TÉCNICO LISBOA

Deep Learning for Protein Thermostability Engineering

Rafael Espinheira Alves

Thesis to obtain the Master of Science Degree in

Biological Engineering

Supervisors: Prof. Dr. Marcel J. T. Reinders
Prof. Dr. Ana Luísa Nobre Fred

Examination Committee

Chairperson: Prof. Dr. Ana Margarida Nunes da Mata Pires de Azevedo
Supervisor: Prof. Dr. Ana Luísa Nobre Fred
Member of the Committee: Prof. Dr. Maria Margarida Campos da Silveira

January 2021

Preface

The work presented in this thesis was performed at the Delft Bioinformatics Lab of Delft University of the Technology (Delft, Netherlands), during the period February-December 2020, under the supervision of Prof. Dr. Marcel J. T. Reinders. The thesis was co-supervised at Instituto Superior Técnico by Prof. Ana Luísa Nobre Fred.

I declare that this document is an original work of my own authorship and that it fulfils all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Acknowledgements

First of all, I would like to thank Marcel for believing in me and giving me the chance to join the Delft Bioinformatics Lab at TU Delft. I only wish that more professors were like you, because only rarely do students find such a motivated, supportive and down-to-earth supervisor. I would also like to thank Jurgen and DSM for accompanying this project with valuable input and Prof. Ana Fred for the supervision.

To Stavros, who I was lucky to have on my side, thank you for everything. To this day, I haven't met many people as great as you, who impact the people around them in such a positive way as you do. Your insight and your mood made the time I spent at the DBL amazing, making my most difficult times easier and my good moments even better, and helped make this project all it could be.

And to everyone else at the DBL, especially to Bram, Irene and Lou, and to Tamim, Mostafa, Ramin and Osman, thank you for making the DBL a welcoming and friendly place.

To my friends in Delft, thank you all for making my stay in this wonderful city the best it could be. To Ana Helena and your amazing group of people, to Pedro, my brother, and to Parreira, Lenny, Marina and Patrícia, and to my roommates Jean and Lohana, Philipp, Mike and Dan.

This marks the end of a hard journey that I could not have accomplished without the wonderful people that are now an essential part of my life. To Carlos, for sharing with me the struggle of university, and for making these years possible. Your unstoppable positivity and support are now a constant in our lives that makes us all happy to be a part of yours. To Henrique, for all the moments we shared in and out of Portugal. In times both hard and happy, your presence lights up our moods and you show us what really is important in life. To Adriana, for being the friend that was missing in my life, and in all of ours. You are the glue that makes us the group that we are, and your room will always be our safe haven. You are the best friends I could ask for, and together we make the best friend group I ever had. Thank you for being a part of this adventure. I cherish all the memories we made, hoping that this is only the beginning of everything else this life has to offer.

To the good people of the MEBiol15 class, thank you for making my times at IST memorable. To Caldas, Afonso and Samurai, for the memories we made playing video games, surviving classes and exams, and working on group projects. To Cristina, for being the amazing person that you are and for sharing with me some of my dearest hobbies. To Catarina, for showing me that life is full of pleasant surprises and for your endless support and belief in me.

And from before IST, to my friends Filipe and Mota, thank you for all the moments we shared and for staying in my life even after we followed different paths in life.

To my family, thank you for providing everything I needed and more. To my brother, for the guidance that helps me know who I want to be. To my grandmother, for filling me with joy like no one else can.

Finally, I would like to thank everyone else that in one way or another had an impact in my life. From professors, to family, and to friends I am no longer in contact with, we shared positive and negative moments that helped shape me as I am today, and helped me steer my life in the right direction.

Abstract

Recent advances in natural language processing show that modern neural networks are capable of learning context and semantics of language with unsupervised pre-training. Using such language models, researchers are now introducing new ways to represent protein sequences as continuous vectors (embeddings) that capture biophysical properties directly from unlabelled sequences. In this work, the embeddings generated by *SeqVec*, a deep learning model based on the *ELMo* language model and trained on the *UniRef50* data set, were studied for their potential to capture protein thermostability.

Three thermostability data sets were prepared and used to train and evaluate several machine learning models for their capability to predict protein thermostability properties using only the *SeqVec* embeddings as features. Although far from perfect, experiments on wild-type proteins show that such models produce meaningful predictions of protein melting temperature, and can isolate proteins with high thermostability. Additionally, models trained to predict the effect of mutations on the protein thermostability were capable of achieving Matthews correlation coefficients as high as 0.354 on independent testing data, a competitive value compared with recent literature.

Using transfer-learning for protein stability prediction opens up a new form of sequence-based tools that do not rely on biophysical features and do not require protein structure information. With this work it was shown that this is a promising approach to protein thermostability prediction, but the lack of data is still a large limitation.

Keywords

Machine learning; Deep learning; Language models; Protein engineering; Protein thermostability prediction.

Resumo

Os mais recentes avanços em processamento de linguagem mostram que redes neuronais artificiais são capazes de aprender conceitos como contexto e semântica apenas com treino não-supervisionado. Usando estes modelos de linguagem, novas formas de representar proteínas como vectores contínuos que capturam propriedades biológicas diretamente de sequências sem anotação estão a ser desenvolvidas. Neste trabalho, os vectores gerados pelo modelo *SeqVec*, inspirado no modelo de linguagem *ELMo* e treinado no conjunto de dados *UniRef50*, foram estudados pela sua capacidade de capturar a estabilidade térmica de proteínas.

Três conjuntos de dados de estabilidade térmica de proteínas foram preparados e usados para treinar e avaliar diversos modelos de aprendizagem automática pela sua capacidade de previsão de valores de estabilidade térmica utilizando apenas os vectores produzidos pelo modelo *SeqVec*. Ainda longe do ideal, experiências com sequências de proteínas naturais mostram que estes modelos são capazes de produzir previsões informativas, e que conseguem isolar proteínas com elevada estabilidade térmica. Adicionalmente, modelos treinados para prever o efeito de mutações na estabilidade térmica de proteínas foram capazes de atingir valores de correlação de Matthews de até 0.354 em dados independentes, um valor capaz de competir com a literatura atual.

A utilização destes métodos para a previsão da estabilidade térmica de proteínas abre uma nova abordagem para a engenharia de proteinas que não requer a preparação de características fisico-quimicas nem de características estruturais para descrever as proteínas. Com este trabalho, foi demonstrado que esta abordagem tem bastante potencial, mas que ainda é limitada pela falta de dados disponível.

Palavras Chave

Aprendizagem automática; Aprendizagem profunda; Modelos de linguagem; Engenharia de proteínas; Previsão de estabilidade térmica.

Contents

1 Introduction **1**

1.1 Context and Motivation . . . 1

1.1.1 Computational methods for protein engineering . . . 1

1.1.2 Using unlabelled protein sequence data . . . 2

1.2 Objectives and proposed framework . . . 4

1.3 Thesis outline . . . 4

2 Background **5**

2.1 Biology background . . . 5

2.1.1 Amino acids and the protein sequence . . . 5

2.1.2 Amino acid interactions and the protein structure . . . 6

2.1.3 Protein folding and stability . . . 8

2.2 Protein thermostability engineering state of the art . . . 10

2.2.1 Obtaining protein thermostability data . . . 10

2.2.2 Protein thermostability databases . . . 12

2.2.3 Computational methods for protein thermostability enhancement . . . 13

2.2.4 Computational methods for protein thermostability prediction . . . 14

2.3 Data science background . . . 16

2.3.1 The data science process . . . 16

2.3.2 Supervised machine learning algorithms . . . 18

2.3.3 Evaluation of supervised machine learning models . . . 21

2.3.4 Recent advances in natural language processing . . . 23

2.3.5 The ELMo language model and the SeqVec protein sequence model . . . 26

3 Validation of the SeqVec model **29**

3.1 Materials and Methods . . . 29

3.1.1 Protein Data Bank secondary structure data set processing . . . 29

3.1.2 Machine learning models implementation . . . 29

3.2 Results . . . 30

3.2.1 SeqVec successfully captures protein and amino acid properties . . . 30

3.2.2 SeqVec embeddings can be used for secondary structure prediction . . . 32

3.3 Discussion . . . 33

4 Thermostability prediction with the ProTherm wild-type data set **35**

4.1 Materials and Methods . . . 35

4.1.1 ProTherm wild-type proteins thermostability data set processing . . . 35

4.1.2 Machine learning models implementation . . . 35

4.2 Results . . . 36
4.2.1 SeqVec embeddings do not capture free Gibbs energy of unfolding . . . 36
4.2.2 Wild-type protein thermostability prediction was not successful . . . 37
4.3 Discussion . . . 39
5 Prediction of thermostability changes with the ProTherm single-mutants data set 41
5.1 Materials and Methods . . . 41
5.1.1 ProTherm single-mutants thermostability data set processing . . . 41
5.1.2 Machine learning models implementation . . . 42
5.2 Results . . . 46
5.2.1 Combining the embedding pairs to describe mutation records . . . 46
5.2.2 The precision-MCC trade-off . . . 48
5.2.3 Mutations to similar sequences are more accurately predicted . . . 49
5.3 Discussion . . . 50
6 The effect of mutations in the SeqVec embeddings 53
6.1 Materials and Methods . . . 53
6.1.1 Describing the effect of a mutation in the embeddings . . . 53
6.1.2 Mutagenesis data set preparation . . . 53
6.1.3 The Wilcoxon rank-sum statistical test . . . 54
6.2 Results . . . 54
6.2.1 Distant amino acid embeddings capture the effect of a mutation . . . 54
6.2.2 Mutations to binding sites cause a stronger effect in the embeddings . . . 55
6.2.3 3D distance is captured by the amino acid embeddings . . . 56
6.3 Discussion . . . 57
7 Thermostability prediction with the Meltome Atlas wild-type data set 59
7.1 Materials and Methods . . . 59
7.1.1 The cross-species Meltome Atlas data set processing . . . 59
7.1.2 Machine learning models implementation . . . 60
7.2 Results . . . 61
7.2.1 Thermophile protein embeddings differ from mesophile embeddings . . . 61
7.2.2 SeqVec features are correlated with properties related to thermostability . . . 61
7.2.3 The embeddings can be used to train a melting temperature predictor . . . 63
7.2.4 The captured thermostability information is too general . . . 64
7.3 Discussion . . . 68
8 Conclusions and future work 71

A Appendix **81**

A.1 Additional implementation details . . . 81

A.1.1 Software used . . . 81

A.1.2 Cross-validation hyper-parameter tuning procedure . . . 81

A.2 Additional data processing details . . . 82

A.2.1 Wild-type protein thermostability data set processing . . . 82

A.2.2 Single-mutants protein thermostability data set processing . . . 83

A.2.3 Mutagenesis data set processing . . . 85

A.2.4 Protein melting temperatures data set processing . . . 86

A.3 Additional model development information . . . 88

A.3.1 Secondary structure prediction . . . 88

A.3.2 Thermostability change upon mutation prediction . . . 88

A.3.3 Melting temperature prediction . . . 92

List of Figures

1.1 Traditional directed evolution procedure and ML-guided directed evolution. 2
1.2 Results obtained by the UniRep mLSTM model on protein biology modelling. 3

2.1 Elements of protein secondary structure. 6
2.2 The relationship between the protein structure levels and function. 7
2.3 Protein denaturation and protein renaturation. 8
2.4 Protein folding thermodynamics depicted as free-energy funnels. 9
2.5 Data obtained by a differential scanning calorimetry experiment. 11
2.6 Performance comparison between different protein thermostability prediction models. . . 14
2.7 Workflow and challenges of applying a machine learning model. 17
2.8 Linear and non-linear decision functions in the same classification problem. 20
2.9 The Alexnet convolutional neural network for image classification. 24
2.10 Schematic of the LSTM model. 25
2.11 The pre-training of the BERT, GPT and ELMo language models. 26
2.12 Schematic of the SeqVec deep learning model. 27
2.13 Dimensionality of the embeddings produced by SeqVec after processing a protein sequence. 28

3.1 t-SNE projection to two dimensions of the protein embeddings obtained from the secondary structure data set, coloured by enzyme class. 31
3.2 t-SNE projection to two dimensions of the amino acid embeddings obtained from the secondary structure data set, coloured by amino acid type, physicochemical properties, size and secondary structure labels. 31
3.3 Confusion matrix and ROC curves of the k-NN secondary structure predictor. 32

4.1 t-SNE and PCA two-dimensional projections of the protein embeddings obtained from the ProTherm wild-type thermostability data set. 37
4.2 Scatter plots of the predicted free Gibbs energy of unfolding values and their true values. 38

5.1 t-SNE and PCA projection to two dimensions of the mutations records, represented by the difference between the wild-type sequence embedding and the mutant sequence embedding 46
5.2 t-SNE and PCA projection to two dimensions of the mutations records, represented by the Diff_5 feature set. 47
5.3 Confusion matrix, PRC and ROC curves of the tuned logistic regression predictor of protein thermostability changes of single mutations. 48

5.4 Confusion matrix, PRC and ROC curves of the tuned linear SVM predictor of protein thermostability changes of single mutations. 49
5.5 Confusion matrix, PRC and ROC curves of the tuned MLP predictor of protein thermostability changes of single mutations. 49
5.6 PRCs of the tuned linear SVM predictor of protein thermostability changes of single mutations on different subsets of the testing set S434 with increasingly higher sequence identity percentages with the training set S3272. 50

6.1 The euclidean distance between the wild-type sequence amino acid embeddings and the mutant sequence amino acid embeddings of the data set S3706. 55
6.2 The euclidean distance between the wild-type sequence amino acid embeddings and the mutant sequence amino acid embeddings of the mutagenesis data set. 55
6.3 Boxplot representation of the euclidean distance value between wild-type sequence amino acid embeddings and mutant sequence amino acid embeddings in segments of sequence distance to the mutation. 56
6.4 Boxplot representation of the euclidean distance value between wild-type sequence amino acid embeddings and mutant sequence amino acid embeddings in segments of distance in space to the mutation. 57

7.1 t-SNE and PCA projection to two dimensions of the protein embeddings obtained from the Meltome Atlas data set, coloured by melting temperature. 62
7.2 Scatter plot of three SeqVec features and three protein features known to be correlated with thermostability. 62
7.3 Scatter plots of the melting temperatures predicted by the MLP model developed with the protein embeddings, and their true values. 64
7.4 Performance metrics of the MLP model developed with the protein embeddings for melting temperature prediction, upon training and evaluation on individual organisms. 65
7.5 Performance metrics of the MLP model developed with the protein embeddings for melting temperature prediction, upon training on all organisms except one, which was used for evaluation. 66
7.6 Scatter plots of the melting temperatures predicted by the MLP model developed with the protein embeddings and their true values, on a training set without proteins from M. musculus, which were used as the testing set. 67
7.7 Performance metrics of the MLP model developed with the protein embeddings for melting temperature prediction, upon training on individual organisms and testing on all the others. 67

