.0552	73.09	8.5493	7.1818	-0.0566	63.1341	7.9457	6.5063
Thermus_thermophilus_HB27_lysate	0.0847	45.8389	6.7704	5.6529	0.1028	41.072	6.4087	5.3268
U937	0.5971	15.4958	3.9365	2.7684	0.1588	17.6512	4.2013	3.4349
colon_cancer_spheroids	0.1604	17.373	4.1681	3.4119	0.0388	19.2596	4.3886	3.5495
pTcells	0.5	26.5078	5.1486	4.3754	0.5	8.3234	2.885	2.6848
MEANS	0.18612	31.2159	5.41316	4.32757	0.158556	34.3465	5.50882	4.25203

Methods	MSE				rankN			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	0.1513	33.8696	5.8198	4.4579	0.1197	32.8702	5.7332	4.5013
Bacillus_subtilis_168_lysate_R1	0.1527	28.1217	5.303	4.1673	0.147	26.1166	5.1104	4.0038
Caenorhabditis_elegans_lysate	0.3198	24.1727	4.9166	3.7621	0.1201	25.0226	5.0023	3.9697
Danio_rerio_Zenodo_lysate	0.6	33.9685	5.8282	4.6552	0.1545	39.9804	6.323	4.7511
Drosophila_melanogaster_SII_lysate	0.3204	10.97	3.3121	2.5842	0.3396	12.9318	3.5961	2.8986
Ecoli_Lysate	0.2583	42.3656	6.5089	5.2799	0.4019	48.3124	6.9507	5.6156
Escherichia_coli_cells	0.367	58.9055	7.675	6.5599	-0.0351	61.2983	7.8293	6.7248
Geobacillus_stearothermophilus_NCA26_lysate	-0.106	39.5683	6.2903	4.9254	0.1681	33.9255	5.8246	4.5943
HAOEC	0.6131	19.766	4.4459	3.5534	0.5151	24.2065	4.92	3.6582
HEK293T	0.5067	14.7227	3.837	3.0679	0.4558	14.4314	3.7989	2.998
HL60	0.5877	9.8337	3.1359	2.5131	0.1545	16.2364	4.0294	3.0754
HaCaT	0.4174	34.1148	5.8408	4.4338	0.3616	36.3923	6.0326	4.6482
HepG2	0.6413	19.7246	4.4412	3.4868	0.5916	23.1588	4.8124	3.8102
Jurkat	0.3163	26.6553	5.1629	4.2983	0.271	24.4258	4.9422	4.2085
K562	1	27.3812	5.2327	5.1922	-1	5.3537	2.3138	1.9069
Mus_musculus_BMDC_lysate	0.2036	12.4449	3.5277	2.4947	0.1087	11.8251	3.4388	2.535
Mus_musculus_liver_lysate	0.2019	26.4774	5.1456	4.1962	-0.0673	30.653	5.5365	4.6092
Oleispira_antarctica_RB-8_lysate_R1	0.0616	18.4959	4.3007	3.3799	0.1153	16.5094	4.0632	3.0985
Picrophilus_torridus_DSM9790_lysate	0.0001	31.7089	5.6311	4.547	0.2134	31.4108	5.6045	4.391
Saccharomyces_cerevisiae_lysate	0.2475	17.3746	4.1683	3.1798	0.2441	23.5091	4.8486	3.9765
Thermus_thermophilus_HB27_cells	-0.015	58.7667	7.6659	6.0328	-0.057	48.9728	6.9981	5.7465
Thermus_thermophilus_HB27_lysate	0.0471	49.1931	7.0138	5.7324	0.1022	37.4915	6.123	5.083
U937	-0.1235	24.0018	4.8992	3.9386	-0.1324	18.1745	4.2632	3.5649
colon_cancer_spheroids	0.0761	18.1357	4.2586	3.4174	0.0622	17.8032	4.2194	3.4283
pTcells	0.5	26.305	5.1288	4.9994	-0.5	18.0692	4.2508	3.7284
MEANS	0.293816	28.2818	5.1796	4.19422	0.114184	27.1633	5.0626	4.06104

Table 10: Assessment of methodologies employing embeddings from protein structures using Inverse Folding algorithms (PiFold): The performance of individual models was assessed for each species and subsequently tested on that target species inside the FLIP partition. Spearman correlation, mean square error (MSE), root mean square error (RMSE) and mean absolute error (MAE). These metrics were used to analyse the effectiveness of the four methods investigated in this study: balanceMSE, biasg, MSE and RankN.

Methods	balanceMSE				biasg			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	0.3505	29.5605	5.437	4.1579	0.3321	30.1566	5.4915	4.1206
Bacillus_subtilis_168_lysate_R1	0.2232	25.9268	5.0918	3.901	0.3015	26.9116	5.1876	4.2273
Caenorhabditis_elegans_lysate	0.368	22.4169	4.7346	3.6779	0.4724	20.5988	4.5386	3.5272
Danio_rerio_Zenodo_lysate	-0.2545	72.4703	8.5129	6.3495	0.3364	48.9019	6.993	5.3776
Drosophila_melanogaster_SII_lysate	0.3374	10.9702	3.3121	2.6325	0.4173	10.1633	3.188	2.5342
Ecoli_Lysate	0.382	40.7093	6.3804	5.0832	0.5134	31.3316	5.5975	4.5145
Escherichia_coli_cells	0.5293	44.9233	6.7025	5.6516	0.5683	34.6805	5.889	4.8422
Geobacillus_stearothermophilus_NCA26_lysate	0.1632	39.2467	6.2647	5.0332	0.2924	31.5223	5.6145	4.5322
HAOEC	0.6825	16.6991	4.0865	2.9802	0.6785	16.2852	4.0355	3.166
HEK293T	0.6503	10.8033	3.2868	2.6429	0.5679	12.6298	3.5538	2.8093
HL60	0.663	9.4643	3.0764	2.4607	0.663	9.4511	3.0743	2.5043
HaCaT	-0.0057	47.7258	6.9084	5.1539	-0.3245	48.5773	6.9697	5.5694
HepG2	0.6987	16.7001	4.0866	3.242	0.7914	14.4484	3.8011	3.0977
Jurkat	0.444	23.4916	4.8468	3.7271	0.6486	16.1099	4.0137	3.2487
K562	-1	11.0728	3.3276	2.8619	-1	3.0531	1.7473	1.3394
Mus_musculus_BMDC_lysate	0.3702	10.9503	3.3091	2.3	0.286	11.6081	3.4071	2.3945
Mus_musculus_liver_lysate	0.2139	26.6532	5.1627	4.2005	-0.0108	31.4768	5.6104	4.6453
Oleispira_antarctica_RB-8_lysate_R1	0.035	24.5496	4.9548	3.5839	0.1266	16.7649	4.0945	3.0388
Picrophilus_torridus_DSM9790_lysate	0.2688	29.8296	5.4617	4.3719	0.0523	27.8212	5.2746	4.1684
Saccharomyces_cerevisiae_lysate	0.3152	15.6218	3.9524	3.1088	0.2702	16.1805	4.0225	3.0964
Thermus_thermophilus_HB27_cells	0.1111	74.9102	8.6551	7.2418	0.0171	55.0932	7.4225	5.989
Thermus_thermophilus_HB27_lysate	0.0642	45.6323	6.7552	5.4227	0.3609	38.8648	6.2342	5.0251
U937	0.6794	12.4133	3.5232	2.9262	0.6765	9.8029	3.131	2.6008
colon_cancer_spheroids	0.1879	16.9934	4.1223	3.4596	-0.1437	18.6022	4.313	3.4689
pTcells	0.5	15.8435	3.9804	2.8804	-0.5	14.422	3.7976	3.5517
MEANS	0.279104	27.8231	5.03728	3.96205	0.255752	23.8183	4.6801	3.73558

Methods	MSE				rankN			
	Spearman	MSE	RMSE	MAE	Spearman	MSE	RMSE	MAE
Arabidopsis_thaliana_seedling_lysate	0.3461	29.1147	5.3958	4.1742	0.1444	33.0563	5.7495	4.4886
Bacillus_subtilis_168_lysate_R1	0.1653	25.9705	5.0961	4.0067	0.1319	25.9661	5.0957	3.9435
Caenorhabditis_elegans_lysate	0.499	20.6519	4.5444	3.4613	0.4784	21.4948	4.6362	3.4993
Danio_rerio_Zenodo_lysate	0.0455	109.125	10.4463	8.6692	0.0364	193.913	13.9253	12.4161
Drosophila_melanogaster_SII_lysate	0.3202	12.096	3.4779	2.7011	0.3287	10.5318	3.2453	2.5795
Ecoli_Lysate	0.4779	31.9304	5.6507	4.5741	0.5269	33.6137	5.7977	4.6787
Escherichia_coli_cells	0.5883	31.4554	5.6085	4.7336	0.5132	65.2721	8.0791	6.593
Geobacillus_stearothermophilus_NCA26_lysate	0.1593	37.1683	6.0966	5.0318	0.0914	33.9548	5.8271	4.8257
HAOEC	0.6695	16.8059	4.0995	3.148	0.6921	18.7967	4.3355	3.1634
HEK293T	0.4082	15.5122	3.9386	3.1006	0.5787	12.4972	3.5351	2.6919
HL60	0.6704	7.8528	2.8023	2.3577	0.6633	12.8484	3.5845	2.8722
HaCaT	0.4248	26.0739	5.1063	4.0582	-0.1043	150.318	12.2604	9.0783
HepG2	0.6422	20.104	4.4837	3.5313	0.7702	18.014	4.2443	3.3321
Jurkat	0.558	18.3289	4.2812	3.419	0.6849	23.2359	4.8204	3.8469
K562	1	42.885	6.5487	6.3414	-1	25.5311	5.0528	4.0778
Mus_musculus_BMDC_lysate	0.3303	11.1626	3.3411	2.47	0.1396	12.6878	3.562	2.5788
Mus_musculus_liver_lysate	0.1039	28.9951	5.3847	4.4282	0.2819	30.818	5.5514	4.3435
Oleispira_antarctica_RB-8_lysate_R1	0.0069	19.2697	4.3897	3.3043	0.2918	15.99	3.9988	2.9648
Picrophilus_torridus_DSM9790_lysate	0.2891	26.4809	5.146	4.1468	0.1515	137.6	11.7303	10.1876
Saccharomyces_cerevisiae_lysate	0.3907	15.1299	3.8897	2.9698	0.2702	16.3817	4.0474	3.1665
Thermus_thermophilus_HB27_cells	0.1968	58.7222	7.663	6.4591	0.1177	47.7578	6.9107	5.6658
Thermus_thermophilus_HB27_lysate	-0.1086	45.6133	6.7538	5.5957	0.0697	35.6751	5.9729	4.9608
U937	0.5059	12.1784	3.4898	3.0624	0.1618	29.4041	5.4226	4.2814
colon_cancer_spheroids	0.0397	18.7022	4.3246	3.4427	0.189	16.633	4.0784	3.201
pTcells	-0.5	9.1506	3.025	2.8475	0.5	198.42	14.0862	13.8163
MEANS	0.329176	27.6192	4.99936	4.08139	0.268376	48.8165	6.22198	5.09014

Table 11: Assessment of methodologies using the combination of ESM2 and PiFold embeddings: The performance of individual models was assessed for each species and subsequently tested on that target species inside the FLIP partition. Spearman correlation, mean square error (MSE), root mean square error (RMSE) and mean absolute error (MAE). These metrics were used to analyse the effectiveness of the four methods investigated in this study: balanceMSE, biasg, MSE and RankN.

E Cross-species Prediction Scatterplots - Global Model Applied to Individual Species

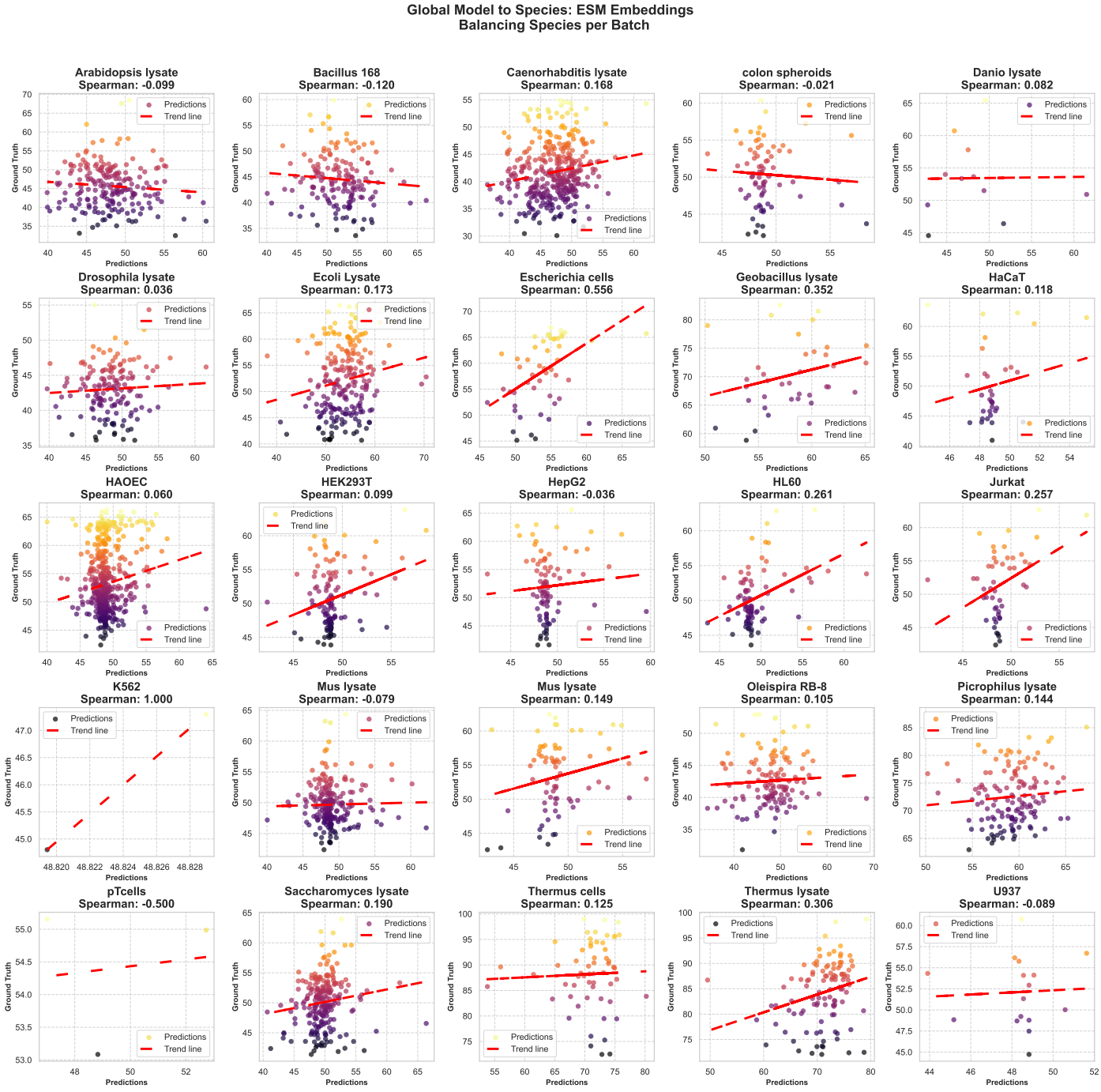

Figure 7: Scatterplot of global model applied to each species using balancing strategy per batch and ESM embeddings

Figure 8: Scatterplot of global model applied to each species using dual loss function and ESM embeddings

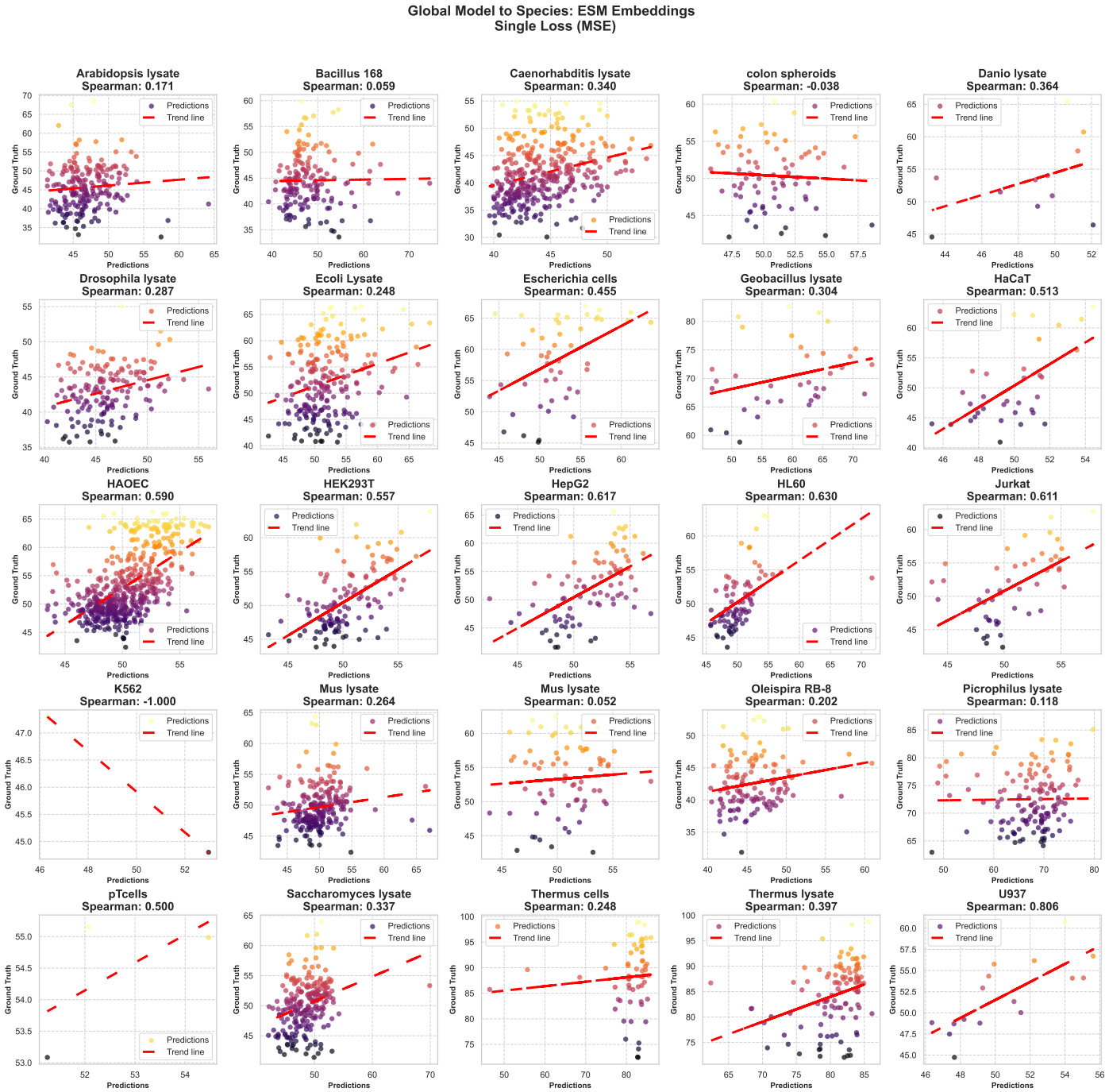

Figure 9: Scatterplot of global model applied to each species using MSE as loss function and ESM embeddings

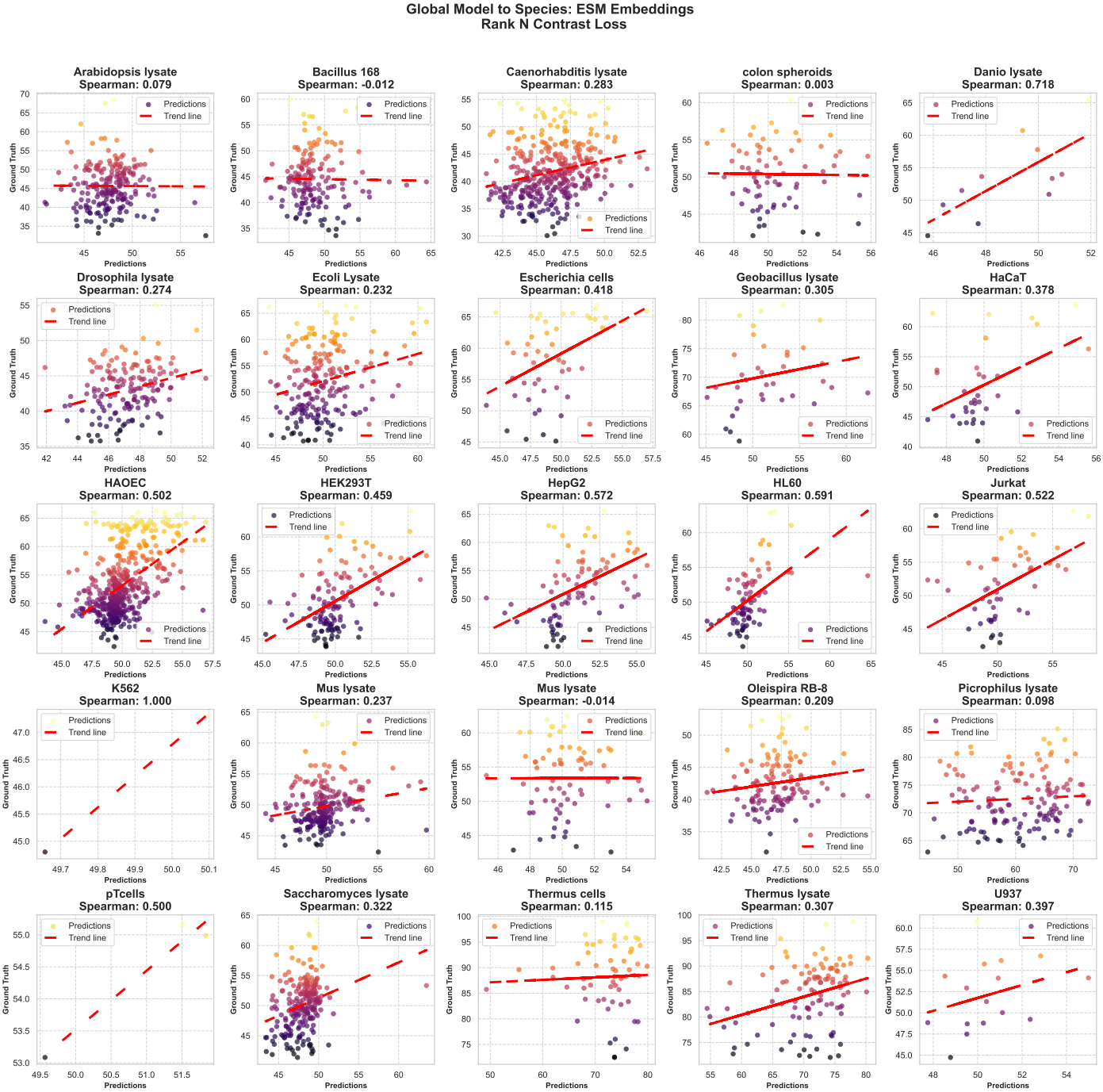

Figure 10: Scatterplot of global model applied to each species using Rank N Contrast loss and ESM embeddings