A.1 Histogram of the distribution of free Gibbs energy of unfolding in the processed ProTherm wild-type data set. 82
A.2 Free Gibbs energy of unfolding values as a function of temperature and pH in the processed ProTherm wild-type data set. 82
A.3 Histogram of the distribution of free Gibbs energy of unfolding changes upon single mutations in the processed data set S3706. 84
A.4 Histogram of the distribution of the euclidean distances between amino acid embeddings of wild-type and mutant sequences of mutations to binding sites in the mutagenesis data set. 85
A.5 Histogram of the distribution of the euclidean distances between amino acid embeddings of wild-type and mutant sequences of mutations outside binding sites in the mutagenesis data set. 85
A.6 Area under the melting curve as a function of the melting temperature of each protein in the Meltome Atlas data set. 86
A.7 Histogram of the distribution of melting temperature values in the processed Meltome Atlas data set. 87
A.8 t-SNE projection to two dimensions of the protein embeddings obtained from the Meltome Atlas data set, coloured by organism. 87
A.9 Cross-validation hyper-parameter tuning process of the k-NN secondary structure predictor. 88
A.10 Cross-validation hyper-parameter tuning process of the logistic regression predictor of protein thermostability changes upon mutations. 89
A.11 Cross-validation hyper-parameter tuning process of the linear SVM predictor of protein thermostability changes upon mutations. 90
A.12 Training history of the MLP predictor of protein thermostability changes upon mutations. . 90
A.13 Cross-validation hyper-parameter tuning process of the MLP predictor of protein thermostability changes upon mutations. 90
A.14 Performance of the DT model with baseline features on the testing set S434. 91
A.15 Sequence identity between each of the sequences in the testing set S434 and their first, second and third best matches in the training set S3272. 91
A.16 Training history of the MLP predictor of protein melting temperature. 92
A.17 Cross-validation hyper-parameter tuning process of the MLP predictor of protein melting temperature. 92
A.18 Scatter plots of the melting temperatures predicted by the MLP model developed with the baseline feature set, and their true values. 93

A.19 Performance metrics of the MLP model developed with the baseline feature set for melting temperature prediction, upon training and evaluation on individual organisms. 94
A.20 Performance metrics of the MLP model developed with the baseline features for melting temperature prediction, upon training on all organisms except one, which was used for evaluation. 95
A.21 Performance metrics of the MLP model developed with the baseline features for melting temperature prediction, upon training on individual organisms and testing on all the others. 95
A.22 RMSE values of a naive melting temperature linear regression baseline in a leave-one-out experiment with the processed Meltome Atlas data set. 96

List of Tables

2.1 The five groups of common amino acids, identified by the polarity and charge (at physiological pH) of their side chains. 5
2.3 Summary of the performance of several protein thermostability predictors. 15

3.1 Description of the data partitions of the secondary structure data set prepared for the separate training and evaluation of the k-NN machine learning algorithm. 30

4.1 Performance of the free Gibbs energy regression models using different pre-processing approaches to take into account the experimental conditions. 38

5.1 Data sets of protein thermostability changes upon single mutations collected for this work. 41
5.3 Description of the data partitions of the data set S3706 of protein thermostability changes upon single mutations used for the separate training and evaluation of the machine learning algorithms. 42
5.5 Description of the different feature sets produced to represent the mutation records in the high-dimensional embedding space. 43
5.7 Performance of the basic set of classifiers on the testing set S434, trained on the data set S3272 with the mutation records, represented by Diff_5 feature set. 47

7.1 Performance of several machine learning regression models trained to predict the melting temperature of the protein embeddings. 63

A.1 Performance of the basic set of classifiers on the testing set S434, trained on the data set S3272, using the different features generated to represent the mutation records. 88
A.2 Performance of the basic set of classifiers on the testing set S434, trained on the data set S3272 without insignificant mutation records, using the different features generated to represent the mutation records. 89
A.3 Performance of the basic set of classifiers on the testing set S434, trained on the data set S3272 balanced with the reverse mutations, using the different features generated to represent the mutation records. 89
A.4 Performance of a linear regression protein melting temperature model with different baseline features on the testing set. 93
A.5 Performance of the melting temperature prediction MLP model implemented with the baseline feature set. 93
A.6 Performance of several naive baselines for melting temperature prediction on all species. 96

Acronyms

3D	Three-dimensional
Acc	Accuracy
ANN	Artificial Neural Network
AUC	Area Under Curve
BERT	Bidirectional Encoder Representations from Transformers
biLSTM	Bidirectional Long-Short Term Memory
BLAST	Basic Local Alignment Search Tool
CD	Circular Dichroism
CD-HIT	Cluster Database at High Identity with Tolerance
CharCNN	Character-level Convolutional Neural Network
CM	Confusion Matrix
DSC	Differential Scanning Calorimetry
DT	Decision Tree
EC	Enzyme Commission
ELMo	Embeddings from Language Models
EVS	Explained Variance Score
FN	False Negative
FP	False Positive
GPT	Generative Pre-trained Transformer
GRU	Gated Recurrent Unit
k-NN	k-Nearest Neighbours
LDA	Linear Discriminant Analysis
LSTM	Long-Short Term Memory
MCC	Matthews Correlation Coefficient
ML	Machine Learning
MLP	Multi-Layered Perceptron
mLSTM	Multiplicative Long-Short Term Memory
MSA	Multiple Sequence Alignment
MSE	Mean Squared Error

NLP	Natural Language Processing
OGT	Optimal Growth Temperature
PCA	Principal Component Analysis
PCC	Pearson Correlation Coefficient
PDB	Protein Data Bank
PRC	Precision-Recall Curve
RBF	Radial Basis Function
ReLU	Rectified Linear Unit
RF	Random Forest
RNN	Recurrent Neural Network
RMSE	Root Mean Squared Error
ROC	Receiver Operating Characteristic
RS	Rank-Sums
SCC	Spearman Correlation Coefficient
SeqVec	Sequence-to-vector
SSE	Sum of Squared Errors
SVM	Support Vector Machine
TN	True Negative
TP	True Positive
t-SNE	t-Distributed Stochastic Neighbour Embedding
UMAP	Uniform Manifold Approximation and Projection

1. Introduction

1.1 Context and Motivation

This work arises from the connection of three problems and the most recent attempts to solve them. The economic interest in engineering proteins to increase their valuable properties, joined with the progress in the development of computer models for human language and the rising abundance of unlabelled protein sequence data makes learning biological function directly from sequence not only possible, but also very interesting.

1.1.1 Computational methods for protein engineering

Protein engineering aims to obtain proteins with useful properties for technology, science and medicine. As the amino acid sequence determines the protein's properties [1], by performing specific amino acid modifications, new proteins have already been designed and optimized, with applications in chemical and pharmaceutical biosynthesis, regenerative medicine, food industries and waste biodegradation [2]. To guide the protein mutation process, protein engineering constantly looks for models to correctly predict protein properties such as function, catalytic activity and stability [3].

Traditionally, protein engineering is based on the rational design strategy, which faces overwhelming amounts of possible mutations to model [3] and from which most are not functional or can produce unaccounted effects in stability [4], and on the directed evolution strategy, which rarely finds beneficial mutations [5] in an iterative approach of trial and error, with expensive and time-consuming screening procedures [3].

New, emerging, directed evolution methods use Machine Learning (ML) models to learn functional properties from the entire fitness landscape, and then apply this knowledge to guide protein mutations towards higher protein fitness levels [3]. This approach is very appealing for protein engineering due to its generalizability: while traditional protein engineering methods are limited to the specific protein families in which they were developed, these models can quickly make predictions about new enzyme variants which were previously unknown, only by applying the knowledge they learned from the data used to train the model [2]. Used extensively in bioinformatics [6], this approach has already seen success in predicting protein structure, function, catalytic activity, solubility and stability, and can be applied in combination with the directed evolution strategy (Figure 1.1) by reducing the experimental effort and improving the exploration of the sequence space of the traditional method [3]. These models are already known to be capable of handling complex relationships in sequence data, and are only limited by the quality and quantity of data available used in the training steps [2].

One of the protein properties with industrial interest is the thermostability of enzymes. Enzymes are the primary catalytic agents of cells, responsible for conducting most chemical reactions, and are

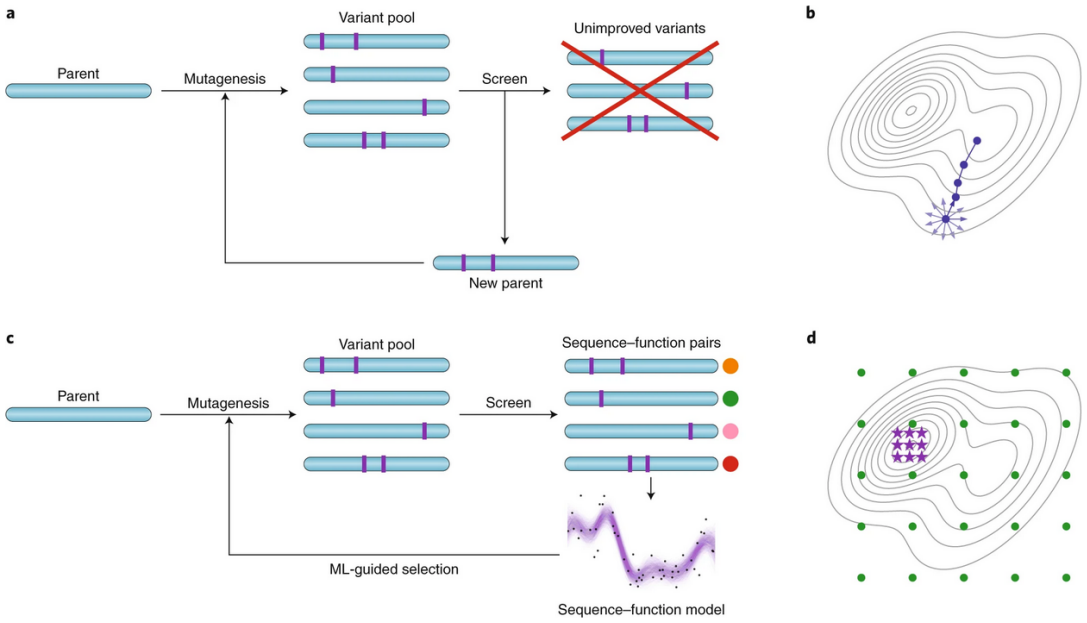

Figure 1.1: Traditional directed evolution procedure and Machine Learning (ML)-guided directed evolution. (a) Traditional directed evolution generation and screening process consists of iterative cycles of trial and error. (b) The screening process of traditional directed evolution improvements as a series of local searches in the protein fitness landscape. (c) ML guided directed evolution chooses mutations based on information learned from all candidates. (d) ML guided directed evolution initially learns the entire fitness landscape (green circles) to then quickly converge towards fitness improvements (violet stars) [3].

frequently applied in chemical, biotechnology and medical industries as catalysts for diverse reactions. Increasing the thermostability of enzymes is useful to facilitate certain purification steps based on heat treatments and to allow the use of higher reaction temperatures, making for a faster and more sterile process, and is also associated with a higher stability do denaturing agents [3], [7].

Numerous protein thermostability models and ML predictors have already been developed, but their performance is still far from ideal, as protein stability modelling is a very difficult task and for which there exists limited data [8].

1.1.2 Using unlabelled protein sequence data

Technological improvements in protein sequencing are currently causing an exponential increase in the size of protein sequence databases such as UniProt [9]. Despite the large amounts of available genomes and proteomes, this information is only as useful as the quality of its annotation, which is dependent on the available tools for their analysis [10]. As a consequence of cheaper sequencing procedures, biological databases are growing faster than the computing power required for annotation of their information [11]

This continuous growth of sequenced genomes is, in its majority, a result of sequencing of very similar and close to identical strains of the same species, where 90% of the proteins have a larger than

90% identity, which presents a large challenge for databases [9]. The most frequently used approaches for protein annotation are based on local alignment tools, but these still take several minutes to process a single protein [12] or have significant hardware requirements [13]. This methodology also can not accurately model highly divergent natural sequences with similar functions, nor highly similar sequences with different properties, and can not be applied to the many proteins that have no known homologs and remain uncategorized as attempts to address these challenges are frequently limited to individual protein families and lack in generalizability [10]. The large number of proteins with no evolutionary information is also a problem, and processing data sets from metagenomic samples is already a major challenge for mainstream protein annotation methods. As databases double in size every two years, this process quadruples in difficulty, resulting in a constant need for faster solutions [11].

A possible solution for this issue is being found in ML models used for Natural Language Processing (NLP). Analogously to how a sentence's meaning is determined by the words that compose that sentence, a protein's structure, function and its properties are also determined by the amino acid sequence that composes that protein [10], and as proteins are one of the most important cellular elements, responsible for most functions necessary for life, a model that successfully predicts protein properties directly from its amino acid sequence is of extreme value [11].

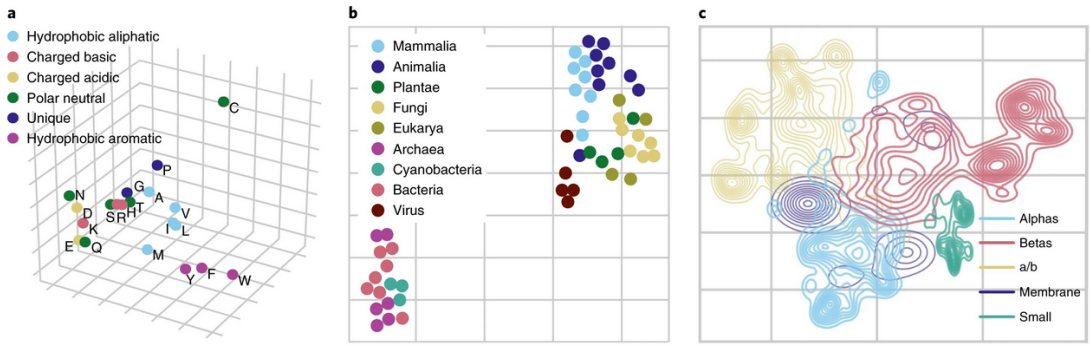

Figure 1.2: Results obtained by the *UniRep* Multiplicative Long-Short Term Memory (mLSTM) model on protein biology modelling. This deep learning model successfully learns content-rich protein and amino acid representations directly from protein sequence. (a) Principal Component Analysis (PCA) of amino acid embeddings shows clusters representative of properties. (b) t-Distributed Stochastic Neighbour Embedding (t-SNE) of proteome-average embeddings of model organisms shows clusters with evolutionary information. (c) t-SNE of protein embeddings show clusters representative of structural classes [14].

Already some efforts in making the connection between sequence and function are being made, with NLP models that find high-level protein representations, called embeddings, that are closely associated with the protein function and properties (Figure 1.2). The abundance of protein sequence databases supplies a large data set to train such models, and the hypothesis that these models can be applied to model protein sequences and learn the biological rules that dictate protein properties directly from the amino acids is seeing a lot of support [10], [11], [14], [15], [16], [17].

Additionally, high-dimensional protein representations learned by a deep learning algorithm that ac-

curately model the functional outcome of a protein sequence can also be used for protein engineering, as proteins can simply be compared by the distance of these representations in the feature space. This allows for the evaluation of which mutations are interesting and increase protein fitness, at accessible computational requirements [10].

The need for better models of protein thermostability makes it a very promising area on which to assess the performance of the protein representations learned by language models trained on biological data, as no such study was published.

1.2 Objectives and proposed framework

This thesis aims to explore the potential of the application of Sequence-to-vector (SeqVec), a Bidirectional Long-Short Term Memory (biLSTM) model based on the Embeddings from Language Models (ELMo) NLP model and pre-trained on the *UniRef50* data set, as published by [11], in protein thermostability engineering. For this, a data science approach was followed, with the goal of predicting protein thermostability directly from protein sequence, without user-defined, evolutionary or structural features, and using only the high-dimensional protein representations learned by the SeqVec model to develop protein thermostability models. First, protein thermostability data was collected and compiled; then, the data was explored in detail and processed accordingly; lastly, several machine learning predictors were chosen, applied and evaluated.

1.3 Thesis outline

The rest of this dissertation is organized as follows. In the next chapter, **Background**, relevant topics from protein biology are introduced, protein thermostability engineering state of the art is discussed, and the concepts from machine learning necessary for this project are presented. The dissertation then details the methodology, results and discussion sections of each experiment separately, where: the first experiment, **Validation of the SeqVec model** aims to explore the embeddings for biological meaning and develop a proof-of-concept secondary structure predictor; the second experiment, **Thermostability prediction with the *ProTherm* wild-type data set**, presents the first effort in thermostability prediction using the SeqVec embeddings; the third experiment details the process of **Prediction of thermostability changes with the *ProTherm* single-mutants data set**; the fourth experiment aims to study **The effect of mutations in the SeqVec embeddings**; the fifth and final experiment, **Thermostability prediction with the *Meltome Atlas* wild-type data set**, presents a final attempt at modelling protein thermostability directly from sequence using the SeqVec embeddings. The last chapter in this thesis, **Conclusions and future work**, summarizes the achievement of the objectives of this thesis and presents an overview of the project, together with some notes on future work based on this dissertation.

2. Background

2.1 Biology background

Proteins are responsible for almost every biological process that happens in a cell and are the most abundant biological macromolecules, presenting an extensive diversity of functions and properties [18]. The genetic information stored in a cell's deoxyribonucleic acid is expressed through proteins, making them the most important machinery of life [11].

2.1.1 Amino acids and the protein sequence

Proteins are composed of amino acids, linked by covalent bonds in a linear sequence. All proteins, from all biological domains and kingdoms, are polymers that use the same set of 20 different amino acids, which are identified by their different side-chains. Some proteins also include nonstandard amino acids, which are usually derivatives of the common amino acids [19].

Table 2.1: The five groups of common amino acids, identified by the polarity and charge (at physiological pH) of their side chains, and their most relevant properties. Adapted from [18].

Amino acid	1-letter code	Group	Properties
Glycine Alanine Proline Valine Leucine Isoleucine Methionine	G A P V L I M	Nonpolar, aliphatic R groups	Hydrophobic, tend to cluster together in the protein's interior, establishing hydrophobic interactions that stabilize the protein
Phenylalanine Tyrosine Tryptophan	F Y W	Aromatic R groups	Relatively hydrophobic, also participate in hydrophobic interactions but have a larger side-chain volume
Serine Threonine Cysteine Asparagine Glutamine	S T C N Q	Polar, uncharged R groups	Relatively hydrophilic, frequently form hydrogen bonds with water
Lysine Arginine Histidine	K R H	Positively charged R groups	The most hydrophilic, basic character
Aspartate Glutamate	D E	Negatively charged R groups	The most hydrophilic, acid character

All amino acids are composed by a central carbon atom, to which are bonded a carboxyl group, an amino group, a single hydrogen atom, and a variable side chain, also called the R group, which is used to group the 20 common amino acids in different classes according to its chemical properties. Usually, this classification is performed by the polarity and charge (at pH 7) of the amino acid, which determines their solubility in water [18], [19]. Based on this property, five main classes of amino acids are usually

described, as summarized in Table 2.1 [18]. Other authors group amino acids in just three classes: nonpolar, uncharged polar and charged polar amino acids [19], while other authors classify the amino acids in a more detailed approach, with more properties of the R group [20].

The formation of peptides, polypeptides, oligopeptides and proteins is done by sequentially linking amino acids, which are joined to each other in a linear structure by peptide bonds. The linked amino acids in a protein, no longer in their complete, isolated form, are now usually called amino acid residues. These amino acid sequences can have a wide variety of sizes, from very small peptides with very few monomers to very large macromolecules [18].

2.1.2 Amino acid interactions and the protein structure

Although composed by a sequence of amino acids, a protein is more than a simple linear structure. There are four degrees of protein structure, from which the sequence of amino acid residues is only the first. The secondary structure is the stable conformation of amino acid residues in several, distinct structural patterns. Upon folding of the polypeptide, emerges the tertiary structure, which describes the entire Three-dimensional (3D) conformation of the protein, and some proteic complexes are formed by more than one polypeptide subunit, with a characteristic spatial arrangement called the quaternary structure [18].

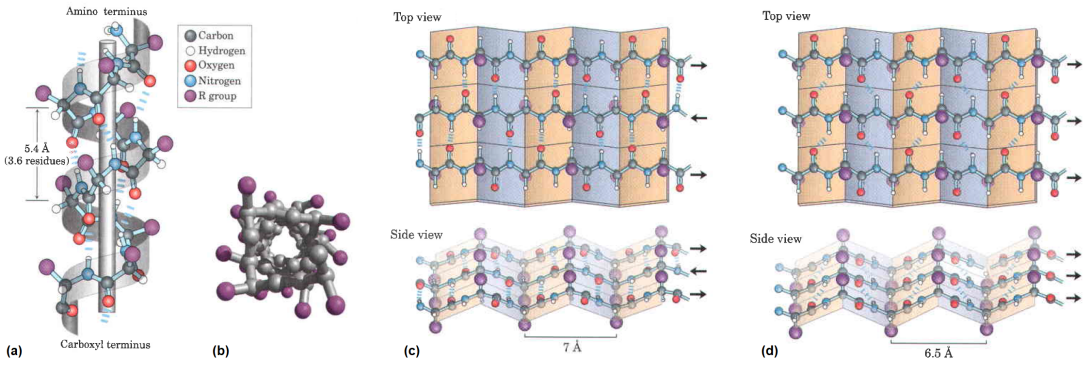

Figure 2.1: Elements of protein secondary structure. (a) Side view of the α helix, detailing the repeating pattern with 3.6 residues in length and the hydrogen-bonds responsible for this arrangement. (b) Top view of the α helix, detailing the positions of the R groups. (c) Top and side views of the anti-parallel β sheet, detailing the hydrogen-bonds and the positions of the R groups. (d) Top and side views of the parallel β sheet, detailing the hydrogen-bonds and the positions of the R groups. Adapted from [18].

The diversity of protein backbone conformations comes from the fact that many of the peptide bonds in a protein are free to rotate in space, guided by the laws of physics towards the most stable global conformation, called the native state of the protein [19]. This makes hydrophobic and ionic amino acid interactions, and the hydrogen and disulfide bonds between them, the most important forces in stabilizing the structural patterns of the protein's secondary structure, from which the most common types are the α helix and the β sheets and turns (Figure 2.1), both held together by hydrogen bonds between amino acid

residues in the protein sequence that arise from the type and spacial orientation of their R-groups [18].

Several elements of secondary structure can be connected together in different motifs (also called a supersecondary structure), and can originate independently stable self-contained patterns called domains [20]. All of these amino acid residue interactions originate a specific three-dimensional arrangement characteristic of each protein, which is called the tertiary structure. Different amino acid sequences give rise to different conformations and different protein properties [18].

The function of proteins is also a consequence of its conformation. Each fibrous protein has long polypeptide backbones in specific repeating secondary structure patterns to correctly guarantee rigidity or flexibility [18], and each ligand-binding protein and each enzyme can discriminate between closely related molecules due to both steric and physical complementarity between the binding site of the protein and the individual ligand, as a result of the specific amino acids that make up the protein's binding site (or the enzyme's active site) [20]. This complex process involves a sequence of reaction intermediates, and where the structure of the protein and the structure of each intermediate play an important role in the physical and chemical interactions necessary for this transformation [18], making protein structure fundamental in determining protein function (Figure 2.2).

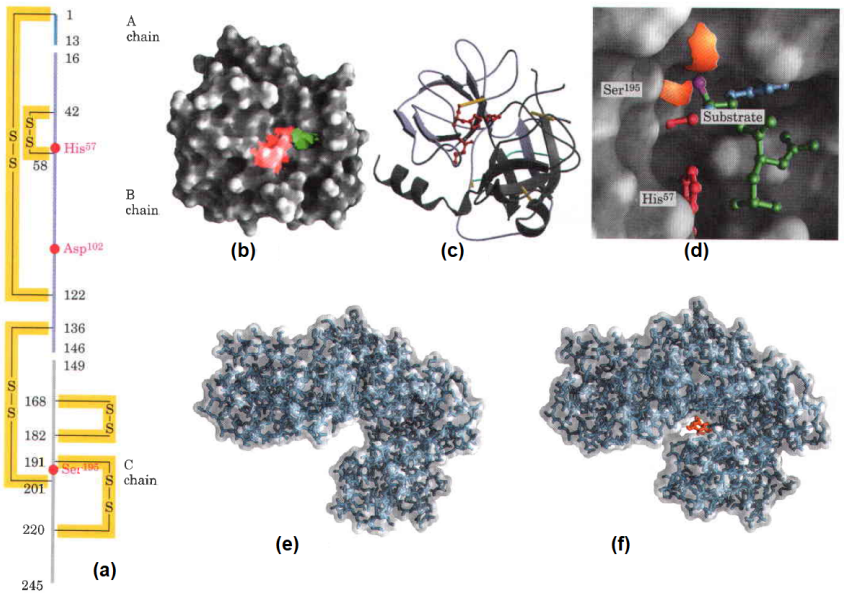

Figure 2.2: The relationship between the first three levels of protein structure, and how they are responsible for protein function. (a) the primary structure of *Chymotrypsin*, detailing the disulfide bonds (yellow) that hold the three polypeptides together. (b) the surface area of *Chymotrypsin*, detailing the residues of the active site of the enzyme (red) and the location where the substrate binds to the protein (green). (c) The polypeptide backbone of *Chymotrypsin*, detailing the multiple secondary structure elements and the disulfide bonds in the three-dimensional space. (d) Detailed view of *Chymotrypsin*'s active site and its complementarity with the substrate. (e) The three-dimensional structure of *Hexokinase* in its free conformation. (f) The three-dimensional structure of *Hexokinase* upon binding of D-glucose (red), detailing the resulting conformational change. Adapted from [18].

2.1.3 Protein folding and stability

Although the amino acid sequence of a protein is known to determine its tertiary structure and function, the process through which a protein folds into its native state is extremely complex and is still only mildly understood, making it difficult to model [18]. Some models describe this folding as a hierarchical process, in which secondary structure elements emerge before long distant interactions similar to the assembly of a puzzle [20], while other models characterize it as a spontaneous collapse of the peptide chains, guided by amino acid interactions [18].

The folding process of some proteins can also be assisted by *chaperone* proteins, which interact with the intermediates of this process to facilitate the correct folding of the protein, or even provide the unique micro-environment in which a specific folding step is made possible. Moreover, after the folding process, some proteins can only exert their functions after specific post-translational modifications, such as acetylation, phosphorilation and methylation reactions, that alter local properties of the structure and conformation of the protein [18].

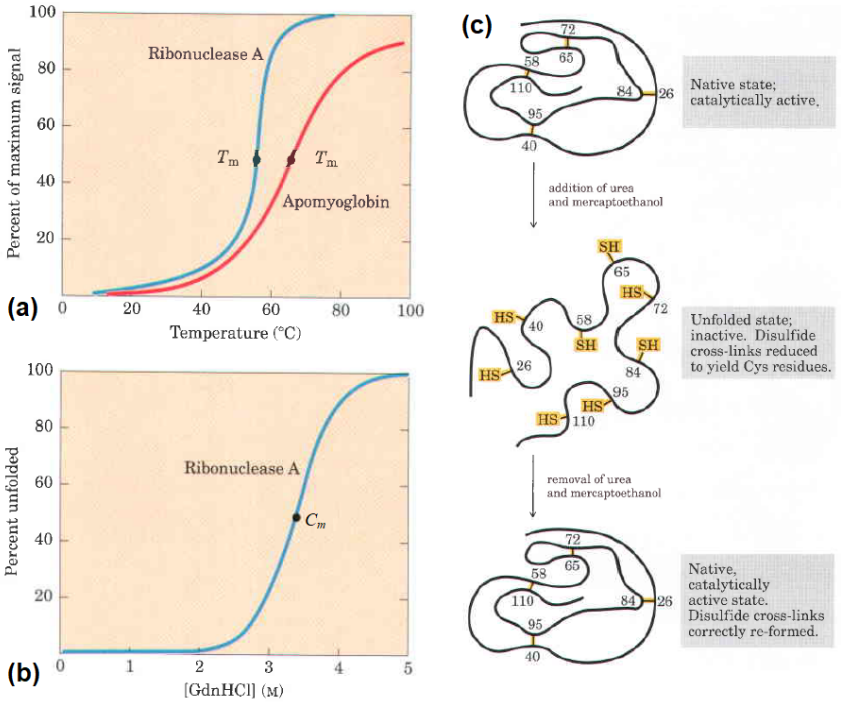

Figure 2.3: Protein denaturation and protein renaturation. (a) Thermal denaturation of proteins, observed by monitoring of the percentage of signal in a circular dichroism experiment. The melting temperature of a protein is defined as the temperature value at which half of the proteins in solution are denatured. In this example, the melting temperature of the Apomyoglobin (red) is shown to be higher than that of Ribonuclease A (blue), indicating that the former is more thermostable. (b) Chemical denaturation of proteins, also monitored by circular dichroism. (c) Simplified scheme of how a denaturing agent interacts with the amino acids in the protein, leading to loss of three-dimensional conformation. In this case, the protein is capable of reestablishing the correct structure in a process called renaturing. Adapted from [18].

As proteins need to have the correct conformation in order to carry out their functions, a loss of 3D conformation can lead to the loss of protein function. This process, called denaturation, can happen as a result of changes in the protein's environment in different forms: as an increase in temperature that breaks the weak hydrogen bonds responsible for the native conformation, as an extreme change in pH affecting the charge of the amino acid residues and thus altering electrostatic forces, and by various organic solvents such as alcohols or detergents that disrupt hydrophobic interactions [18], [20]. As such, the stability of a protein can be described as a measure of how resistant its structure is to these changes (Figure 2.3). However, the connection between sequence, structure and stability is not direct, since proteins with very different heat resistances can have very similar structures [18], and other neighbouring molecules, including other proteins, in dense cellular environments, can also result in complex interactions that can alter the stability of the native state of a protein [21].

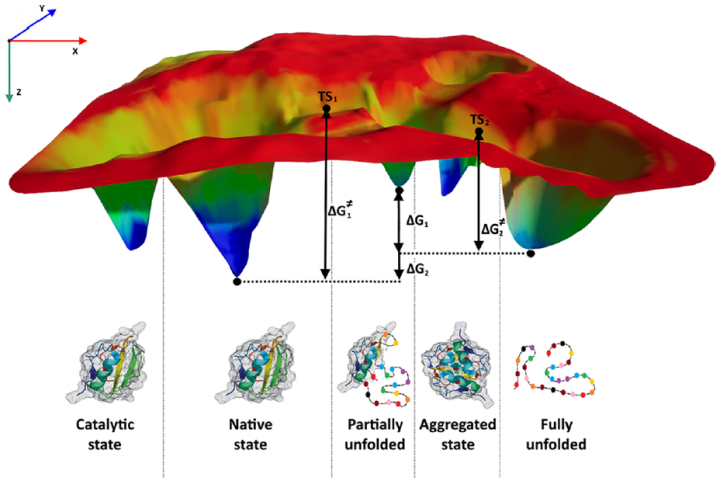

Figure 2.4: Protein folding thermodynamics depicted as an energy landscape. Several characteristic conformational states are detailed as free-energy funnels, indicating a higher stability. These states are separated by energy barriers with frequent depressions on the sides of the funnels that represent semistable folding intermediates [18]. The z axis is also representative of the percentage of amino acids in native conformation, and the width of the energy well at any point is correlated of the entropy in that intermediate state. These conformations are called molten globules, as these collapsed states are not entirely fixed [18]. Also note that the fully unfolded protein is quite stable, due to a large number of interactions with the surrounding water molecules. Adapted from [4].

The protein conformation that is most thermodynamically stable is the one having the lowest Gibbs free energy (G). However, the reduced G difference between the folded and unfolded states (ΔG) makes the native conformation of most proteins only slightly stable [18]. Held by electrostatic forces, this conformation is mostly stabilized by the weak interactions between the amino acids, namely hydrophobic interactions predominating in the protein's interior and hydrophilic interactions predominating on the protein's exterior by forming a solvation layer with the sorrounding water molecules. Although ionic interactions are stronger, they do not greatly stabilize proteins [19], as well as hydrogen bonds in the protein, but disulfide bonds show a great role in this stabilization [19].