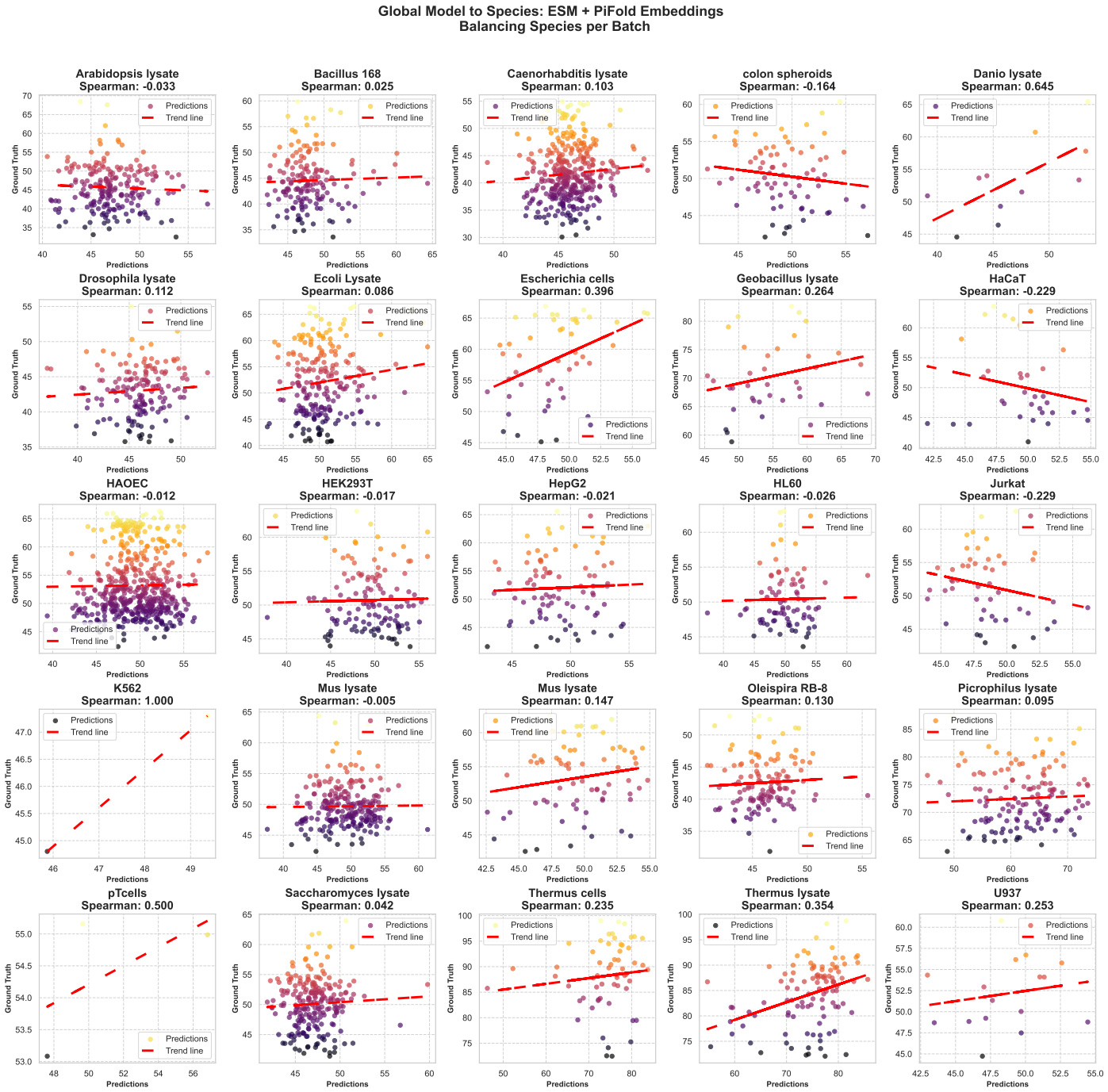

Figure 11: Scatterplot of global model applied to each species using balancing strategy per batch in combination with ESM and PiFold embeddings

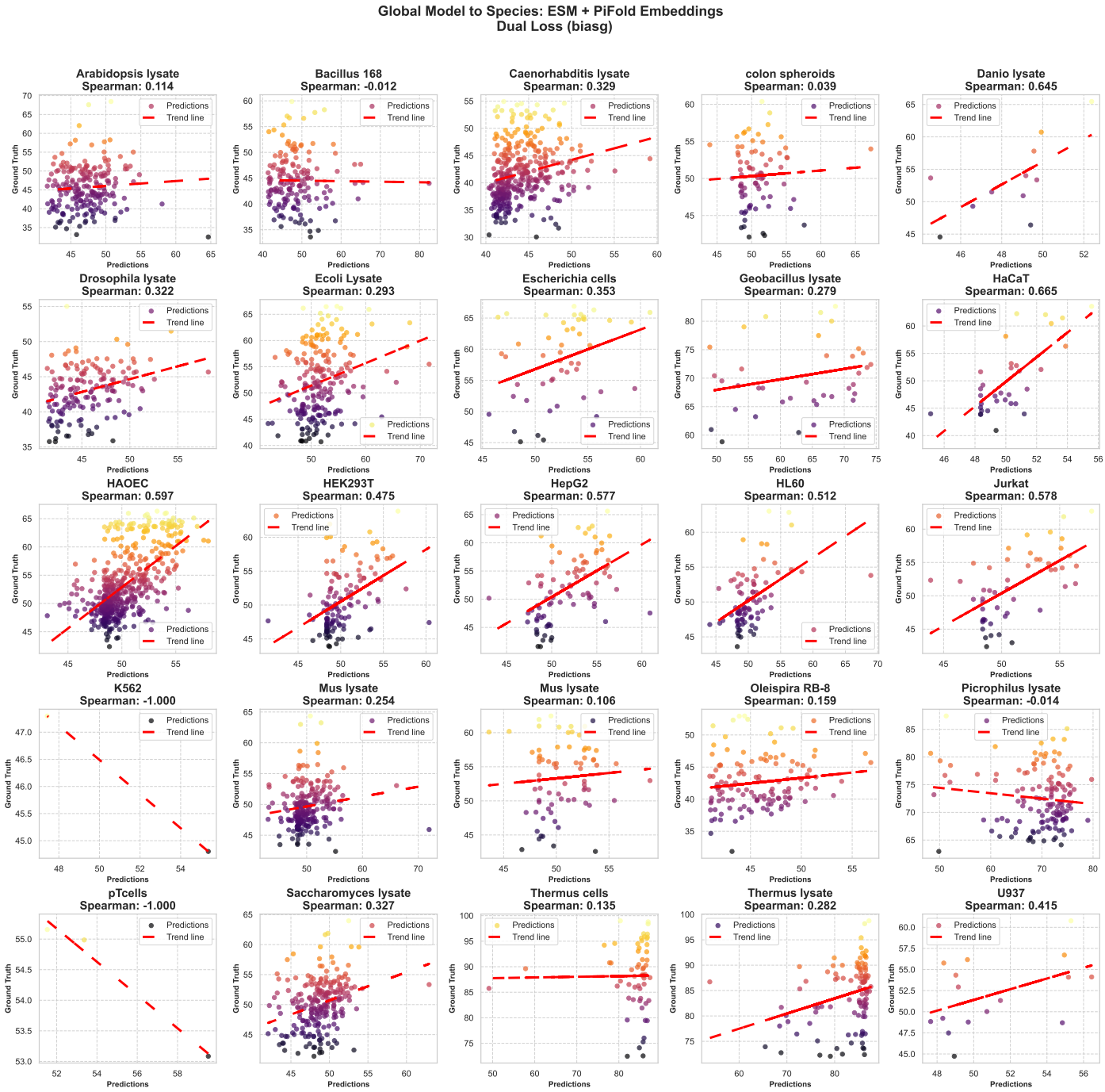

Figure 12: Scatterplot of global model applied to each species using dual loss function in combination with ESM and PiFold embeddings

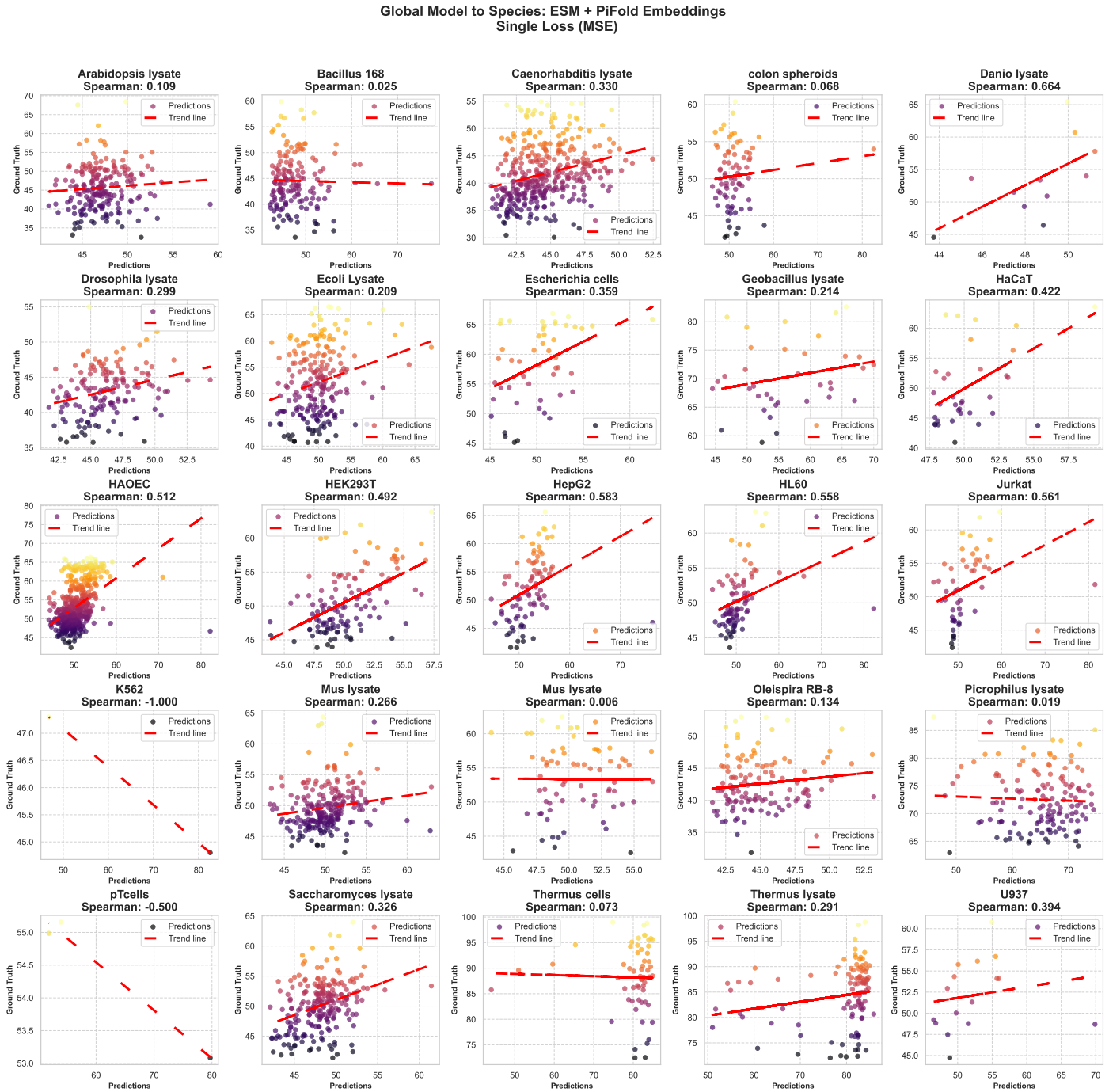

Figure 13: Scatterplot of global model applied to each species using MSE as loss function in combination with ESM and PiFold embeddings