The thermodynamics of protein folding can be represented as free-energy funnels, where different protein conformations can be compared for stability (Figure 2.4), but given the high complexity of the protein folding process, protein stability modelling is still a very difficult task [4]. The extreme importance of proteins in biology and the growing use of enzymes for industrial purposes makes modelling the correct 3D conformation of proteins an ongoing challenge of great interest.

2.2 Protein thermostability engineering state of the art

Protein engineering uses amino acid mutations to increase the usefulness of proteins. However, this process can have the unwanted side effect of causing structural changes in the protein that can affect protein stability. Either to evaluate if a mutation will disrupt the stability of the designed protein, or to design proteins with higher stability, the development of computational tools for prediction of protein stability is greatly pursued in protein engineering [8].

Empirical approaches such as random extension of the protein terminals and chemical modification with polymers have been outperformed by site specific mutations [7], but due to the slow and costly process of experimental screening for thermostability of mutation libraries, *in silico* methods are generally preferred. These are usually based on phylogenetics, structural analysis, free energy calculations, or machine learning, with a recent trend towards machine learning-based predictors. As the performance of these methods is tied to the amount of available high-quality data, there is a growing demand for systematic collection, validation and organization of protein thermostability databases [22].

2.2.1 Obtaining protein thermostability data

Protein stability is usually described by the free Gibbs energy difference between the folded and the unfolded states of the protein, ΔG [23], defined in Equation (2.1) as a function of the gas constant R, the absolute temperature T and the equilibrium constant of the unfolding transition K, calculated as the concentration of folded proteins divided by the concentration of unfolded proteins in equilibrium [24]. This can be used to characterize the thermostability of the protein, but also the stability of proteins in extreme environments in general, since protein structure integrity in extreme temperatures is correlated to integrity in extreme pH values and to the application of chemical denaturants [18]. The experimental determination of the free Gibbs energy of unfolding of a protein is usually done using Circular Dichroism (CD) spectroscopy, fluorescence spectroscopy or Differential Scanning Calorimetry (DSC) [25].

$$\Delta G = -RTlnK \tag{2.1}$$

The CD technique is based on a spectroscopic evaluation of the conformation of proteins. Proteins are optically active molecules, in the sense that they absorb the components of circularly polarized

light differently [26]. By analyzing of the CD spectrum at a given temperature, the fraction of folded and unfolded protein in solution can be determined, and by collecting these values as a function of temperature, the ΔG can be obtained by fitting the experimental values to the modified Gibbs-Helmholtz equation at constant pressure Equation (2.2) [23].

$$\Delta G = \Delta H(1 - \frac{T}{T_M} - \Delta C_p[(T_M - T) + Tln(\frac{T}{T_M})] \tag{2.2}$$

where ΔH is the enthalpy change (considered to be independent from temperature), T_M is the melting temperature of the protein and ΔC_p is the change in the heat capacity due to the unfolding. These values are obtained by fitting a nonlinear least squares method, and from which the ΔG is determined [24].

The fluorescence spectroscopy technique is based on the analysis of the response of tryptophan residues in the protein to fluorescent light, as this response is dependent on the accessibility of the tryptophan residues to the surface of the protein. Similarly to CD spectroscopy, by studying the fluorescence spectra of a protein with tryptophan residues at different temperatures, the fraction of unfolded and folded protein in solution can be determined as a function of temperature [27], allowing a similar mathematical calculation of the value of ΔG [28].

The DSC technique is also based on the Gibbs-Helmholtz equation Equation (2.2), but analyses the excess heat capacity of a solution as a function of temperature. The unfolding of a protein is described by a sharp endothermic peak centered at the melting temperature. By integration of this curve, the value of the transition enthalpy, ΔH and the value of the change in heat capacity as a result of the mutation, ΔC_p, are calculated directly, allowing the determination of the ΔG value (Figure 2.5) [23].

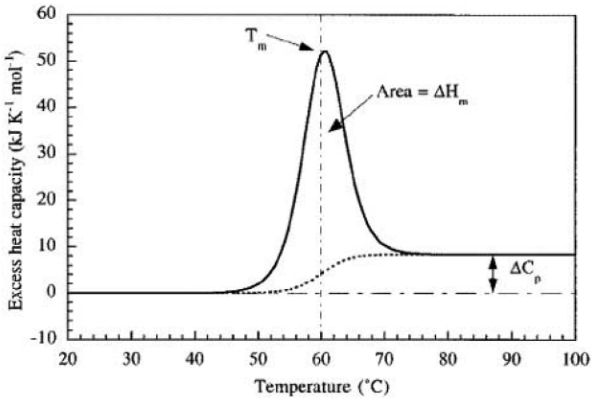

Figure 2.5: Data obtained by a differential scanning calorimetry experiment. The excess heat capacity of a solution (y-axis) is measured at different temperatures (x-axis), in which the two-state unfolding of a globular protein can be followed. The observed endothermic peak is centered at the protein's melting temperature, which can be integrated for calculation of the transition enthalpy. The difference in heat capacity as a result of the unfolding can also be obtained from this experiment. Adapted from [23].

The free Gibbs energy of unfolding parameter is also widely used to describe the effect that amino acid mutations have in the thermostability of the protein, by analysing the difference in free Gibbs energy

of unfolding change caused by the mutation, $\Delta\Delta G$ [25]. The most frequently used definition, given in Equation (2.3), describes a mutation that caused an increase in stability by a positive $\Delta\Delta G$ value [8].

$$\Delta\Delta G_{wt \to mt} = \Delta G_{mt} - \Delta G_{wt} \tag{2.3}$$

Two major issues arise from the experimental calculation of the change in free Gibbs energy of unfolding: the fitting of enthalpy change as a function of the temperature assumes that the unfolding of the protein follows a two-state equilibrium, and does not consider stable folding intermediates [23], and the value of ΔG can only be measured directly in the denaturation transition zone, the area where there are both native and unfolded proteins in solution. As ΔG is dependent on the environmental conditions, its values in physiological conditions need to be obtained by extrapolation [20].

Although there is an abundance of protein sequence and also protein structure data, data on protein stability is still not widely available, as this data is not only difficult to gather but also difficult to organize in coherent and well-curated databases. With the rising use of high-throughput techniques, more effort is being put into obtaining and processing this data more uniformly, and also in quantities more suitable for the development of protein thermostability models. [2].

2.2.2 Protein thermostability databases

The most widely referenced database of protein thermostability information is the *ProTherm* database [2], [22]. The last publlished description of this database, from 2006, mentions 7014 wild-type proteins, 8202 single-mutant proteins, 1277 double-mutants and 620 multiple-mutant proteins, retrieved from a collection of 1497 scientific articles, totalling 17113 entries from 771 different proteins [25]. The database contains protein information such as identifiers and mutation details, experimental conditions information such as temperature, pH, buffers and experimental methods, and thermodynamic data such as the free Gibbs energy of unfolding (ΔG), difference of ΔG caused by the mutations ($\Delta\Delta G$), concentration of denaturing agents for chemical denaturation and melting temperature (T_m) for thermal denaturation, melting temperature change caused by the mutations (ΔT_m), and enthalpy and heat capacity changes caused by the mutations (ΔH and ΔC_p) [25].

This database contains now over 25000 records, but it is becoming more evident that the *ProTherm* database is not actively maintained and suffers from numerous issues such as inaccurate or incomplete annotations, and even wrong values [22], [29]. Adding to this, different experimental conditions for the same record also impact some machine learning predictors [30], [31].

To deal with these problems, different attempts were made to manually filter and curate this database. Several papers make their own data sets available to the scientific community, from which the *PoPMuSiC-2.0* [30] and the *I-Mutant2.0* [32] readily available data sets are frequently mentioned [31], [8]. Other efforts to compile a curated database for training of machine learning predictors are recently gaining

some attention, such as *ProtaBank* [33], a repository with several mutation data sets for diverse protein engineering applications, including the previously mentioned protein thermostability data sets collected and processed by different authors, and the *FireProtDB* [22] database, which attempts to keep a manually curated version of the *ProTherm* database.

In sum, the best protein thermostability databases are still very limited in size, where in fact the *FireProtDB* and *ProTherm* databases have only 1329 manually curated single-point mutations from 79 proteins, and 1564 single-point mutations from 99 proteins after cleanup, respectively [2].

Organisms can also be categorized in psycrophilic, mesophilic and thermophilic, according to their Optimal Growth Temperature (OGT). With this in mind, the *ProtDataTherm* database collected all available protein sequences from microorganisms that have been categorized based on their growth temperature. This database contains over 14 million protein sequences with a *UniProt* [9] identifier, from which over 30 thousand also have a secondary structure Protein Data Bank (PDB) identifier [34], clustered by Pfam protein family identifiers and their corresponding thermostability categories: psycrophilic if OGT$<$20 ℃, mesophilic if 20 ℃$<$OGT$<$40 ℃ and thermophilic if 40 ℃$<$OGT [35]. With multiple protein families with sequences belonging to multiple thermostability categories, this database can be used to compare engineered protein designs with homologues with higher thermostability and for analysis of modulating factors of thermostability, across different protein families and within specific families [35]. This database has already been successfully used to to develop a pattern recognition algorithm that suggests thermostability improving mutations [35], and to classify an organism's OGT by proteomic analysis [36], but although it is described as an emerging protein thermostability database, it has not yet been used for the development of protein thermostability prediction methods.

Another effort to record the proteome stability of a large number of proteins from several organisms across the tree of life is the *Meltome Atlas*. This data set contains the melting curves of over 48000 proteins, with high-quality annotations obtained by a systematic mass spectrometry approach [37], but is yet to be used in protein thermostability engineering methods.

2.2.3 Computational methods for protein thermostability enhancement

Protein engineering methods for thermostability enhancement that do not require structural information are preferred due to their applicability in proteins for which this data is unavailable [7]. These sequence-based methods are founded on the consensus concept for protein thermostability engineering [38], where a Multiple Sequence Alignment (MSA) across homologous proteins is used to determine non-consensus residues, which are substituted by consensus ones. Limited by the contradictory effect of some of the suggested mutations, these methods can be improved by including amino acids interactions, by reducing the MSA to thermophilic homologues and by incorporating structure information [7].

When the structure information of the protein is available, fragile regions of the protein can be identi-

fied and strengthened by introducing, for example, hydrogen bonds or disulfide bridges. These methods are limited to proteins with such information, and are limited by the experimental conditions at which crystallographic data is obtained, the time-consuming and inaccurate modelling of large molecular systems and by not taking into account solvent molecules and long range electrostatic interactions [7].

Although quite useful, computational protein thermostability enhancement methods only suggest amino acids to mutate and protein locations to modify, and do not actually predict the outcome that these changes will have on the protein.

2.2.4 Computational methods for protein thermostability prediction

Protein thermostability engineering methods that accurately predict the effect of any mutation on the stability of a protein are of great interest, because with information about the protein sequence or structure, these methods can be used to predict the difference in free energy of unfolding or melting temperature caused by amino acid mutations (Figure 2.6) [7]. Protein thermostability predictors are either based on energy function calculations or on machine learning algorithms, but even the best thermostability predictors still need further improvement [7].

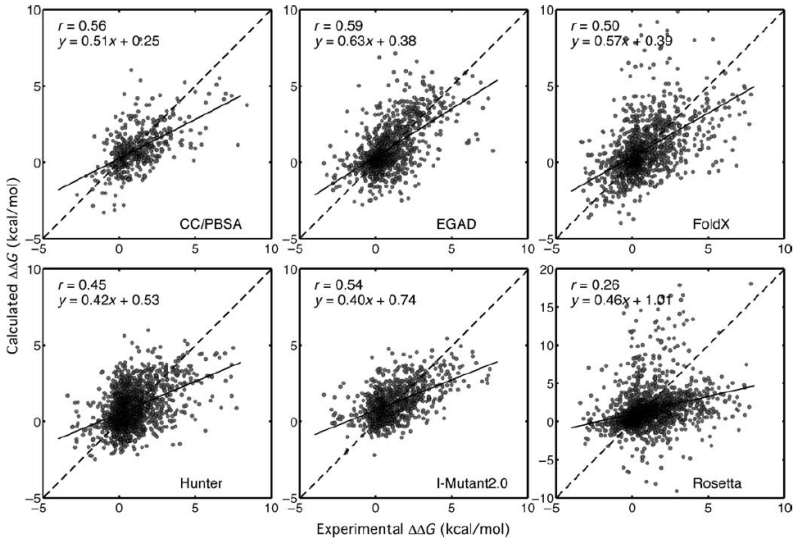

Figure 2.6: Performance of different protein thermostability prediction algorithms, evaluated in an independent testing set for prediction of the $\Delta\Delta G$ value of mutations. From the correlation coefficient (r) and the equation of regression (y) of each model, we can observe that the best results, which belong to the *CC/PBSA* and the *EGAD*, are still far from ideal. Adapted from [39].

Methods based on energy functions derive their models from physical-based potentials that use fundamental analyses of the forces between atoms, such as *EGAD* [40] and *CC/PBSA* [41] that use different force-field models to generate and score the structures, or from knowledge-based potentials that use statistical analyses of different protein properties, such as *FoldX* [42] and *Hunter* [39]. There are also hybrid models, such as *Rosetta* [43], which is one of the most used methods to reproduce native

protein structures from sequence [39].

Methods based on ML use different approaches to generate the features describing each protein and each mutation, sometimes including energy potentials, and also diverge in choice of learning algorithm. There are sequence-based methods that consider the amino acids in the protein to generate physico-chemical properties and scoring matrices, such as *I-Mutant* [32], *MUpro* [44], *iPTREE-STAB* [45], and structure-based methods that also incorporate 3D protein-structure information to generate features related to secondary structure and atomic interactions, such as *PoPMuSiC* [30] and *DUET* [46]. In general, structure-based methods outperform sequence-based methods [8], and several efforts have also been employed to integrate multiple of the previously mentioned methods, from which the *iStable2.0* algorithm shows the most promising results, achieving the best prediction performance in literature with a Matthews Correlation Coefficient (MCC) score of 0.708 on independent testing data (Table 2.3) [8].

However, the performance of each of these models depends on the data used to train and test them. Another recent thermostability prediction model based on convolutional neural networks with physicochemical features of the amino acid substitution achieved a MCC of 0.56, and performed a comparison with other previously mentioned models. In their test data, the *PoPMuSiC* and *I-Mutant* models achieved MCC values of only 0.20 and 0.25, respectively [47]. This makes a comparison of different models inadequate if different data sets are used.

Table 2.3: Summary of the performance of several protein thermostability predictors on the data set S630, a protein thermostability data set obtained from *ProTherm*, curated and published by [30] for testing of prediction models. The *iStable2.0* model, which integrates other protein thermostability prediction models including those in the presented table, outperforms even the best models. Adapted from [8].

	Tool	Classification		Regression	Features used	Algorithm used
		Acc	MCC	PCC		
Structure -based models	iStable2.0	0.892	0.708	0.714	Integrating various models	XGBoost
	I-Mutant2.0	0.837	0.547	0.669	Mutation details and neighbouring residues	SVM
	DUET	0.776	0.358	0.458	Statistical potential energy function and Geometric and physicochemical properties	SVM
	PoPMuSiC	0.757	0.291	0.424	Linear combinations of statistical potentials	MLP
Sequence -based models	iStable2.0	0.873	0.652	0.695	Integrating various models	XGBoost
	I-Mutant2.0	0.819	0.491	0.546	Mutation details and neighbouring residues	SVM
	iPTREE-STAB	0.810	0.443	0.496	Mutation details and neighbouring residues	DT
	MUpro	0.756	0.248	–	Mutation details and neighbouring residues	MLP

Protein thermostability prediction is by itself a very difficult task, and is further limited in performance due to the lack of high-quality data, as even the best thermostability prediction models have serious overfitting issues [48] and are very biased to mutations with negative effects in stability, as these are the most abundant records in the databases [49]. Given the wide range of enzyme mechanisms, reactions, experimental conditions, families and properties, it is not easy to choose, apply and explain a ML model for protein engineering as it implies a certain level of understanding in data science [2].