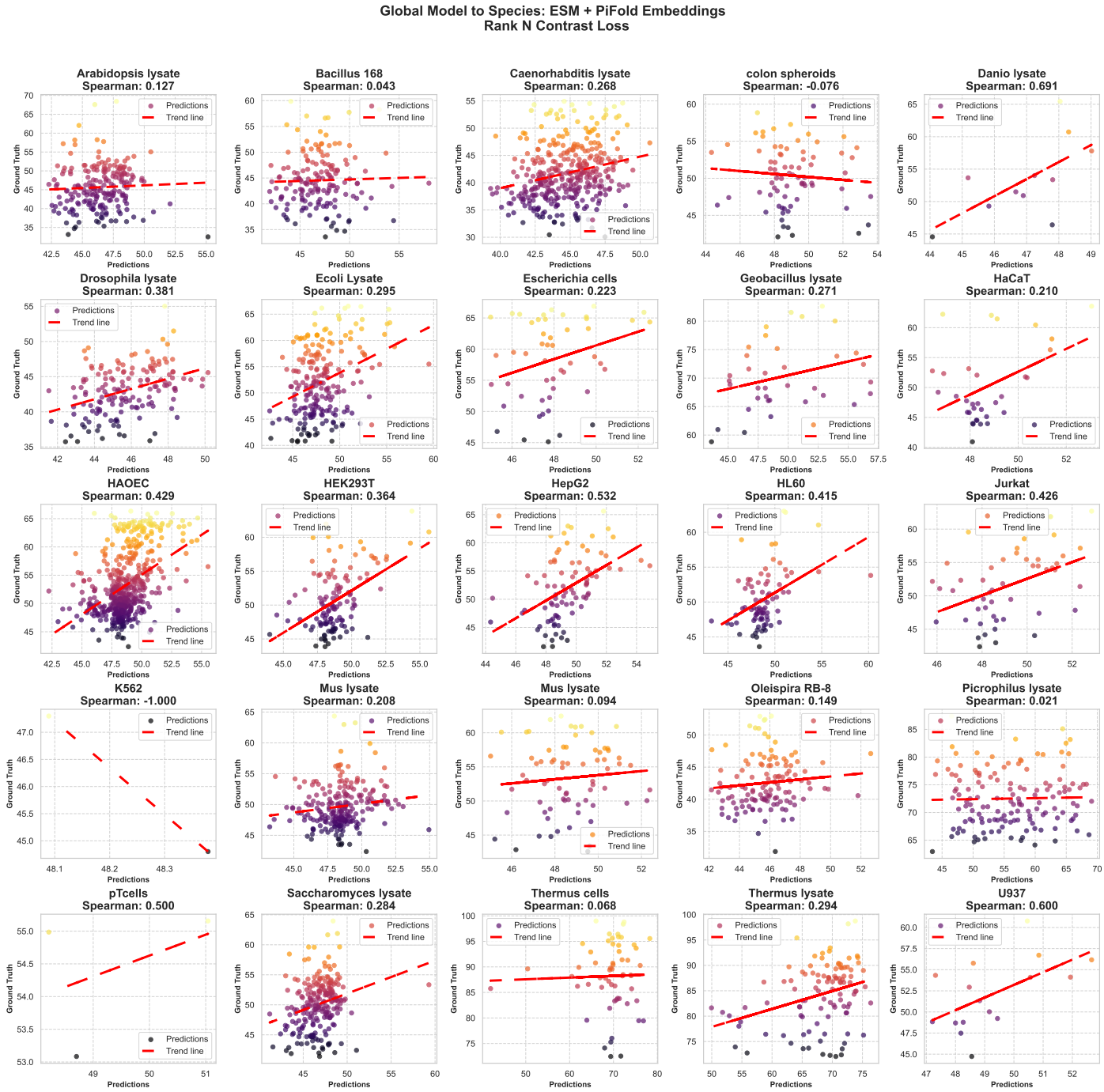

Figure 14: Scatterplot of global model applied to each species using Rank N Contrast loss in combination with ESM and PiFold embeddings

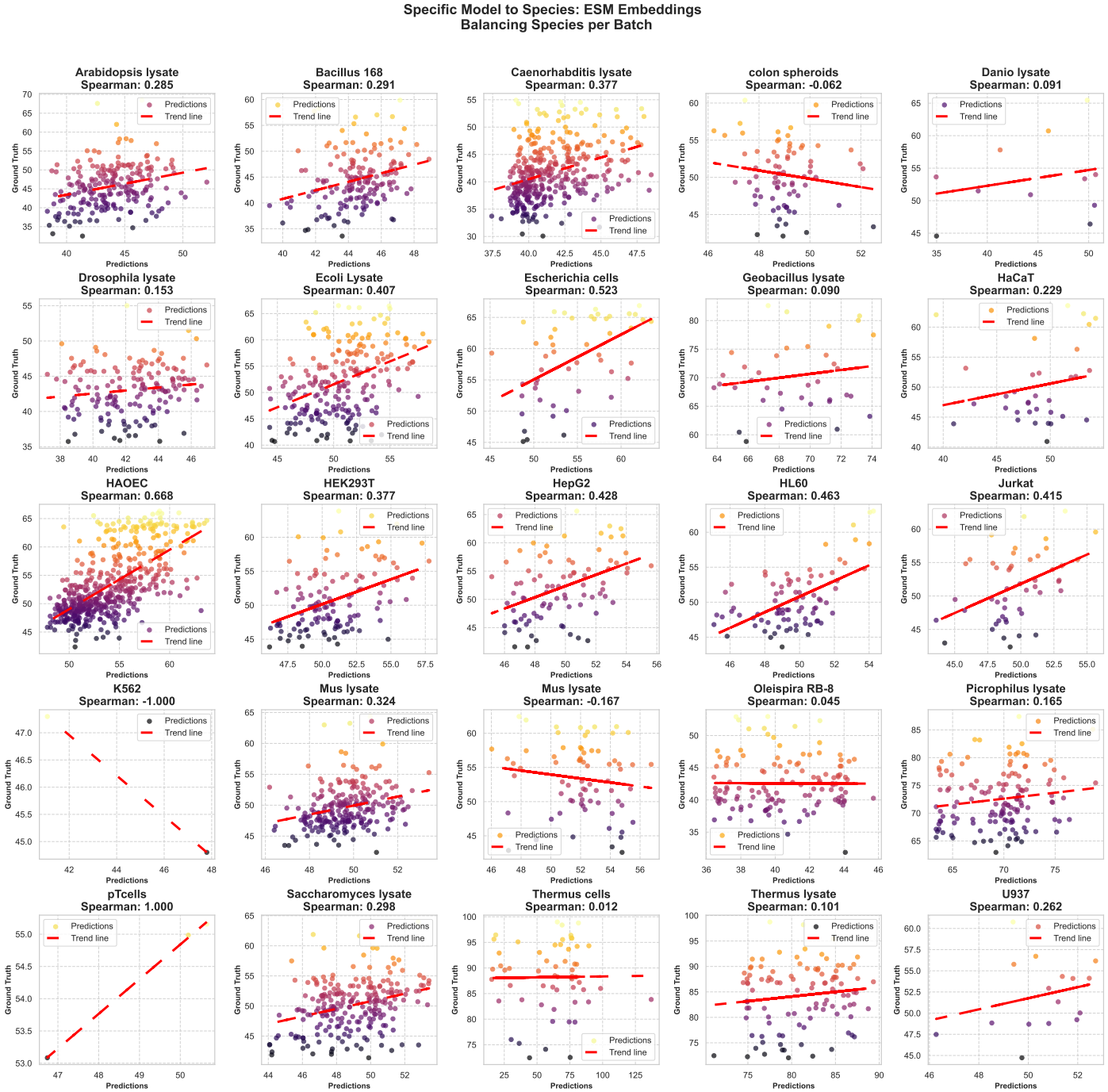

Figure 15: Scatterplot of individual models applied to their target species using balancing strategy per batch with ESM embeddings

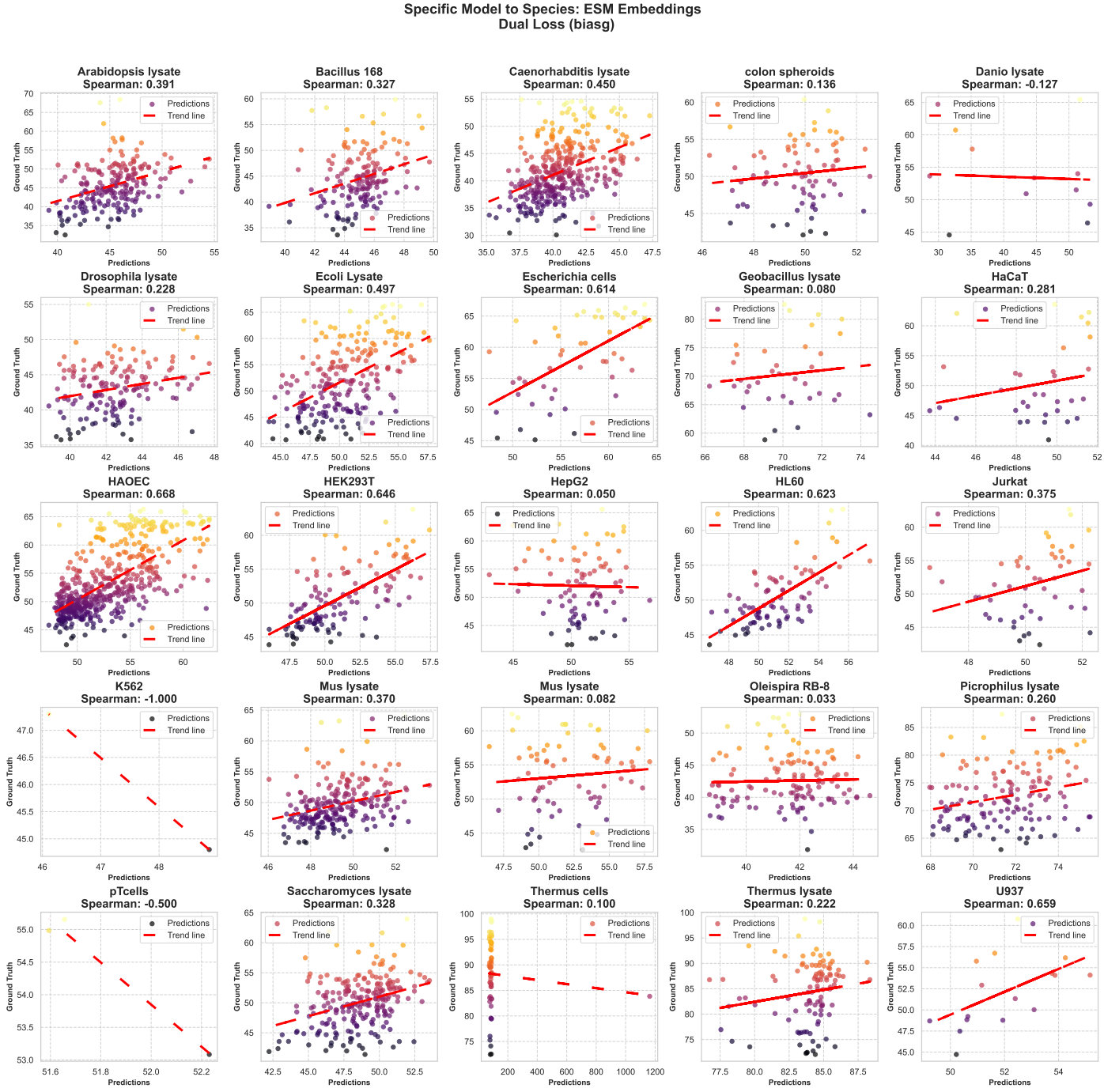

Figure 16: Scatterplot of individual models applied to their target species using dual loss function with ESM embeddings

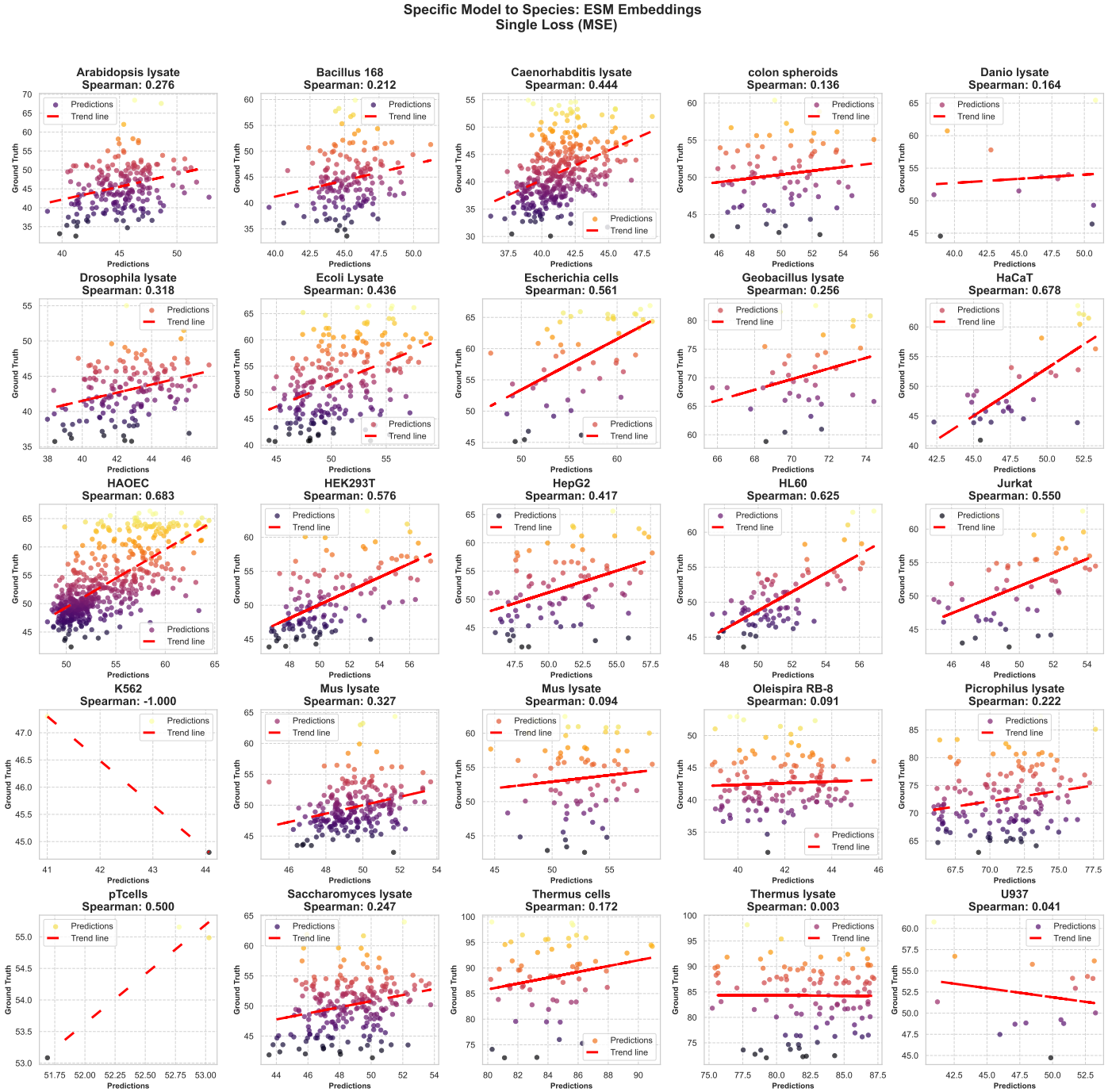

Figure 17: Scatterplot of individual models applied to their target species using MSE as loss function with ESM embeddings

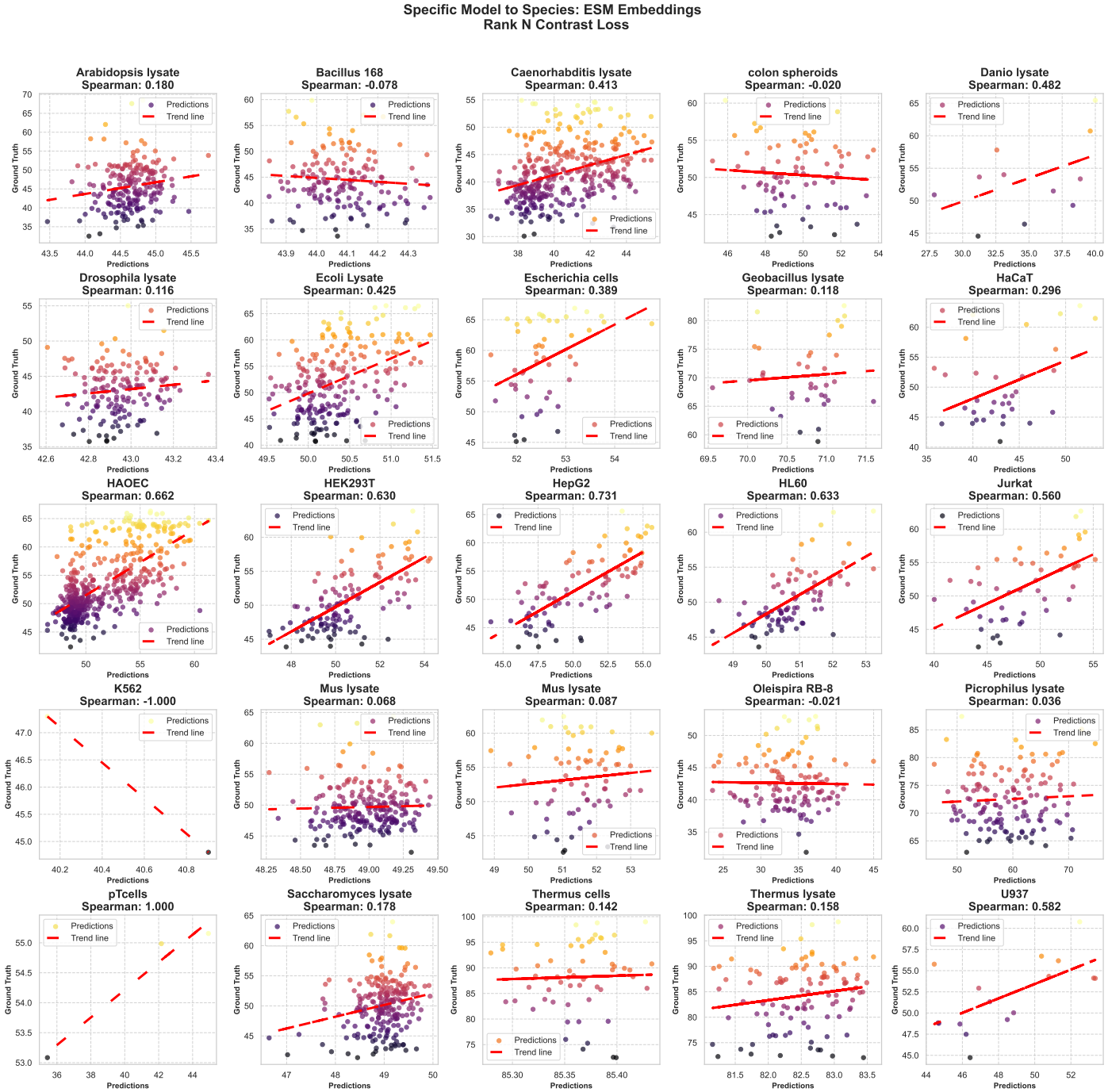

Figure 18: Scatterplot of individual models applied to their target species using Rank N Contrast loss with ESM embeddings

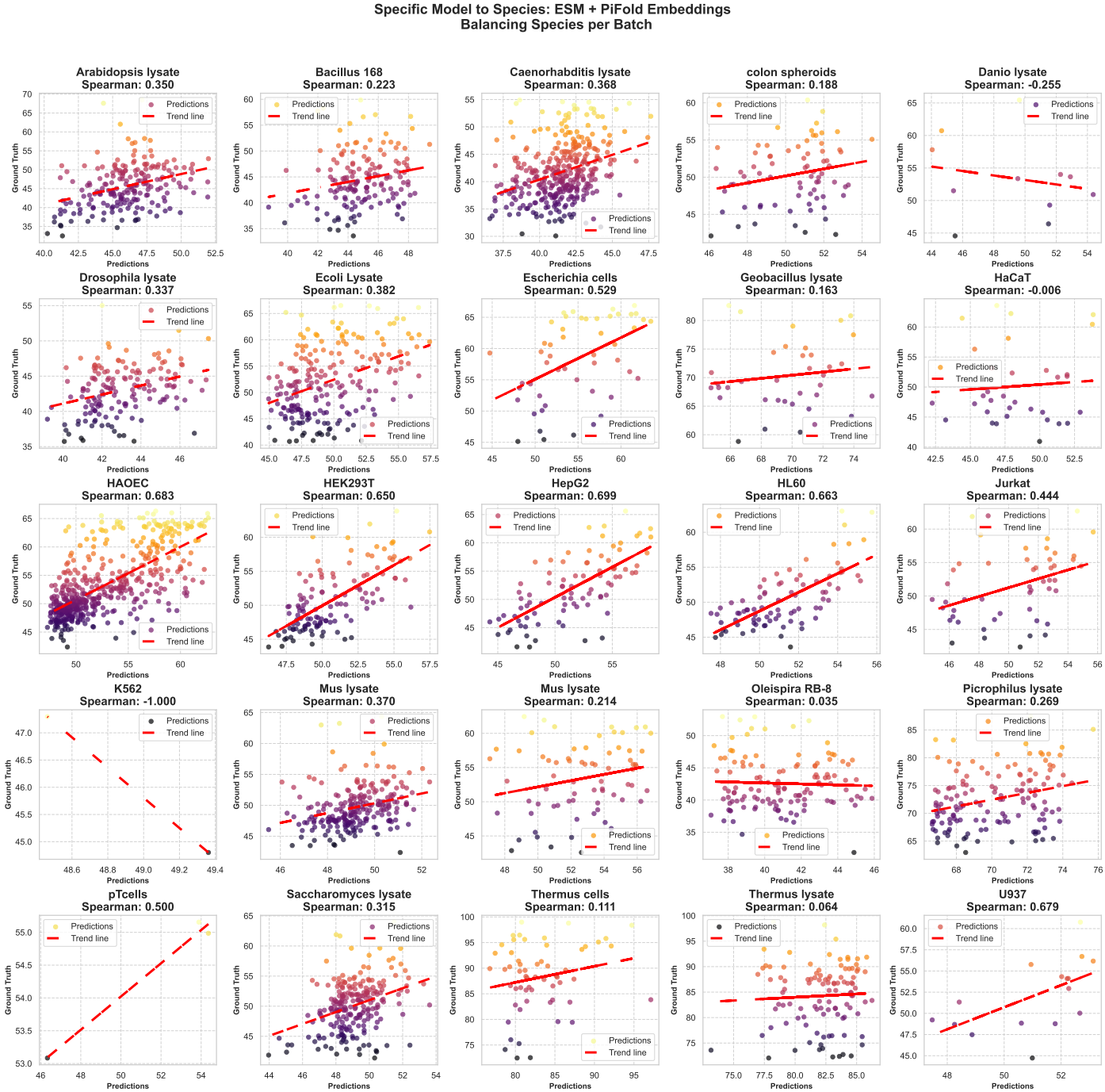

Figure 19: Scatterplot of individual models applied to their target species using balancing strategy per batch in combination with ESM and PiFold embeddings