2.3 Data science background

With the open and widespread use of the Internet, data collection has seen an explosive growth [50]. Data science aims to extract knowledge from data by developing strategies and methods to analyze the increasing amount of data that is generated daily [51]. The process of learning what information is stored in big data collections is called data mining, and usually makes use of Machine Learning (ML) algorithms [52].

A ML application can usually be identified as one of the classic machine learning problems, from which the unsupervised and supervised learning problems are the most frequent. In unsupervised learning problems, the data records are not identified by any particular label, and uses algorithms that focus on the structure of the data, identifying for example clusters, frequent patterns and association rules, and are out of the scope of this thesis. In supervised learning problems, the data records are labelled, further described as classification problems if the labels are discrete classes, or as regression problems if the labels are continuous variables, and uses algorithms that can be trained to predict the label of new records [51], [52].

2.3.1 The data science process

The choice and application of a machine learning model to a data science project is only one of the steps in the process of knowledge discovery from data. The data science process encompasses three main steps, summarized in Figure 2.7, namely: data preparation, training of the ML algorithm, and a performance evaluation [2].

In the first step of data collection and processing, data sets are often cleaned of noise and inconsistent data, as well as irrelevant data that is removed in a data selection process. Some applications also require the data to be transformed, using summary or aggregation operations that reduce the level of detail, normalization operations that map the values to different, more informative, ranges, and balancing operations that aim to remove or generate new records so that all classes are adequately present in the data set. This step also involves the use of data visualization and representation techniques, which usually require the mapping of high-dimensional data to a two-dimensional or three-dimensional observable space [51], with resource to techniques such as Principal Component Analysis (PCA), t-Distributed Stochastic Neighbour Embedding (t-SNE) or Uniform Manifold Approximation and Projection (UMAP) [53], [54], [55]. The processed data is then usually split in a training set and in a testing set, which are then used in the second and third steps of the construction of a ML model [2].

In the second step, the training of the predictor is performed. In this stage, underfitting and overfitting need to be limited, by finding the best trade-off between a model that can learn the required dependencies in data while preventing the learning of noise in the data. [2].

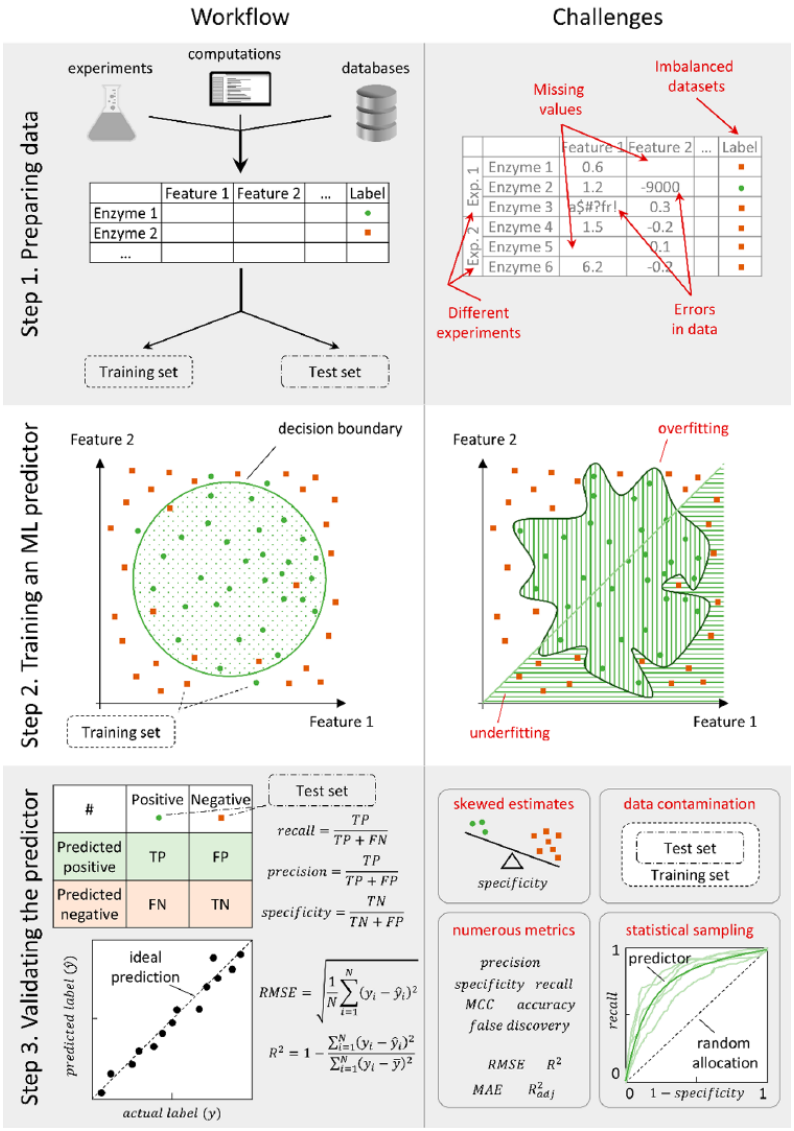

Figure 2.7: Summary of the workflow and challenges associated with the construction of a ML model. Step 1 details the data gathering and processing stage, where experimental conditions, missing values, erroneous annotations and data imbalance need to be accounted for. Step 2 details the training of the model, where underfitting and overfitting are the main issues. Step 3 details the evaluation of the model, where data contamination needs to be avoided, and the correct performance metrics need to be evaluated and statistical sampling can be used to explain skewed estimates. Adapted from [2].

Different model hyperparameters and different predictors are usually attempted and chosen by a K-fold cross-validation procedure where the training set is partitioned into K subsets, which are then cycled in a training of the models on $K-1$ subsets and evaluation on the remaining subset, and from which the model with the best average performance is chosen. Cross-validation has the added benefits of maximizing the use of the available training data while preventing overfitting to the data set, and of allowing an unbiased choice of models [51]. Depending on the choice of model and data available, an additional dimensionality reduction step, by feature selection (using methods such as the filter and the

wrapper) or feature extraction (using methods such as PCA), may need to be applied to avoid the *curse of dimensionality*, which sees reduced performances in ML models that use large numbers of features compared to the number of data samples [56].

In the third step, the trained ML model is evaluated for its performance on unseen data, using the testing set to draw conclusions on the model's generalizability. Choosing the correct metrics and correctly dealing with skewed estimates and data contamination is paramount in analysing ML models [2]. This stage of the process involves a detailed comparison between different models, where the use of adequate data visualization and representation techniques provides additional information to discuss the success of the model. In addition to the performance metrics, model simplicity can also be a decisive factor, where simple models are generally favoured against overly complex models, to avoid overfitting issues [51].

The performance of a ML predictor depends severely on the availability of large amounts of quality data to train the model but also depends greatly on the choice of prediction algorithm, for which there exists a great diversity. While a linear predictor can facilitate the analysis of the model and draw conclusions about the predictions, sometimes only a complex non-linear model can accurately capture the required relationships in the data. [2].

2.3.2 Supervised machine learning algorithms

A trained ML model takes data records as inputs, which are described by different features, and outputs a prediction of each record's label. In classification problems, the model uses a decision function to identify the classes of the records, which is usually obtained by optimization of a loss function. In regression problems, the training of the model is also usually performed by optimization of a loss function, but in these problems the model learns a mathematical relationship between the data features and the output [56]. Different algorithms use different loss functions, and obtain different results (Figure 2.8).

The k-Nearest Neighbours (k-NN) classifier is a form of lazy learning, because no model is actually constructed. This algorithm predicts the class of an input vector x as the most voted class among its k nearest training points, and all computation time is spent in the classification step. Although it can be used to learn decision functions with irregular shapes, the extended classification time and issues with noisy data are the main disadvantages of this algorithm [51]. The hyper-parameter k has a great impact on the performance of the classifier, and depends on each application, although in general increasing its value removes the effects of noise in data, but can also lead to less strict decision boundaries [57]. **The k-NN regression** is based on the classification algorithm, and estimates the outcome variable of a record by a local interpolation of the values of the k nearest training records [57].

The simplest ML models are based on linear relationships in data. **The linear regression** finds the linear relationship between the features and the target variable that minimizes the Sum of Squared

Errors (SSE). The coefficients w of this equation are obtained by minimizing an ordinary least squares loss function [57].

To deal with overfitting, regularization is usually applied with this model. By adding a l_2 regularization term $\alpha||w||_2^2$, called **the Ridge regression**, the coefficients of the model are also minimized, and by adding a l_1 regularization term $\alpha||w||_1$, called **the Lasso regression**, the optimization problem produces sparse coefficients, which can be useful to handle high-dimensional data. In these formulations, the hyperparameter α determines the regularization strength [56].

Either of these models can also be generalized to include polynomial features, where the design matrix is now extended with additional columns, defined by the degrees of the features to be included. Called **the polynomial regression**, this approach is usually employed to model more complex, non-linear relationships between the features and the output variable [56].

Another notable extension of the linear regression, which by optimization of the SSE estimates the mean value of the outcome variable, is **the quantile regression**, which estimates a chosen quantile of the target variable. It can be implemented to predict the confidence intervals of the linear regression, or can also be used to estimate the median value of the outcome variable, by minimization of the sum of absolute errors. It is usually employed because it is more robust to outliers, or to take into account skewed data distributions [58].

For classification, a frequently used linear model is the **the logistic regression classifier**, which aims to approximate a probability distribution of each class when given the features, modelling the outcome of a record according to a logistic function [56]. This model is fit to the data by optimizing the conditional log-likelihood function, using a gradient descent algorithm [57].

The Support Vector Machine (SVM) classifier is based on a more complex approach to decision functions. It finds the decision function that best separates the classes by using a set of training records to calculate the hyperplane with the maximum margin between the nearest points of the two different classes, called the support vectors. These algorithms require a long training time, but can reach high accuracy metrics and are quite resistant to overfitting. The explicit decision function can also be quite useful for interpretation of the learned model, and the ease of application of the kernel method, by mapping the data to higher dimensional feature spaces implicitly during training, can be used to produce non-linear decision functions at no additional computation time [51].

The training of this algorithm involves a C hyperparameter, which dictates the strength of the penalty given to records on the wrong side of the decision function, and also involves a manual choice of the kernel. Usually, the Radial Basis Function (RBF) and the polynomial kernels are applied, but their choice is not always clear and can have serious impacts on performance [57]. **The SVM regression** is adapted to continuous labels by including error penalties in the optimization problem [56].

A different approach to classification uses logical inference of the classes during training, developing rules during training to be used to predict the classes of new records. **The Decision Tree (DT) algorithm** generates a flowchart of the decisions that produce the predictions, which can be used for interpretation of the models [59]. The generation of a DT is based on an iterative procedure where the data is split according to the variable that leads to the most informative split, usually calculated according to the information gain criteria or the *Gini* impurity criteria [57]. The process then continues in a top-down approach through the several generated branches.

These algorithms are very robust to different data types and scales, and can manage large data sets without extensive computational times. They can, however, suffer from bias to imbalanced classes, and although they are prone to overfitting, several pruning algorithms and stopping criteria have been developed to address these issues [51]. **DT regression** can also be performed, and in this case the model finds local linear regressions that approximate the mean value of the data in different segments, both found during the iterative training of the algorithm [57].

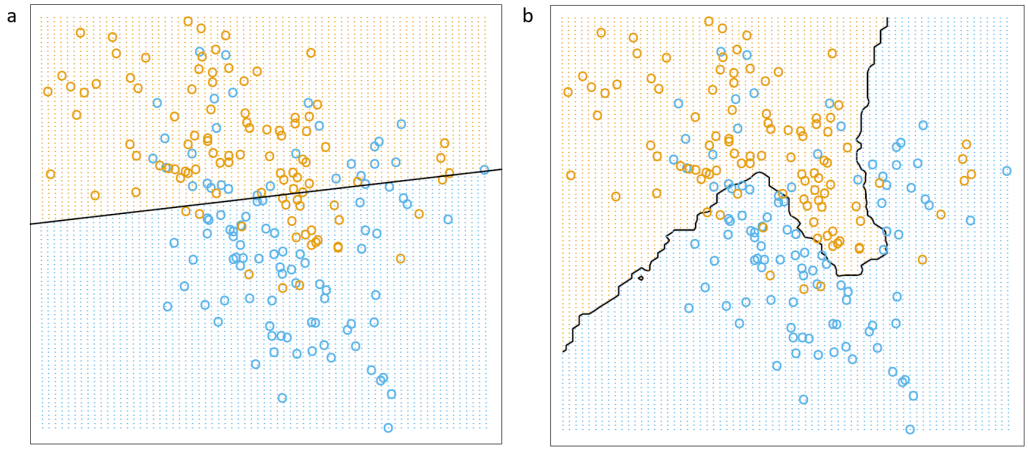

Figure 2.8: The decision functions of two different ML models on the same classification problem. (a) By coding the classes as a binary variable, a decision boundary can be obtained by the linear regression $x^T\hat{\beta} = 0.5$, which separates the classes by a straight line. (b) By implementing a k-NN algorithm that predicts the class of a record as the majority vote of its 15-nearest neighbours, a non-linear decision function can be obtained. With a non-linear decision boundary fewer records are misclassified. Adapted from [56].

The more complex **Artificial Neural Networks (ANNs)** are models based on arrangements of perceptrons, their functional units [59]. A perceptron has a linear part, where a weighted sum of its inputs is calculated, followed by a non-linear activation function that determines the output of the unit based on this sum [60]. The **Multi-Layered Perceptron (MLP) model** is one of the simplest implementations of ANNs, consisting of a feed-forward network of fully connected perceptron layers, with at least one hidden layer. By arranging multiple perceptrons organized in layers and using continuous and differentiable activation functions, a MLP can be used to deal with linearly non-separable data and very complex tasks [51]. The Rectified Linear Unit (ReLU) activation function, defined as $y = max(0, x)$, is frequently

used for perceptrons in the hidden layers due to its resistance to exploding and vanishing gradients [61], but other sigmoid or hyperbolic tangent are also widely applied. These models are difficult to analyze due to their black box nature [51], but by using different architectures, activation functions and loss functions, ANNs can be applied for a wide variety of learning problems of high difficulty [56].

Training of these models is done in a highly distributed process using the backpropagation algorithm. With this algorithm, the weights of the connections in the network are tuned iteratively by passing the training data through the network, where a chosen loss function is minimized by means of a gradient descent algorithm. Multiple passages of the training data can be applied, called training epochs, allowing the model to use each training record multiple times. This training is usually studied by leaving out a partition of the training data for validation, where after each epoch the model is evaluated for its performance in the validation data to prevent overfitting to the training data by applying too many training epochs. Also important to the training of these models is the learning rate of the gradient descent algorithm, where a small learning rate can cause the training to progress too slowly or become limited to the first local minimum it finds, and a large learning rate can cause the algorithm to skip over minimum values of the loss function [52]. For binary classification, the binary cross-entropy loss function is frequently used, while for regression problems the Mean Squared Error (MSE) can be used [62].

The naïve Bayes classifier is also frequently used. This model is based on the Bayes' theorem of conditional probability, and assumes that each feature has an independent effect on a given class to estimate the conditional likelihood of a record belonging to every class, and predicts the class of the record as the most probable one [59]. Although very simplistic and easy to implement, it can achieve good prediction performances, and can also be extended to include conditional dependencies between the features, using Bayesian networks and Markov logic networks [51].