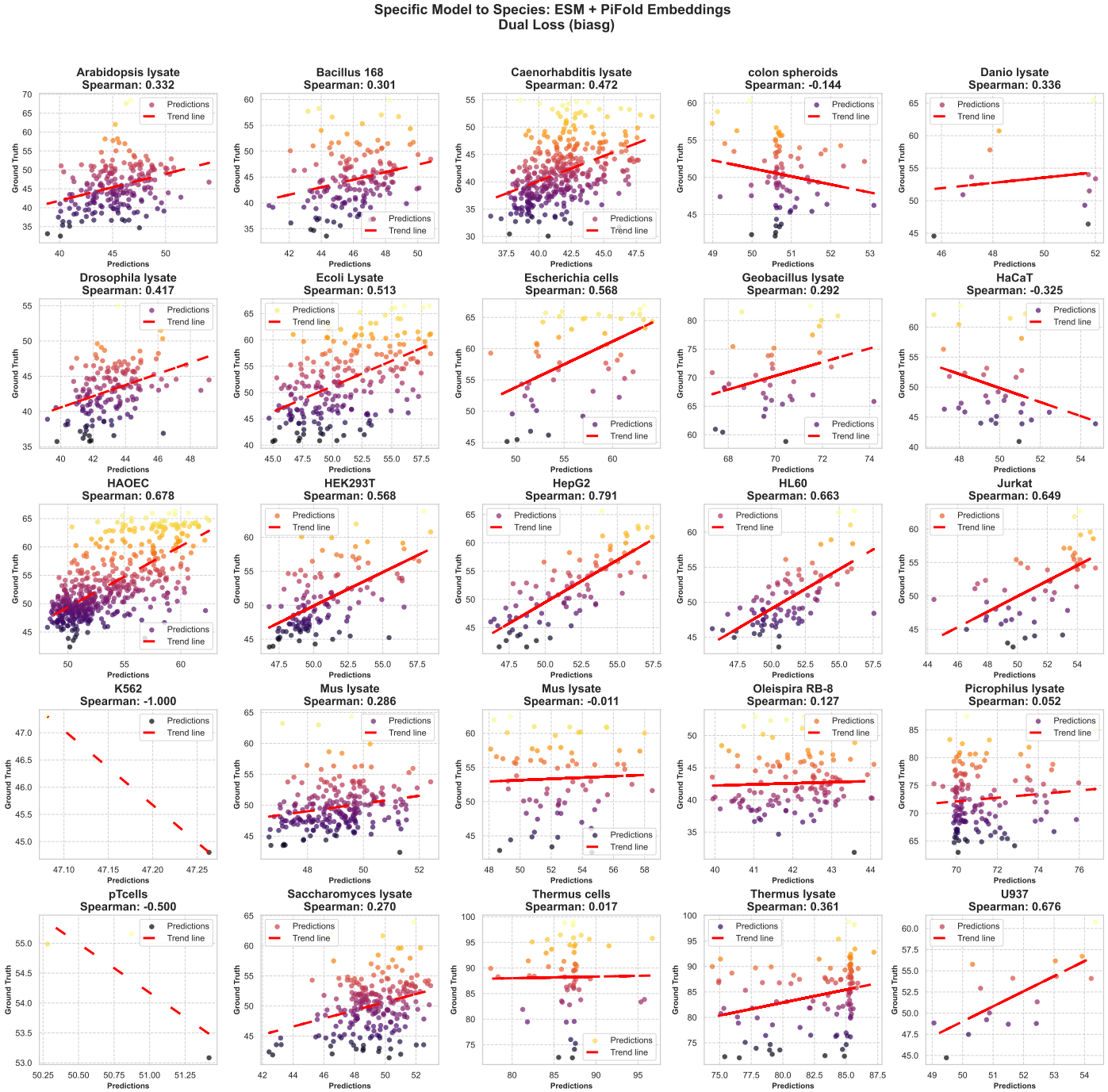

Figure 20: Scatterplot of individual models applied to their target species using dual loss function in combination with ESM and PiFold embeddings

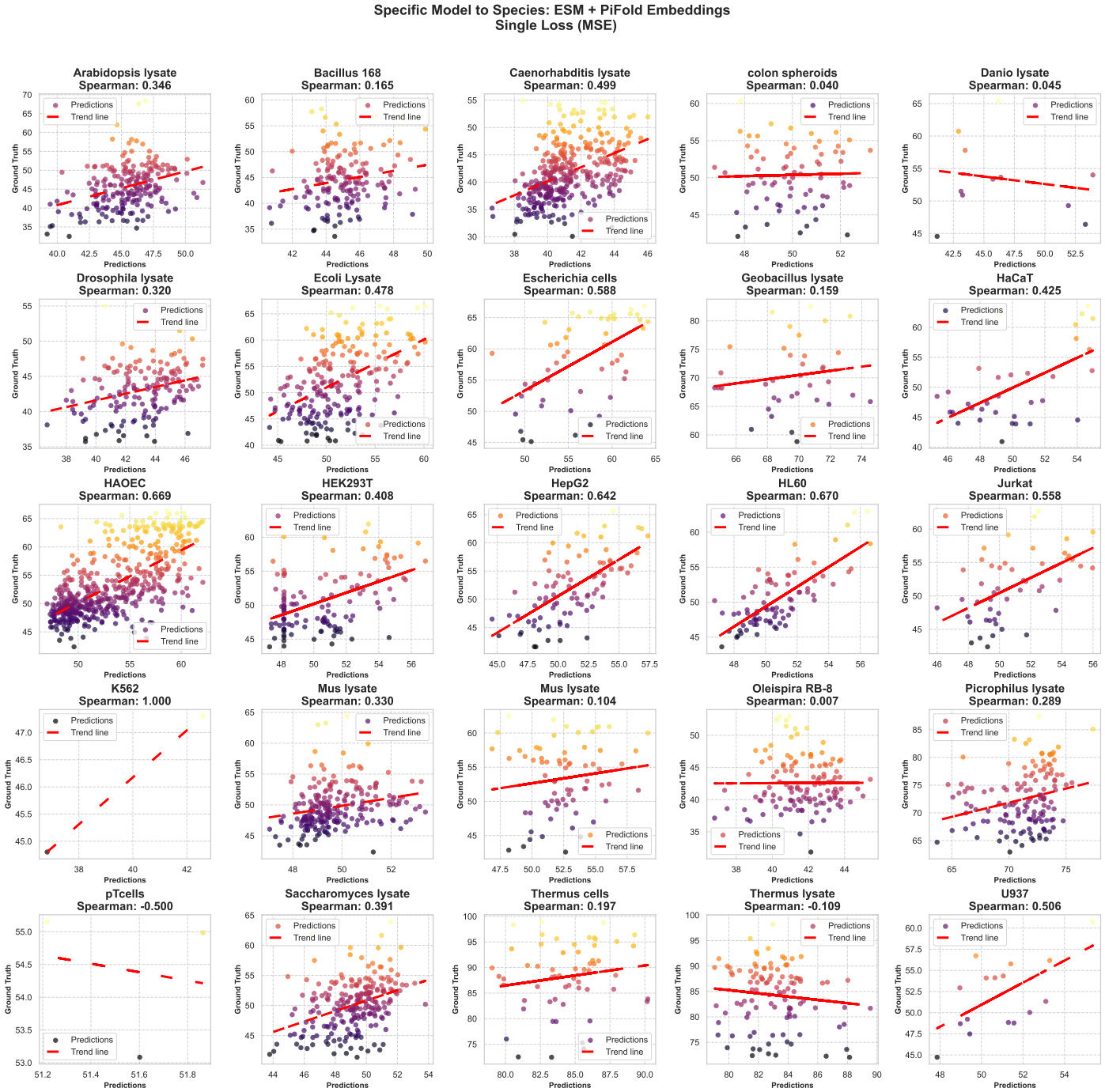

Figure 21: Scatterplot of individual models applied to their target species using MSE as loss function in combination with ESM and PiFold embeddings

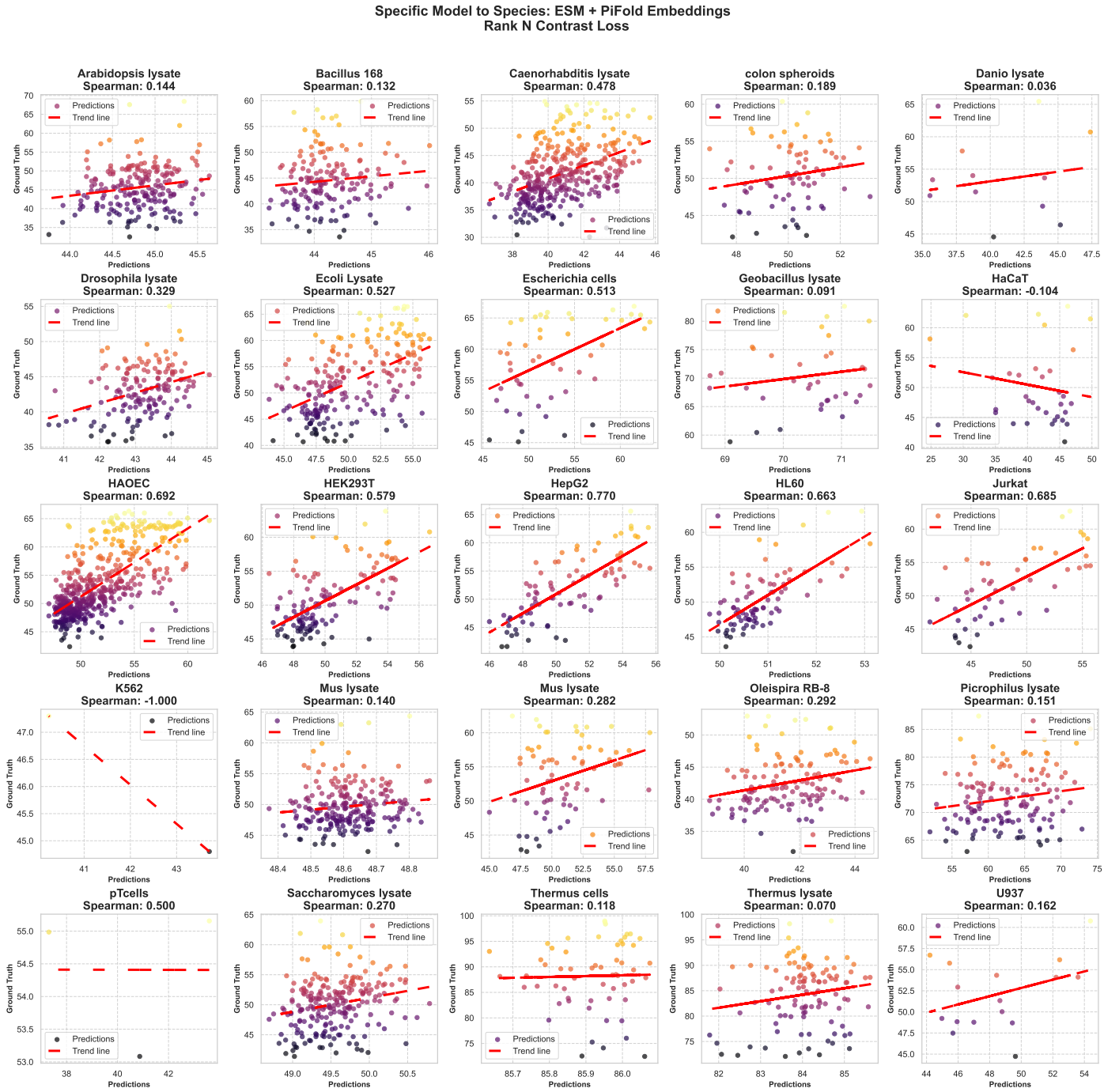

Figure 22: Scatterplot of individual models applied to their target species using Rank N Contrast loss in combination with ESM and PiFold embeddings

Chapter 5

Summary and Discussion

We begin this section by noting that, despite significant advances in biology through the use of representations from both embedding-based and MSA-based approaches to machine learning, there is still no clear consensus as to which representation is superior.

It is true that the extensive use of embeddings as a representation has undoubtedly revolutionised biology, particularly with the incorporation of protein language models [4], [2]. These models have very attractive properties, so we have a natural tendency to want to select them to obtain features for our modelling: their scalability due to the wide range of proteins used for training [3], [4], [8]; Their strong generalisation capabilities and implicitly capture of evolutionary relationships due precisely the amount of sequences in which they have been trained [3], [8], [35]; their easy way to be coupled with other types of representation [77], and also the flexibility to use transfer knowledge for downstream tasks in many other subfields like protein modelling and protein engineering [77], amoung others. Nevertheless, MSA-based representations remain relevant and continue to play an important role in several applications.

While protein language models have become very dominant in various applications in bioinformatics and computational biology, MSA is still relevant in a number of applications: for example, MSA are still widely used in variant effect prediction (VEP), where the focus is on assessing the impact of mutations or variants in proteins, how this affects their function and the subsequent impact on biological processes, which is very useful in determin-

ing which mutations may be associated with diseases [85], [32], [86]. In this particular application, VEP, it is clear that if we have a lot of information about protein families or homologous sequences, then the use of MSA can be a very convenient way of representing that knowledge. This knowledge can then be encapsulated or condensed into a Variational Autoencoder, which could be a very good predictor for capturing correlations between residues [32], [87]. The use of MSA also remains relevant as it has had a direct impact on protein folding algorithms such as Alphafold2 [1]; residue interaction tools such as EVcoupling [28]; and even the use of language models that operate on sets of MSAs such as MSAtransformer for unsupervised contact prediction and other related applications [35]. Likewise, MSA continue to be highly relevant in protein design, particularly when used in directed evolution and protein fitness frameworks [87], [86], [88], while maintaining their importance in state-of-the-art techniques.

That said, the takeaway message is that both embedding-based and MSA-based representations are powerful ways to represent, but it will largely depend on the type of application we are targeting. In this regard, it makes sense to think that generating contributions in both representation schemes is the way to go.

The first contribution of the project, as presented in chapter 3, demonstrates as a proof of concept that it is possible to perform MSA using geometric means such as a parametric spatial transformations. These transformations, specifically, the adaptation of CPAB transformations that we proposed, can be easily incorporated into probabilistic frameworks, such as densities, which allow probabilistic modelling for inferring sequence alignments. The use of probabilistic methods, particularly those based on density estimation, offers the advantage of quantifying uncertainty in the modelling process. In this context, there is the potential to assess uncertainty within alignments where this uncertainty is derived from the model itself rather than resulting from an end-to-end solution. Our approach is consistent with the objectives outlined in chapter 3. Another important contribution is the demonstration of an alternative method for sequence alignment that does not rely on the Markov assumptions. Instead, our model works by shifting between vertices representing residue positions, relying on transformations derived from the spatial transformation block, rather than relying on previous states as HMM approaches [7] do. Among other contributions, we demonstrated the capa-

bilities of spatial transformations to generalize alignments. By estimating the distributions of optimal transformation parameters across homologous sequences via PGMs, we can leverage this knowledge to guide the alignment of new sequences. This approach is analogous to the functionality provided by tools like HMMER [89].

Despite the contributions presented in the work derived from Chapter 3, it is important to emphasize that this remains a proof of concept. Although the results are promising, further experimentation is required. For example, although one potential contribution is the use of VAEs as informative priors for aligned sequences within the graphical model that we have proposed to infer the optimal transformation for sequence alignment, this approach may be a double-edged sword. In this configuration, the success of the alignment is highly dependent on how well the prior has been trained to guide the alignments. Another aspect that requires further experimentation is the configuration used to estimate the alignments. We currently use an informative prior, but the case where the entire graphical model is estimated from scratch has not yet been explored. As described in the methodological framework for dealing with long-range dependencies in chapter 3, the inclusion of a new latent variable Ψ does not affect the property of the PGM of being a Directed Acyclic Graph (DAG) when we use the prior. However, if the prior is omitted and the model is trained from scratch, a cyclic dependency would be created in the DAG, potentially complicating the training process in theory. Therefore, alternative training approaches should be considered. The remaining experiments are ongoing and continue to be conducted as future work, but so far the proposal has proved the initial research questions set out in Chapter 1.1.2.1 and Chapter 3.

Regarding the second contribution of the project, presented in Chapter 4, we demonstrated that when predicting melting temperatures in the presence of a significant imbalance in data distribution across species, there is a strong bias toward species with more abundant information. This, however, negatively impacts the prediction performance for underrepresented species. In light of such data imbalance in regression tasks, a more practical approach would be to train species-specific models rather than relying on a global model. However, training individual models for each species is insufficient, as embeddings derived solely from protein language models (pLMs) tend to have high variance in small datasets. It is therefore essential to integrate/combine

structural features, such as those provided by embeddings from inverse folding models, and ideally combine them with contrastive methods to optimize the models. These approaches have yielded the best results in the work presented in chapter 4.

In addition to the points mentioned above, it is important to note that many proposals for melting temperature prediction, such as those in [60] and [61], have relied primarily on Spearman correlation as the primary evaluation metric. However, we show that this measure can be misleading when assessing specific-species predictions. Therefore, in the context of melting temperature prediction, it is essential to complement Spearman correlation with other metrics, such as mean squared error (MSE), to obtain a more accurate assessment of model performance. Furthermore, we show that combining features from protein language models with inverse folding, alongside a simple contrastive method, such as Dual Loss versus Rank N Contrast Loss [71], can yield significant improvements. However, it is important to note that because Rank N Contrast Loss relies on the type of transformations used in its data augmentation process, several factors remain to be explored. Future work could investigate the choice of similarity measures for improving this methodology as well as the choice of transformations for the same component. In general, the overall contribution of the work presented in Chapter 4 is oriented towards the analysis of thermostability in the context of melting temperatures at the species level and the phenomena associated with this application. Something that has not been done previously.

List of Publications

The work presented in this thesis has lead to the following publications.

1. Sebastián García López, Søren Hauberg, and Wouter Boomsma. Probabilistic multiple sequence alignment using spatial transformations. *bioRxiv*, 2024. Preprint.

2. Sebastián García López, Jesper Salomon, and Wouter Boomsma. Cross-species vs species-specific models for protein melting temperature prediction. *bioRxiv*, 2024. Preprint.

Note At the time of submission of this thesis, both manuscripts had been uploaded to BioRxiv and are expected to be available online soon, though a DOI has not yet been assigned.

Bibliography

[1] John Jumper, Richard Evans, Alexander Pritzel, Tim Green, Michael Figurnov, Olaf Ronneberger, Kathryn Tunyasuvunakool, Russ Bates, Augustin Žídek, Anna Potapenko, et al. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[2] Jesse Vig, Ali Madani, Lav R Varshney, Caiming Xiong, Richard Socher, and Nazneen Fatema Rajani. Bertology meets biology: Interpreting attention in protein language models. *arXiv preprint arXiv:2006.15222*, 2020.

[3] Alexander Rives, Joshua Meier, Tom Sercu, Siddharth Goyal, Zeming Lin, Jason Liu, Demi Guo, Myle Ott, C Lawrence Zitnick, Jerry Ma, et al. 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.

[4] A. Elnaggar, M. Heinzinger, C. Dallago, and et al. Protbert: A universal language model for protein sequence representation and analysis. *Bioinformatics*, 37(12):1707–1716, 2021.

[5] Debora S Marks, Lucy J Colwell, Robert Sheridan, Thomas A Hopf, Andrea Pagnani, Riccardo Zecchina, and Chris Sander. Protein 3d structure computed from evolutionary sequence variation. *PloS one*, 6(12):e28766, 2011.

[6] Debora S Marks, Thomas A Hopf, and Chris Sander. Protein structure prediction from sequence variation. *Nature biotechnology*, 30(11):1072–1080, 2012.

[7] Anders Krogh, Michael Brown, I Saira Mian, Kimmen Sjölander, and David Haussler. Hidden markov models in computational biol-

ogy: Applications to protein modeling. *Journal of molecular biology*, 235(5):1501–1531, 1994.

[8] Ahmed Elnaggar, Michael Heinzinger, Christian Dallago, Ghalia Rehawi, Yu Wang, Llion Jones, Tom Gibbs, Tamas Feher, Christoph Angerer, Martin Steinegger, et al. Prottrans: Toward understanding the language of life through self-supervised learning. *IEEE transactions on pattern analysis and machine intelligence*, 44(10):7112–7127, 2021.

[9] Chloe Hsu, Robert Verkuil, Jason Liu, Zeming Lin, Brian Hie, Tom Sercu, Adam Lerer, and Alexander Rives. Learning inverse folding from millions of predicted structures. In *International conference on machine learning*, pages 8946–8970. PMLR, 2022.

[10] Zhangyang Gao, Cheng Tan, Pablo Chacón, and Stan Z Li. Pifold: Toward effective and efficient protein inverse folding. *arXiv preprint arXiv:2209.12643*, 2022.

[11] David De Juan, Florencio Pazos, and Alfonso Valencia. Emerging methods in protein co-evolution. *Nature Reviews Genetics*, 14(4):249–261, 2013.

[12] Ulrike Göbel, Chris Sander, Reinhard Schneider, and Alfonso Valencia. Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Bioinformatics*, 18(4):309–317, 1994.

[13] IN Shindyalov, NA Kolchanov, and Chris Sander. Can three-dimensional contacts in protein structures be predicted by analysis of correlated mutations? *Protein Engineering, Design and Selection*, 7(3):349–358, 1994.

[14] John A Capra and Mona Singh. Predicting functionally important residues from sequence conservation. *Bioinformatics*, 23(15):1875–1882, 2007.

[15] Stephen F Altschul, Thomas L Madden, Alejandro A Schäffer, Jinghui Zhang, Zheng Zhang, Webb Miller, and David J Lipman. Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic acids research*, 25(17):3389–3402, 1997.

[16] Chern-Sing Goh, Andrew A Bogan, Marcin Joachimiak, Dirk Walther, and Fred E Cohen. Co-evolution of proteins with their interaction partners. *Journal of molecular biology*, 299(2):283–293, 2000.

[17] Burkhard Rost. Twilight zone of protein sequence alignments. *Protein engineering*, 12(2):85–94, 1999.

[18] Jimin Pei and Nick V Grishin. Al2co: calculation of positional conservation in a protein sequence alignment. *Bioinformatics*, 17(8):700–712, 2001.

[19] Faruck Morcos, Andrea Pagnani, Bryan Lunt, Arianna Bertolino, Debora S Marks, Chris Sander, Riccardo Zecchina, José N Onuchic, Terence Hwa, and Martin Weigt. Direct-coupling analysis of residue coevolution captures native contacts across many protein families. *Proceedings of the National Academy of Sciences*, 108(49):E1293–E1301, 2011.

[20] Alan Lapedes, Bertrand Giraud, and Christopher Jarzynski. Using sequence alignments to predict protein structure and stability with high accuracy. *arXiv preprint arXiv:1207.2484*, 2012.

[21] Lukas Burger and Erik Van Nimwegen. Accurate prediction of protein–protein interactions from sequence alignments using a bayesian method. *Molecular systems biology*, 4(1):165, 2008.

[22] Lukas Burger and Erik Van Nimwegen. Disentangling direct from indirect co-evolution of residues in protein alignments. *PLoS computational biology*, 6(1):e1000633, 2010.

[23] David T Jones, Daniel WA Buchan, Domenico Cozzetto, and Massimiliano Pontil. Psicov: precise structural contact prediction using sparse inverse covariance estimation on large multiple sequence alignments. *Bioinformatics*, 28(2):184–190, 2012.

[24] Martin Weigt, Robert A White, Hendrik Szurmant, James A Hoch, and Terence Hwa. Identification of direct residue contacts in protein–protein interaction by message passing. *Proceedings of the National Academy of Sciences*, 106(1):67–72, 2009.

[25] Thomas A Hopf, Lucy J Colwell, Robert Sheridan, Burkhard Rost, Chris Sander, and Debora S Marks. Three-dimensional structures of membrane proteins from genomic sequencing. *Cell*, 149(7):1607–1621, 2012.

[26] Joanna I Sułkowska, Faruck Morcos, Martin Weigt, Terence Hwa, and José N Onuchic. Genomics-aided structure prediction. *Proceedings of the National Academy of Sciences*, 109(26):10340–10345, 2012.

[27] William R Taylor, David T Jones, and Michael I Sadowski. Protein topology from predicted residue contacts. *Protein Science*, 21(2):299–305, 2012.

[28] Thomas A Hopf, Anna G Green, Benjamin Schubert, Sophia Mersmann, Charlotta PI Schärfe, John B Ingraham, Agnes Toth-Petroczy, Kelly Brock, Adam J Riesselman, Perry Palmedo, et al. The evcouplings python framework for coevolutionary sequence analysis. *Bioinformatics*, 35(9):1582–1584, 2019.

[29] Richard Durbin, Sean R Eddy, Anders Krogh, and Graeme Mitchison. *Biological sequence analysis: probabilistic models of proteins and nucleic acids*. Cambridge university press, 1998.