No single model is ideal to any given application. Different models may perform better on different data sets of the same problem, or two models may perform quite well in a particular scenario although they learn different information. When a model's decision function systematically deviates from the true labels, it is said to be biased, and a model is said to have large variance if its decision boundary is different with different training data sets [63]. Ensemble classifiers, such as the **Random Forests (RFs) model**, aim to solve these issues by combining multiple simple models, and are effective at reducing overfitting and variance, and improving the efficiency of the use of the data and robustness of the model to noise and outliers [51], but the choice of algorithm is always particular to each application, taking into account the available data and each model's strengths and weaknesses.

2.3.3 Evaluation of supervised machine learning models

Classification models are usually evaluated in terms of the Confusion Matrix (CM), which counts the number of True Positive (TP), False Positive (FP), True Negative (TN) and False Negative (FN) predic-

tions. From this, the Accuracy (Acc), which measures the percentage of records correctly predicted from the total number of predictions, and error rate, which measures the percentage of records wrongly predicted from the total number of predictions, are usually calculated if the data is balanced [2].

However, if the data is imbalanced, metrics such as the recall (also called TP rate or sensitivity), precision and specificity (also called TN rate) are used instead. These take into account the different representation of each class in the data set, and prevent a biased evaluation of the model [2]. The precision measures the ability of the model to correctly label a positive record (Equation (2.4)), and the recall measures the ability of the model to find all the positive records (Equation (2.5)), and these can be measured by the F_1 score, which is an average of the model's precision and recall. This measure is adequate for imbalanced data sets in binary classification [64]. The specificity, also called TN rate is used to measure the capacity of the model to find all the negative records (Equation (2.7))

$$Precision = \frac{TP}{TP + FP} \tag{2.4}$$

$$Recall = \frac{TP}{TP + FN} \tag{2.5}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{2.6}$$

$$Specificity = \frac{TN}{TN + FP} \tag{2.7}$$

Additionally, the MCC is also frequently used for binary classification with imbalanced data because it takes into account all the parameters of the CM of the predictions (Equation (2.8)) [8].

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{2.8}$$

Additional statistical sampling techniques can also be performed for evaluation of the model, such as Receiver Operating Characteristic (ROC) curve and Precision-Recall Curve (PRC) analysis [65]. The ROC curve displays the performance of a classifier with different decision thresholds, by representing the TP rate of the model at different FP rates. Calculation of the ROC Area Under Curve (AUC) value summarizes the entire plot in a single value, and is also sometimes useful for comparison of different models. The PRC, which provides a description of the model's precision as a function of its recall at decreasing decision thresholds, is also useful if the precision of the model is of interest, specifically in cases where the positive class is under-represented [65].

Regression models are usually evaluated in terms of Root Mean Squared Error (RMSE), coefficient of determination r^2 and correlation coefficients such as the Pearson Correlation Coefficient (PCC).

The RMSE is the squared root of the mean squared error, an average of the squared errors that corresponds to the expected value of the SSE, and is a measure of the accuracy of the model (Equa-

tion (2.9)). The r^2 score is the coefficient of determination, and measures the goodness of the fit by evaluating the proportion of variance of the dependent variable that is explained by the model, and is calculated as a proportion between the SSE and the variance of the target variable (Equation (2.10)). The Explained Variance Score (EVS), very similar to the r^2 score, also measures the proportion of explained variance of the model, but takes into account the mean error of the model (Equation (2.11)), and together with the r^2 score can be used to measure the bias of the predictions [57].

$$\text{RMSE}(y, \hat{y}) = \sqrt{\frac{1}{n}\sum_{i=0}^{n-1}(y_i - \hat{y}_i)^2} \tag{2.9}$$

$$r^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} = 1 - \frac{\frac{SSE}{n}}{Var\{y\}} \tag{2.10}$$

$$\text{EVS}(y, \hat{y}) = 1 - \frac{Var\{y - \hat{y}\}}{Var\{y\}} = 1 - \frac{\frac{SSE-ME}{n}}{Var\{y\}} \tag{2.11}$$

In these formulations, y are the true values and $\hat{y}$ are the predictions, n is the number of samples, SSE is the sum of squared errors, Var is the variance and ME is the mean error. The concept of variance can be interpreted as the expected value of the squared deviation from the mean value [57].

The PCC is used to measure the linear relationship between the predictions and the true values (Equation (2.12)), and can be used as a measure of the dispersion of the predictions. The Spearman Correlation Coefficient (SCC) can also be used to allow possible nonlinear (but monotonic) relationships between the predictions and the true values to be measured, in which case the PCC would not be high but a correlation could exist nonetheless. This correlation coefficient is also more resistant to outliers. After ranking the records in ascending order of the true values, the SCC is calculated by Equation (2.13) [66].

$$r_p(y, \hat{y}) = \frac{\sum(y - \bar{y})(\hat{y} - \bar{\hat{y}})}{\sqrt{\sum(y - \bar{y})^2(\hat{y} - \bar{\hat{y}})^2}} \tag{2.12}$$

$$r_s(y, \hat{y}) = \frac{cov\{ry, r\hat{y}\}}{\sigma_{ry} \times \sigma_{r\hat{y}}} \tag{2.13}$$

where $cov\{ry, r\hat{y}\}$ is the covariance of the ranks of the true values and the predicted values, and σ_{ry} and $\sigma_{r\hat{y}}$ are the standard deviations of the true values and the predictions, respectively [66].

2.3.4 Recent advances in natural language processing

Ever since the first perceptron model was published [60], Artificial Neural Networks (ANNs) have been pushed to state of the art performances in very complex computational tasks such as image and video

recognition, and speech and text processing [67]. The building blocks of ANNs are layers of perceptrons, with different kinds of connections and operations. By using dense, fully connected layers, an ANN can model any continuous function [68], and by using convolutional layers, an ANN can learn complex features from the inputs [69]. Several other layer operations such as pooling, flattening and normalization can be used with different purposes, and different mathematical operations can be used to connect distant units in the network, which can be combined to achieve the previously mentioned modelling tasks (Figure 2.9) [67].

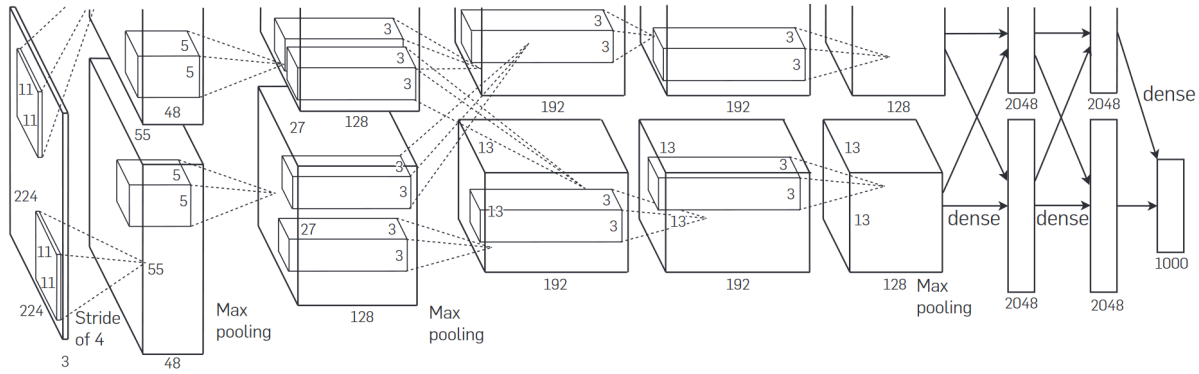

Figure 2.9: The *Alexnet* convolutional neural network for image classification. This feed-forward model takes as input a 224 by 224 pixels image with a depth of 3, for each of the colours in a pixel, and has 8 perceptron layers. The first five layers are convolutional with different kernel sizes to learn complex features of the image such as edges and shapes, while the last three layers are fully connected. Different pooling operations that reduce the depth of the arrays are performed between these layers, and the layers communicate with different parts of the image throughout the process. Adapted from [69].

However, to model sequential inputs such as text, a feed-forward architecture is not enough. The best Natural Language Processing (NLP) models make use of deep neural network architectures that can learn distant relationships between words in a sentence. State of the art results in sequence modelling, language modelling and machine translation have been held by Recurrent Neural Networks (RNNs), namely the Long-Short Term Memory (LSTM) and Gated Recurrent Unit (GRU) models, until the recent development of the Transformer model. These models can learn context, syntax and semantics of language due to their memory and attention capacity, respectively [70], [71], [72], [73].

Like all RNNs, the LSTM model possesses feedback connections between the layers, which allows the model to retain information during the processing of a sequential input, such as a sentence. This model, however, builds upon the simple feedback loop, and introduces several layers that increase its capacity to model long-distance relationships. An LSTM unit is described by a cell state, its memory component that is updated based on a new input vector and on the output of the previous pass, also called the hidden state. To update its cell state, the LSTM has a *forget gate*, which chooses the information that can be reset, an *input gate*, which chooses the information that will be stored, and an *output gate*, which chooses the information to send to the next unit. Different sigmoid and hyperbolic

tangent activation functions connect these gates to the operations used to update the cell state and to produce the output of each unit [74]. Several variations to the LSTM have been published, with more complex designs that have additional gates or *peephole connections* between the gates and the cell state (Figure 2.10), and more simplified designs, such as the GRU model that has the input and the forget gate coupled in a single *update gate*, are also widely used [75]. These models are, however, limited by the long training times, large memory requirements, and difficulty to capture very long-distance relationships [70].

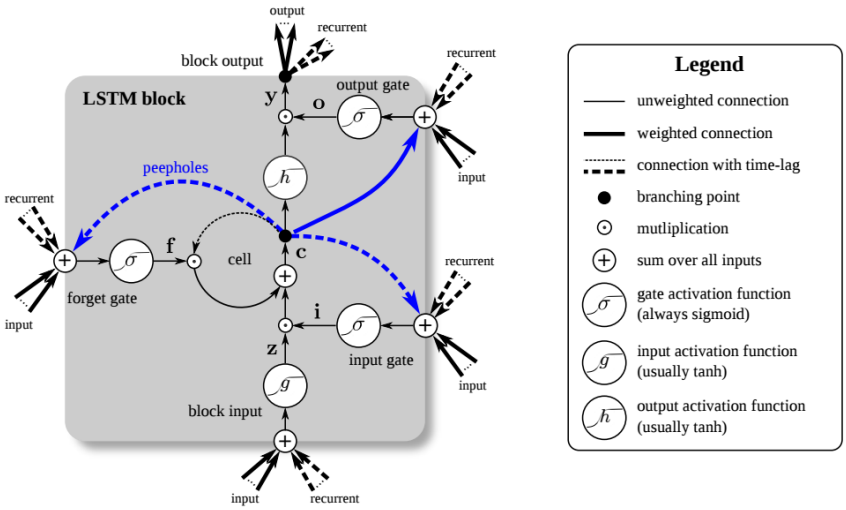

Figure 2.10: Schematic of the LSTM model with an additional *block input* gate and *peephole connections*, detailing the processing of an input, as it flows through the gates together with the previous hidden state (here noted as *recurrent*). The cell state is updated according to the result of the gate operations on the input and the previous hidden state, and its hidden state output is then used for the processing of the next input. Adapted from [74].

To combat these issues, the Transformer model relies entirely on an attention mechanism, without recurrence, which allows for more parallelization. This model has two main blocks: an encoder and a decoder. In the encoder, the entire sequence of inputs is mapped to an embedding space, taking into account their context and using attention mechanisms to determine the importance of each input. The decoder then takes these attention vectors, and constructs the outputs with information about which of the inputs it should focus on, which allows the model to capture long-distance relationships quite effectively [70]. However, due to their complexity, Transformer models have even longer and more memory intensive training steps. The most recent efforts have produced different kinds of attention models that take into account the memory limitations of the original Transformer network, such as the agglomerative attention model [76] and the Reformer model [77].

The most recent and successful applications of these models to NLP have been in the form of Bidirectional Long-Short Term Memory (biLSTM) networks, such as the ELMo model [71], increasingly larger left-to-right transformers, such as the Generative Pre-trained Transformer (GPT)-3 with up to 175 bil-

lion parameters [78], and Bidirectional Transformer networks, with less parameters but larger training data requirements such as the Bidirectional Encoder Representations from Transformers (BERT) model (Figure 2.11) [72].

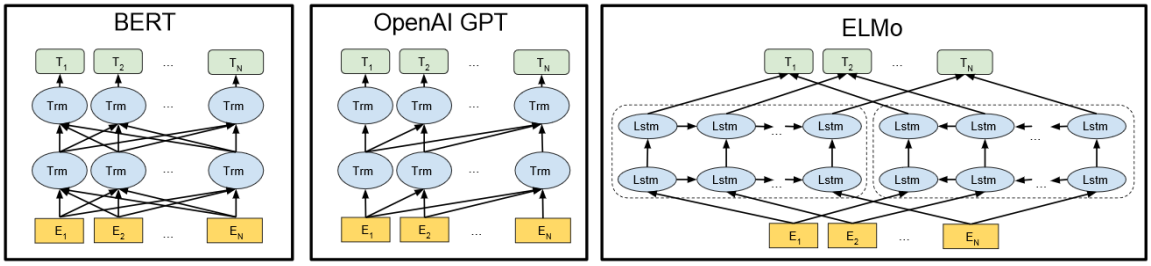

Figure 2.11: The pre-training of the BERT, GPT and ELMo language models, detailing how the processing of each input uses information from the other inputs in the sequence. The first one is based on Bidirectional Transformers, the second one uses a left-to-right Transformer network and the ELMo model uses independently trained biLSTMs. Adapted from [72].

The training of these models is done in two steps. In the first step, called pre-training, a self-supervised training is performed using large amounts of text data, where the model is tasked to predict a masked word in a sentence. Some models predict a masked word given the entire sentence, while other models read the sentence sequentially, predicting the next, masked, word. This step requires an extensive training data set, but self-supervision allows the use of unlabelled text data, which is more readily available. In the second step, called fine-tuning, the model is now trained for a specific supervised task, using the word representations it produces to perform diverse language understanding tasks. For this, a labelled data set is required so that the model can learn to classify the word representations accordingly, and although this training can be performed with a smaller data set, each different task requires a different labelled data set which is not always easily available. This supervised training step can also be performed using a separate model, that takes as inputs the word representations of the pre-trained (but not fine-tuned) language model. This allows for an easier development of diverse models to fit specific cases, which is usually called transfer-learning [11], [71], [72].

The success of these models in language tasks has also led to the successful use of transfer-learning, with language models trained on biological sequences, in several protein engineering tasks [17]

2.3.5 The ELMo language model and the SeqVec protein sequence model

ELMo is an auto-regressive model trained on big unlabelled text data sets such as Wikipedia to predict the next word in a sentence given all previous words in that sentence, and was shown to accurately learn syntax and semantics of language. This model is based on a biLSTM network with three layers, that produces a vector to represent each word, called an embedding. The embeddings of each word include information from all three layers, and are contextualized in the sentence, meaning that the same word can have different embeddings if it belongs to different sentences [71].

SeqVec is an adaptation of the ELMo model to deal with protein sequences (composed of amino acids), instead of sentences (composed of words) [11]. SeqVec was developed to evaluate the possible application of ELMo to proteins, and is trained on the *UniRef50* [9] data set to predict the next amino acid in the protein sequence, given all previous amino acids in that protein sequence.

As described by the authors, three obstacles arise from this application. Firstly, protein sequences can have very different lengths, and are also much longer than the average English sentence. This requires more memory, and some long-distance relationships might not be captured by the underlying LSTM layers. The second obstacle is due to the large difference in tokens: only 20 standard amino acids, compared to all possible words in the English language. A smaller vocabulary for a task with comparable complexity makes this application more difficult. Finally, *UniRef50* is close to 10 times larger than the largest NLP data set. This means that the model will need to be capable of absorbing much more information. After applying the necessary changes to handle these obstacles, the SeqVec model was found to capture secondary structure elements, regions of protein disorder and subcellular localization of the proteins, in spite of these difficulties [11].