[30] Sean R Eddy. Biological sequence analysis probabilistic models of proteins and nucleic acids, 1998.

[31] Sean R. Eddy. Profile hidden markov models. *Bioinformatics (Oxford, England)*, 14(9):755–763, 1998.

[32] Adam J Riesselman, John B Ingraham, and Debora S Marks. Deep generative models of genetic variation capture the effects of mutations. *Nature methods*, 15(10):816–822, 2018.

[33] Samantha Petti, Nicholas Bhattacharya, Roshan Rao, Justas Dauparas, Neil Thomas, Juannan Zhou, Alexander M Rush, Peter Koo, and Sergey Ovchinnikov. End-to-end learning of multiple sequence alignments with differentiable smith–waterman. *Bioinformatics*, 39(1):btac724, 2023.

[34] Eli N Weinstein and Debora Marks. A structured observation distribution for generative biological sequence prediction and forecasting. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings*

of Machine Learning Research, pages 11068–11079. PMLR, 18–24 Jul 2021.

[35] Roshan M Rao, Jason Liu, Robert Verkuil, Joshua Meier, John Canny, Pieter Abbeel, Tom Sercu, and Alexander Rives. Msa transformer. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8844–8856. PMLR, 18–24 Jul 2021.

[36] Ginevra Carbone, Francesca Cuturello, Luca Bortolussi, and Alberto Cazzaniga. Adversarial attacks on protein language models. *bioRxiv*, pages 2022–10, 2022.

[37] Pola Elisabeth Schwöbel. *Learned Data Augmentation for Bias Correction*. PhD thesis, Technical University of Denmark, 2022.

[38] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. *Advances in neural information processing systems*, 28, 2015.

[39] Oren Freifeld, Søren Hauberg, Kayhan Batmanghelich, and Jonn W Fisher. Transformations based on continuous piecewise-affine velocity fields. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2496–2509, 2017.

[40] Nicki Skafte Detlefsen, Oren Freifeld, and Søren Hauberg. Deep diffeomorphic transformer networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4403–4412, 2018.

[41] Nicki Skafte and Søren Hauberg. Explicit disentanglement of appearance and perspective in generative models. *Advances in Neural Information Processing Systems*, 32, 2019.

[42] Pola Schwöbel, Frederik Rahbæk Warburg, Martin Jørgensen, Kristoffer Hougaard Madsen, and Søren Hauberg. Probabilistic spatial transformer networks. In *Uncertainty in Artificial Intelligence*, pages 1749–1759. PMLR, 2022.

[43] David M Blei, Alp Kucukelbir, and Jon D McAuliffe. Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877, 2017.

[44] Christopher M Bishop and Hugh Bishop. *Deep learning: Foundations and concepts*. Springer Nature, 2023.

[45] Diederik P Kingma and Max Welling. Auto-encoding variational bayes. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2014.

[46] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[47] Matthew J Johnson, David K Duvenaud, Alex Wiltschko, Ryan P Adams, and Sandeep R Datta. Composing graphical models with neural networks for structured representations and fast inference. *Advances in neural information processing systems*, 29, 2016.

[48] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

[49] Jungo Kasai, Hao Peng, Yizhe Zhang, Dani Yogatama, Gabriel Ilharco, Nikolaos Pappas, Yi Mao, Weizhu Chen, and Noah A Smith. Finetuning pretrained transformers into rnns. *arXiv preprint arXiv:2103.13076*, 2021.

[50] Narendra Patwardhan, Stefano Marrone, and Carlo Sansone. Transformers in the real world: A survey on nlp applications. *Information*, 14(4):242, 2023.

[51] Jacob Devlin. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[52] Alec Radford. Improving language understanding by generative pre-training. 2018.

[53] Joshua Meier, Roshan Rao, Robert Verkuil, Jason Liu, Tom Sercu, and Alexander Rives. Language models enable zero-shot prediction of the effects of mutations on protein function. *bioRxiv*, 2021.

[54] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Allan dos Santos Costa, Maryam Fazel-Zarandi, Tom Sercu, Sal Candido, et al. Language models of protein sequences at the scale of evolution enable accurate structure prediction. *bioRxiv*, 2022.

[55] Maxat Kulmanov and Robert Hoehndorf. Deepgoplus: improved protein function prediction from sequence. *Bioinformatics*, 36(2):422–429, 2020.

[56] Vladimir Gligorijević, P Douglas Renfrew, Tomasz Kosciolek, Julia Koehler Leman, Daniel Berenberg, Tommi Vatanen, Chris Chandler, Bryn C Taylor, Ian M Fisk, Hera Vlamakis, et al. Structure-based protein function prediction using graph convolutional networks. *Nature communications*, 12(1):3168, 2021.

[57] Deshan Zhou, Zhijian Xu, WenTao Li, Xiaolan Xie, and Shaoliang Peng. Multidti: drug–target interaction prediction based on multi-modal representation learning to bridge the gap between new chemical entities and known heterogeneous network. *Bioinformatics*, 37(23):4485–4492, 2021.

[58] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[59] Zeming Lin, Halil Akin, Roshan Rao, Brian Hie, Zhongkai Zhu, Wenting Lu, Nikita Smetanin, Robert Verkuil, Ori Kabeli, Yaniv Shmueli, et al. Evolutionary-scale prediction of atomic-level protein structure with a language model. *Science*, 379(6637):1123–1130, 2023.

[60] Jin Su, Chenchen Han, Yuyang Zhou, Junjie Shan, Xibin Zhou, and Fajie Yuan. Saprot: Protein language modeling with structure-aware vocabulary. *bioRxiv*, pages 2023–10, 2023.

[61] Christian Dallago, Jody Mou, Kadina E Johnston, Bruce J Wittmann, Nicholas Bhattacharya, Samuel Goldman, Ali Madani, and Kevin K Yang. Flip: Benchmark tasks in fitness landscape inference for proteins. *bioRxiv*, pages 2021–11, 2021.

[62] Robert Schmirler, Michael Heinzinger, and Burkhard Rost. Fine-tuning protein language models boosts predictions across diverse tasks. *Nature Communications*, 15(1):7407, 2024.

[63] Carlos Outeiral and Charlotte M Deane. Codon language embeddings provide strong signals for use in protein engineering. *Nature Machine Intelligence*, 6(2):170–179, 2024.

[64] Yang Tan, Mingchen Li, Ziyi Zhou, Pan Tan, Huiqun Yu, Guisheng Fan, and Liang Hong. Peta: evaluating the impact of protein transfer learning with sub-word tokenization on downstream applications. *Journal of Cheminformatics*, 16(1):92, 2024.

[65] Shawn Reeves and Subha Kalyaanamoorthy. Zero-shot transfer of protein sequence likelihood models to thermostability prediction. *Nature Machine Intelligence*, 6(9):1063–1076, 2024.

[66] Justas Dauparas, Ivan Anishchenko, Nathaniel Bennett, Hua Bai, Robert J Ragotte, Lukas F Milles, Basile IM Wicky, Alexis Courbet, Rob J de Haas, Neville Bethel, et al. Robust deep learning–based protein sequence design using proteinmpnn. *Science*, 378(6615):49–56, 2022.

[67] Baris E Suzek, Yuqi Wang, Hongzhan Huang, Peter B McGarvey, Cathy H Wu, and UniProt Consortium. Uniref clusters: a comprehensive and scalable alternative for improving sequence similarity searches. *Bioinformatics*, 31(6):926–932, 2015.

[68] Bowen Jing, Stephan Eismann, Patricia Suriana, Raphael JL Townshend, and Ron Dror. Learning from protein structure with geometric vector perceptrons. *arXiv preprint arXiv:2009.01411*, 2020.

[69] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020.

[70] Ching-Yao Chuang, Joshua Robinson, Yen-Chen Lin, Antonio Torralba, and Stefanie Jegelka. Debiased contrastive learning. *Advances in neural information processing systems*, 33:8765–8775, 2020.

[71] Kaiwen Zha, Peng Cao, Jeany Son, Yuzhe Yang, and Dina Katabi. Rank-n-contrast: learning continuous representations for regression. *Advances in Neural Information Processing Systems*, 36, 2024.

[72] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[73] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021.

[74] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020.

[75] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. *Advances in neural information processing systems*, 33:18661–18673, 2020.

[76] Yuzhe Yang, Kaiwen Zha, Yingcong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. In *International conference on machine learning*, pages 11842–11851. PMLR, 2021.

[77] Henry Dieckhaus, Michael Brocidiacono, Nicholas Randolph, and Brian Kuhlman. Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *bioRxiv*, 2023.

[78] Lasse M Blaabjerg, Maher M Kassem, Lydia L Good, Nicolas Jonsson, Matteo Cagiada, Kristoffer E Johansson, Wouter Boomsma, Amelie Stein, and Kresten Lindorff-Larsen. Rapid protein stability prediction using deep learning representations. *Elife*, 12:e82593, 2023.

[79] Bian Li, Yucheng T Yang, John A Capra, and Mark B Gerstein. Predicting changes in protein thermostability upon point mutation with deep 3d convolutional neural networks. *bioRxiv*, 2020.

[80] Henry Dieckhaus, Michael Brocidiacono, Nicholas Z Randolph, and Brian Kuhlman. Transfer learning to leverage larger datasets for improved prediction of protein stability changes. *Proceedings of the National Academy of Sciences*, 121(6):e2314853121, 2024.

[81] Matteo Cagiada, Sergey Ovchinnikov, and Kresten Lindorff-Larsen. Predicting absolute protein folding stability using generative models. *bioRxiv*, pages 2024–03, 2024.

[82] Eric A Franzosa, Kevin J Lynagh, and Yu Xia. Structural correlates of protein melting temperature. *Experimental Standard Conditions of Enzyme Characterizations*, pages 99–106, 2009.

[83] Anna Jarzab, Nils Kurzawa, Thomas Hopf, Matthias Moerch, Jana Zecha, Niels Leijten, Yangyang Bian, Eva Musiol, Melanie Maschberger, Gabriele Stoehr, et al. Meltome atlas—thermal proteome stability across the tree of life. *Nature methods*, 17(5):495–503, 2020.

[84] Jose M Sanchez-Ruiz. Protein kinetic stability. *Biophysical chemistry*, 148(1-3):1–15, 2010.

[85] Benjamin J Livesey, Mihaly Badonyi, Mafalda Dias, Jonathan Frazer, Sushant Kumar, Kresten Lindorff-Larsen, David M McCandlish, Rose Orenbuch, Courtney A Shearer, Lara Muffley, et al. Guidelines for releasing a variant effect predictor. *ArXiv*, 2024.

[86] Pascal Notin, Aaron Kollasch, Daniel Ritter, Lood van Niekerk, Steffanie Paul, Han Spinner, Nathan Rollins, Ada Shaw, Rose Orenbuch, Ruben Weitzman, Jonathan Frazer, Mafalda Dias, Dinko Franceschi, Yarin Gal, and Debora Marks. Proteingym: Large-scale benchmarks for protein fitness prediction and design. In A. Oh, T. Neumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, editors, *Advances in Neural Information Processing Systems*, volume 36, pages 64331–64379. Curran Associates, Inc., 2023.

[87] David Ding, Ada Y Shaw, Sam Sinai, Nathan Rollins, Noam Prywes, David F Savage, Michael T Laub, and Debora S Marks. Protein design using structure-based residue preferences. *Nature Communications*, 15(1):1639, 2024.

[88] Pascal Notin, Nathan Rollins, Yarin Gal, Chris Sander, and Debora Marks. Machine learning for functional protein design. *Nature biotechnology*, 42(2):216–228, 2024.

[89] Zemin Zhang and William I Wood. A profile hidden markov model for signal peptides generated by hmmer. *Bioinformatics*, 19(2):307–308, 2003.

[90] Sebastián García López, Søren Hauberg, and Wouter Boomsma. Probabilistic multiple sequence alignment using spatial transformations. *bioRxiv*, 2024. Preprint.

[91] Sebastián García López, Jesper Salomon, and Wouter Boomsma. Cross-species vs species-specific models for protein melting temperature prediction. *bioRxiv*, 2024. Preprint.