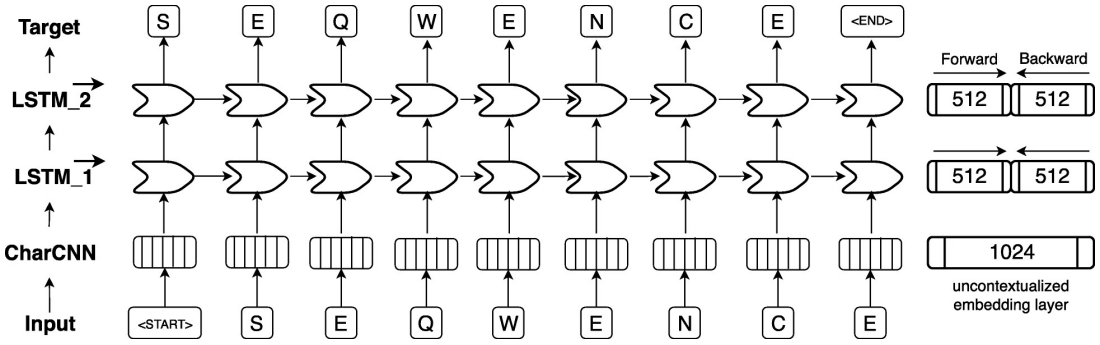

Figure 2.12: Schematic of the SeqVec deep learning model, as it takes a protein sequence as input and outputs contextualized high-dimensional representations for each of the amino acids in the sequence. This architecture, adapted from the ELMo language model for protein sequences, has an uncontextualized convolutional layer followed by two Bidirectional Long-Short Term Memory (biLSTM) layers that learn high-dimensional embeddings of each amino acid, taking into account their position in the sequence and capturing information from the other amino acids. This network was trained on the *UniRef50* data set to predict the next amino acid in the protein sequence, and was shown to capture aspects of protein biology only with self-supervised training. Adapted from [11].

The process of embedding a protein sequence is as follows: first, the amino acid sequence is padded with special tokens that indicate the start and the end of the sentence. Then, a context-independent Character-level Convolutional Neural Network (CharCNN) layer, which is usually applied in NLP to obtain vectors of fixed length from words with varying lengths, is here used solely to map each amino acid to a 1024-dimensional feature space without information from neighboring amino acids. Then, the output of this layer is used as input by a biLSTM layer that processes the amino acids sequentially, introducing context-specific information. Another biLSTM layer uses the output of this layer as direct input, and tries to predict the next amino acid in the sequence. Note that although the inference step of the biLSTM

layers incorporates the embedding of both the forward and the backward pass (in order to capture the context of the amino acid in both directions of the protein sequence), the forward pass needs to be trained separately from the backwards pass to prevent the model from knowing the masked word during training. The forward-pass of the processing of an example sequence by this model is shown in Figure 2.12.

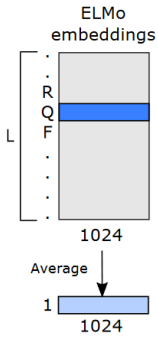

Figure 2.13: Dimensionality of the embeddings produced by SeqVec after processing a protein sequence. Each amino acid is represented by a 1024-dimensional embedding, and the embedding of a protein sequence with length L is a L by 1024-dimensional array. To represent a protein sequence by a single embedding independent of protein length, the average of the amino acid embeddings is used, resulting in a single 1024-dimensional array that represents a protein by a single point in this high-dimensional feature space. Adapted from [79].

Producing three 1024-dimensional vectors upon processing of an amino acid through the model, the authors sum the outputs of the three layers in a single 1024-dimensional vector that describes each amino acid residue in the sequence. As such, the process of embedding a protein sequence of length L results in a L by 1024-dimensional array of all the residue embeddings. Following the procedure usually performed in NLP, a protein sequence is represented in the embedding space by the average over all amino acid residue embeddings of the protein (Figure 2.13) [11]. The protein embeddings can then be used as features for diverse modelling tasks, but this ML approach has not yet been implemented in thermostability prediction.

3. Validation of the SeqVec model

Aiming to explore the biological meaning of the SeqVec embeddings and to validate the original paper's results, a data set was prepared for the visualization of biological properties in the embedding space and for the implementation of a secondary structure prediction algorithm.

3.1 Materials and Methods

3.1.1 Protein Data Bank secondary structure data set processing

The secondary structure data set from PDB was obtained directly from the database's *website* [34], and contains a total of 458764 protein sequences and their corresponding secondary structure information in the form of a *FASTA* file. To reduce the data set size while correctly evaluating the embeddings across the entire spectrum of available proteins, the Cluster Database at High Identity with Tolerance (CD-HIT) web suite [80] was used to remove proteins with high similarity from the data set. CD-HIT is a biological sequence clustering algorithm that is based on short word filtering and a greedy incremental clustering algorithm. It is used to manage large data sets by grouping homologous biological sequences and storing a representative sequence for each group [80]. The algorithm was applied with predefined parameters at a 50% sequence identity cut-off, producing a data set with 32948 representative protein sequences. The Structure Integration with Function, Taxonomy and Sequence database [81] was then used to obtain the Enzyme Commission (EC) numbers of the proteins in this data set, from which 26999 proteins with a EC number were stored and processed by SeqVec.

Two approaches were used to work with the three outputs of SeqVec. First, as proposed by the original authors, the outputs of the three layers were summed, to form a single feature vector with 1024 dimensions that describes each amino acid in the sequence. This was used for the study of protein function, where the protein embeddings were obtained by the sequence average of the amino acid embeddings. However, since the first layer involves no contextual information, and the third layer is optimized for the prediction task used in the training, using just the embedding of the middle layer of SeqVec to describe each amino acid was also studied, and better results were usually observed. This was used with the first 100 proteins (by alphabetical order, with a total of 23969 amino acid residues) to study the capacity of the embeddings to capture amino acid properties.

3.1.2 Machine learning models implementation

From the previously mentioned subset of 100 proteins, the amino acids of 15 of these proteins were isolated (3314 residues (14%)) from the remaining 20 655 residues (86%) of the other 85 proteins, to be later used as an independent testing set for a k-NN machine learning model trained in the 85 proteins to predict the secondary structure of each amino acid. These data partitions are described in Table 3.1.

Table 3.1: Description of the data partitions of the secondary structure data set prepared in Section 3.1.1, used for the separate training and evaluation of the k-NN machine learning algorithm, evidencing the imbalanced representation of the different classes in the data sets.

		Training subset		Testing subset	
Label	Description	Count	Percentage	Count	Percentage
H	Alpha helix	6306	30.5%	1372	41.4%
B	Isolated beta-bridge	248	1.20%	28	0.84%
E	Extended strand	4268	20.7%	448	13.5%
G	3-helix (3/10 helix)	734	3.55%	118	3.56%
I	5-helix (pi helix)	Not represented		Not represented	
T	Hydrogen bonded turn	2205	10.8%	363	11.0%
S	Bend	1996	9.66%	239	7.21%
-	None	4898	23.7%	746	22.5%

Since the performance of this algorithm is severely impacted by high dimensionality [51], PCA was fit to the training set (with a 40.2% explained training data variance) and used to reduce both data partitions to 100 components. In spite of the reduced explained variance percentage, experiments with this classifier using different numbers of principal components did not show significant improvements. The models were evaluated in terms of CM analysis, Accuracy (Acc), F_1 score and ROC curve analysis. For this experiment, the F_1 score was extended to multi-class classification by using the F_1 score of each class to find a weighted average of this score that takes into account the number of records of each label, and the ROC analysis was implemented by producing the ROC curves of each class in a one-vs-rest approach.

The only hyper-parameter of this model, k, was determined with a cross-validation approach as described in Appendix A.1.2, where the optimal value of 25 nearest neighbours was chosen based on a mean cross-validation F_1 score average of all classes, weighted according to the number of records with each label, of 0.508, the best score observed (Figure A.9).

3.2 Results

3.2.1 SeqVec successfully captures protein and amino acid properties

To confirm the capacity of the SeqVec model to capture biological properties, the embeddings generated from the protein sequences in the data set obtained in Section 3.1.1 were projected to two-dimensions by t-SNE (preceded by PCA to 100 components, with 78.84% explained data variance), and were visualized according to their EC class (Figure 3.1). The representation obtained shows several small isolated clusters that are representative of enzyme class, with a mixed and uninformative central cluster.

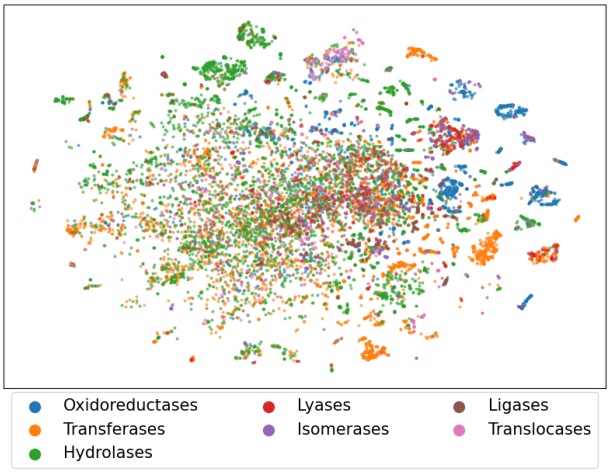

Figure 3.1: t-SNE projection to two dimensions of the protein embeddings from the data set obtained in Section 3.1.1, coloured by EC number of the proteins (x-axis: t-SNE 1; y-axis: t-SNE 2). The projection shows small isolated clusters that are representative of protein function.

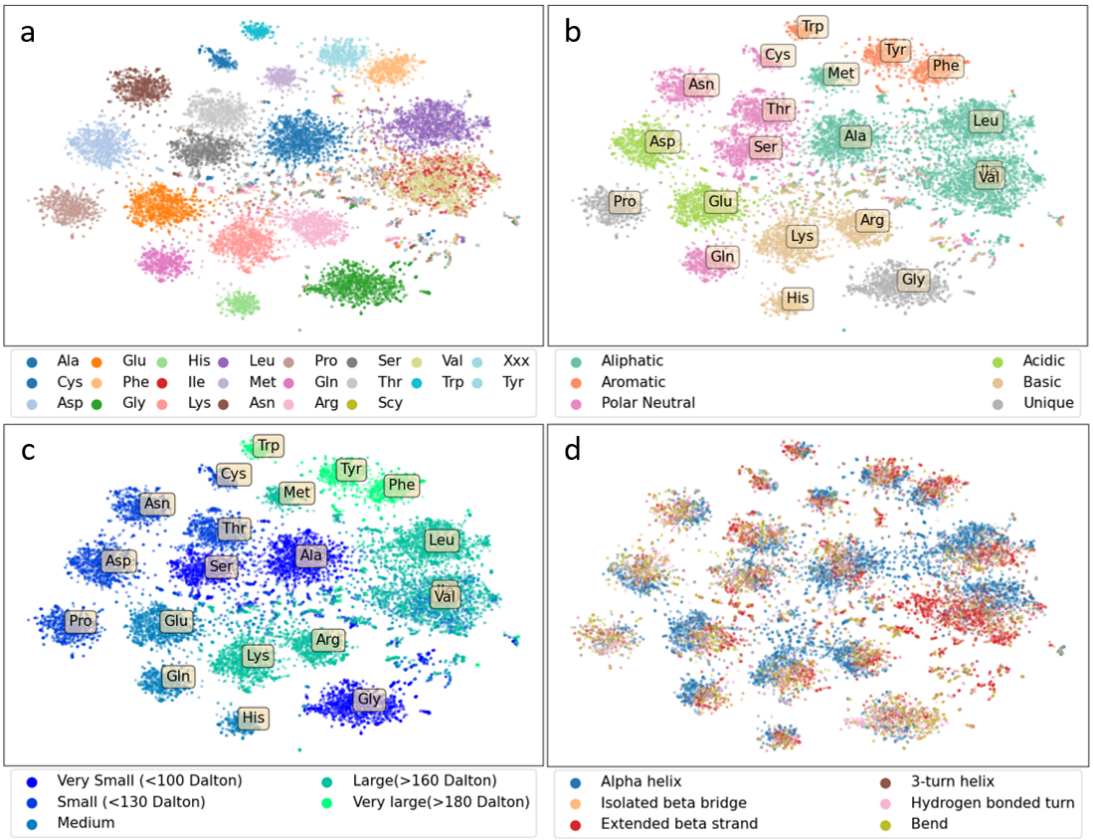

Figure 3.2: t-SNE projection to two dimensions of the amino acid embeddings from the data set obtained in Section 3.1.1 (for each figure, x-axis: t-SNE 1; y-axis: t-SNE 2). (a) Colouring by amino acid type shows a clear distinction between most amino acids. (b) Colouring by amino acid physicochemical properties shows that similar amino acids tend to cluster together. (c) Colouring by amino acid size shows that amino acids of similar size are more similar in the embedding space. (d) Colouring by the secondary structure label of the amino acids shows some separation between different structures.

The same two-dimensional projection procedure was done on the amino acid embeddings (by t-SNE, preceded by PCA to 100 dimensions with 39.68% explained data variance), showing that these group each amino acid type in mostly separate clusters, indicating that the model learns how to identify each amino acid (Figure 3.2 a). The small differences between embeddings of the same amino acid are due to the different protein sequence contexts captured by the model. Colouring the same plot by amino acid properties demonstrates that amino acids with similar physicochemical properties tend to be closer in the embedding space than amino acids with different properties (Figure 3.2 b), confirming that SeqVec correctly models aspects of biochemistry. Additionally, the same embeddings were also coloured by amino acid size and by the secondary structure in which the amino acid is found, revealing that these properties are also captured by the model, as amino acids with similar size are mostly grouped together in the embedding space, and also that the larger embedding clusters are mostly separated in smaller clusters that are representative of the secondary structure label of the amino acids (Figure 3.2 c and d).

3.2.2 SeqVec embeddings can be used for secondary structure prediction

To evaluate the applicability of the SeqVec embeddings for protein engineering prediction tasks, a k-NN classification algorithm was applied to predict the secondary structure label of amino acids (Section 3.1.2). Evaluation of this model in the testing set produced an accuracy score of 57.8% and an F1 score (weighted average based on class support) of 54.9%. Analysis of the model's CM and ROC curves show a good performance of the model for the most common labels and that it can still maintain a better than random performance on the most difficult classes (Figure 3.3).

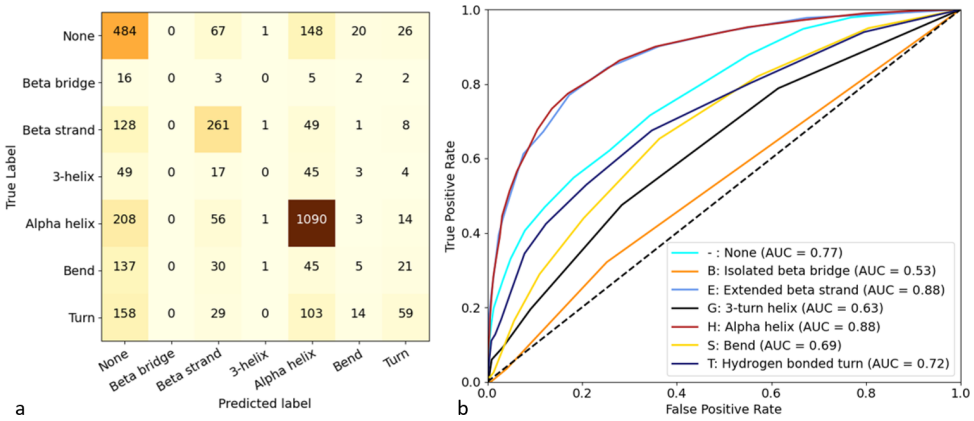

Figure 3.3: Performance of the k-NN secondary structure predictor on the testing set. (a) Confusion matrix, detailing the predictions of each label according to their true labels; (b) ROC curves of each class, obtained by a one-vs-rest approach, detailing the TP rate as a function of the FP rate at different decision thresholds. Although the predictions are very biased to the most represented classes, the model performs better than a random prediction for all labels, and has very good performances in the two most common secondary structures.

3.3 Discussion

As was observed by the authors of the SeqVec model, the unsupervised embeddings learned by the ELMo model trained on protein sequences contain biological information which can be used to model aspects of protein biochemistry [11]. Such conclusions were also obtained by other authors of relevant deep learning protein embedders such as *UniRep* [14], *D-SPACE* [10], the bidirectional transformer as published by [15], and the transfer-learning repository published by [17].

The obtained t-SNE projection of the protein embeddings from the PDB data set is similar to the authors' results with the same procedure on the *SCOPe* data set. The performance of SeqVec in protein-level prediction tasks, such as cell location and structural class, was shown by the authors to be close to current state of the art methods. By obtaining equivalent results in this small experiment, the previously mentioned conclusions of the authors are further cemented, given the generalizability of the results to different data sets.

On the other hand, the authors' application of the SeqVec residue-level embeddings for amino acid prediction tasks was not as good as current state of the art methods, with no further visualization analysis. In our exploration efforts we observed that, with a similar procedure as above, the SeqVec amino acid embeddings show the capacity to learn the physico-chemical properties of the amino acids, as well as some indication that they can be used for secondary structure prediction tasks, with clusters representative of the secondary structure labels. However, due to time limitations, this experiment used a reduced data set, with only 100 proteins, and although it was processed to remove protein sequences with at least 50% sequence similarity, it may not be representative of the diverse range of proteins that are found in nature.

The success of the k-NN algorithm implemented to predict the secondary structure label of the SeqVec amino acid residue embeddings is also dependent on an analysis of its performance metrics, and on a comparison with other established methods. We obtained an accuracy score of 57.8% that is inferior to the best application of SeqVec by the authors, which implements a deep learning model with evolutionary profiles together with the amino acid embeddings, and obtained an accuracy of 64.1%, and even this method was inferior to the state of the art secondary structure prediction method *NetSurfP-2.0*, which was applied by the SeqVec model's authors and obtained an accuracy score of 71.1%. Another of the previously mentioned transfer-learning effort used for secondary structure prediction is the bidirectional transformer, which shows similar performances, with an accuracy score of 60.8%.

First, the different data sets used for the training of the models need to be discussed. In our experiment, 85 random proteins with at most 50% sequence identity were used as a training set for the model, while the implementation of SeqVec for secondary structure prediction was trained on a data set with over 10000 different proteins, and this can already partly explain the inferior performance of our model.

On the other hand, the previously mentioned SeqVec implementation, as well as its comparison with the *NetSurfP-2.0* model, were evaluated in a subset of the *CASP12* data set with only 21 proteins, equivalent in size to our test set of 15 randomly chosen proteins with the reduced sequence identity guarantee of diversity. Also, one of the secondary structure labels was not presented to our model in either stage, which made this prediction task relatively more simple than the previously mentioned models. Additionally, although the F_1 score is not discussed by the original authors, our use of the weighted average of the F_1 score is expected to be overinflated, as this score is usually higher for the most frequent classes and was not adequately applied. A micro or macro average approach should have been used instead.

Second, the chosen ML algorithm needs to be compared to the complex deep learning method with evolutionary information of the SeqVec implementation, as well as with the convolutional and LSTM neural networks of *NetSurfP-2.0*. The k-NN is one of the simplest ML algorithms available, and was chosen to see to what extent could the secondary structure be predicted by only looking at the neighbours in the embedding space. The reduced training set and the accentuated class imbalance can impact this algorithm severely, which is the most likely cause for the inferior performance observed in our implementation. The fact that deep learning models trained with simple amino acid encoding methods by the SeqVec authors could reach at most 54.5% accuracy further supports this. It is, however, interesting that a simple k-NN algorithm that uses the SeqVec protein embeddings as features can compete with such models, as it suggests that the distance between two embeddings can be used to extract conclusions about their properties.

Finally, the individual class performances should also be compared, which was not done by the SeqVec authors. The three most represented classes are also the most accurately predicted, which was expected with the k-NN model. This could have been handled by using the class-weighted implementation of k-NN, but was not considered. Alternatively, by reducing the number of neighbours, k, the model could be less susceptible to class imbalance, but this hypothesis was not tested since this hyperparameter was determined by cross-validation. We could also formulate this problem as a three-class secondary structure prediction instead, which labels the records as helix, strand and other. This simpler prediction task is also usually performed, with increases in the accuracy of over 10% in the previously mentioned papers, and similar results are expected in our application.

Additionally, more effort could have been performed to explore individual protein sequences and their secondary structure prediction, because certain patterns might have arisen that could explain why this amino acid prediction task was not considered optimal by the authors, facilitating further efforts to improve this method for protein annotation.

4. Thermostability prediction with the ProTherm wild-type data set

The first effort to develop a model of protein thermostability directly from protein sequences was attempted with the *ProTherm* database. Using its wild-type data set, a ML regression model was implemented, using only the SeqVec protein embeddings as features, to test whether these encode thermostability information.

4.1 Materials and Methods

4.1.1 ProTherm wild-type proteins thermostability data set processing

The *ProTherm* database [25] is the main reference of most protein thermostability prediction models, and to date contains over 10000 records of wild-type proteins and their thermodynamic information, namely the free Gibbs energy of unfolding, ΔG, and the conditions of each experiment.

However, there are frequently multiple records per protein, and a closer inspection reveals that only 518 different proteins are present. Not only that, but some records are missing a ΔG annotation, and so only a total of 794 records, coming from a reduced sample of 119 different PDB identifiers, could be used.

In order to correctly use this database for the development of sequence-based thermostability prediction regression models, each record needs to be described by only one ΔG value, and all ΔG values also need to be in the same units. For these reasons, all ΔG records were converted to $kcal/mol$, and the mean ΔG value of each protein across all available experimental conditions was calculated, producing a data set with a ΔG distribution shown in Figure A.1.

The amino acid embeddings were obtained for each protein as the outputs of the middle layer of SeqVec, from which the protein embeddings were generated as the sequence average embedding of each protein. These were used to study the capacity of SeqVec to capture protein thermostability.

4.1.2 Machine learning models implementation

The data set obtained in Section 4.1.1 was discretized into 5 bins of equal ΔG intervals, and a random, stratified stratified split was performed, where 85% of the data was used for the training of a *Lasso* linear regression model, and the remaining 15% for an independent evaluation of the model on unseen data.

Given the small amount of data and the fact that each protein is described by a 1024-dimensional vector, PCA was applied to reduce the dimensionality of the data to 50 dimensions, fit to the training set (with an explained data variance of 94.33%) and used to reduce both partitions. The models were

applied using the PCA-reduced protein-level embeddings as features to predict the outcome variable ΔG, and were evaluated in terms of RMSE, r^2 score and EVS.

The constant α that is multiplied to the regularization term was manually chosen as 0.1, after several experiments with different values. No cross-validation hyper-parameter tuning was performed, given that in no attempt could we obtain promising results. This model was also generalized with polynomial features of degrees 2, 3 and 4, to model non-linear relationships in data, but more complex regression models were not attempted due to the small amount of data available and because data exploration efforts did not show much potential to capture the protein's ΔG value directly from these embeddings.

Considering the poor performances obtained with this implementation, additional approaches to incorporate the experimental conditions were implemented:

Removing the effect of T and pH on the data set: For all records in the data set, a linear regression of ΔG was fitted using just the T and pH as features, and then the residual of each record was removed from the data set.

Individually removing the effect of T and pH for each protein: For each protein, a linear regression of ΔG was fitted using just the T and pH of the records as features. The ΔG value for each protein was set as its individual regression's prediction at 25 ℃ and pH 7.

Using T and pH as additional features: Using all 794 records, T and pH were used as additional features for the machine learning model.

Calculating a weighted mean ΔG value for each protein: For each protein, the difference between the experimental condition of each record and the reference state of 25 ℃ and pH 7 was considered. Adapted from [30], records closer to this state were given a higher weight by applying Equation (A.1) to calculate the ΔG value. The presence of denaturing additives was ignored due to incoherent units in the database.

4.2 Results

4.2.1 SeqVec embeddings do not capture free Gibbs energy of unfolding

The protein embeddings generated from the wild-type thermostability data set prepared in Section 4.1.1 were projected to two-dimensions by t-SNE (predeced by PCA to 100 dimensions, with an explained data variance of 99.59%) and also by PCA directly (Figure 4.1 (a) and (b), respectively), and coloured by their ΔG label. Upon observation of the projections, neither the t-SNE nor the PCA approach show promising results, with no separation between proteins with different ΔG values. This suggests that the embeddings do not capture protein thermostability directly from the sequences in this data set.

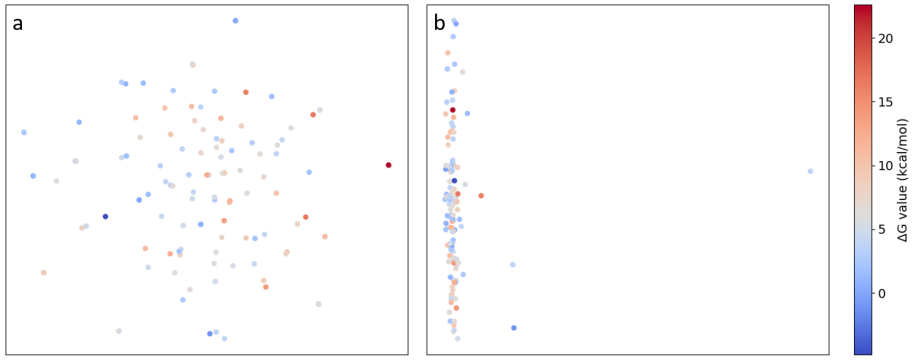

Figure 4.1: Two-dimensional projections of the protein embeddings obtained from the wild-type thermostability data set processed in Section 4.1.1 by (a) t-SNE (x-axis: t-SNE 1; y-axis: t-SNE 2); (b) PCA (x-axis: PCA 1; y-axis: PCA 2). Inspection of both figures shows that the protein embeddings do differentiate significant differences between proteins with high ΔG and low ΔG values.

4.2.2 Wild-type protein thermostability prediction was not successful

A linear regression model with different polynomial features was applied (Section 4.1.2) and evaluated for its capacity to predict the ΔG value directly from the protein embeddings. The best performing model used polynomial features of degree 3, and could achieve a test RMSE of 4.474, but the test set r^2 score of -0.036 and EVS of -0.027 are very poor. The negative predictive power of this model was compared to a baseline model, where a constant prediction of the data set's average ΔG produced a test RMSE of 4.088. This indicates that using the SeqVec protein embeddings to train a thermostability prediction model directly from sequence is not directly possible, which was studied further with additional approaches to including the experimental conditions (Section 4.1.2). The results of all attempts are summarized in the following Table 4.1.

The inclusion of the temperature and pH as additional features for the model produced the best results, with the lowest test RMSE and the highest r^2 and EVS scores of all the experiments, as expected since it uses all 794 records in the data set and includes large amounts of repeated protein sequences. The second baseline indicates that the temperature and pH by themselves do not include any relevant information for the prediction of protein thermostability, which can also be observed by the dispersed (and uncorrelated) plotting of the ΔG values as a function of each of these conditions (Appendix A.2.1). It is also noteworthy that the experiment where the effect of the experimental conditions was globally removed from the data set produced very similar ΔG values for all records, producing an almost constant prediction of the ΔG values, inaccurate in the testing set. In addition, removing the effect of the experimental conditions from each protein separately produced the worst model in this experiment, as a result of records with extreme experimental conditions being assigned unexpected ΔG values during the data processing. We can also observe that using the weighted average of each protein's ΔG values produced better results than a simple average calculation of this value, with similar RMSE values, but

positive r^2 and EVS values in the testing set, which were only achieved by the second baseline and the model which included the experimental conditions.

Table 4.1: Performance of the free Gibbs energy regression models using different pre-processing approaches to take into account the experimental conditions. Calculating a weighted mean of ΔG that gives more weights to experiments closer to physiological conditions was considered the best approach, because it was the only approach with positive r^2 and EVS that did not use repeated records.

Model	Features degree	RMSE train	RMSE test	r^2 train	r^2 test	EVS train	EVS test
Average ΔG baseline	1	6.898	4.088	-1.621e-04	-0.015	1.110e-16	2.220e-16
T and pH only baseline	1	6.771	4.009	0.036	0.024	0.036	0.049
Mean ΔG	3	3.612	4.474	0.194	-0.036	0.194	-0.027
Weighted mean ΔG	2	3.950	4.592	0.190	0.035	0.190	0.048
Remove T and pH effect from the entire data set	1	1.249	7.860	0.000	-0.229	0.000	0.000
Remove T and pH effect from each protein	1	6.515	8.455	0.139	-0.175	0.139	-0.171
Including T and pH as additional features	3	4.680	3.643	0.546	0.461	0.546	0.463

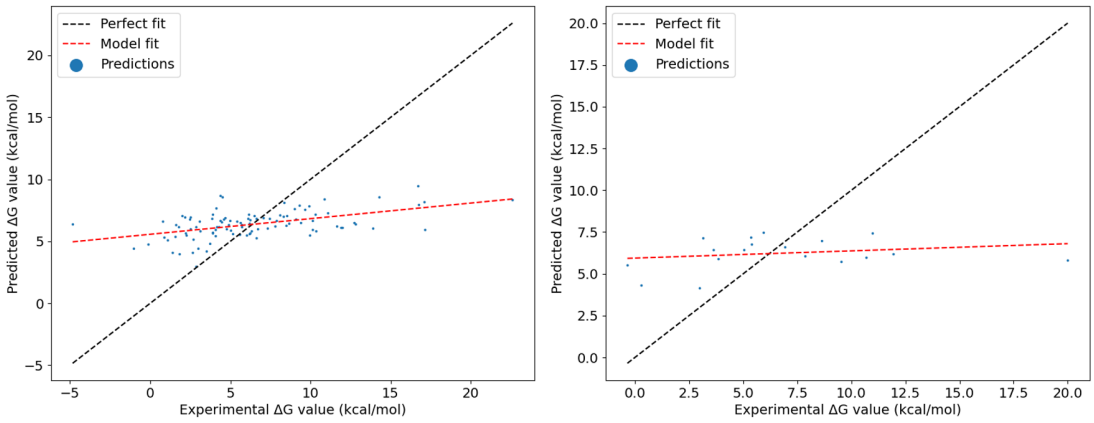

Figure 4.2: Scatter plots of the predicted free Gibbs energy of unfolding values (y axis) and their true values (x axis). The performance on the training set (left) shows that the regression fit is not dispersed, but the trend is too horizontal and indicates that the SeqVec features do not accurately capture the ΔG values of the proteins. The performance on the testing set (right) further proves this, showing an uninformative trend.

In a final effort to extract conclusions from this model, scatter plots of the predictions of this model in the training set and in the testing set were produced (Figure 4.2), from which we can observe a positive correlation and overall positive slope in both data sets, indicating that this model can in fact capture some thermostability information. However, the decrease in performance between training data and testing data, and the almost horizontal regression fit in the testing data further prove the difficulty of developing such a machine learning protein thermostability predictor from this data set.

4.3 Discussion

Prediction of the free Gibbs energy of unfolding of wild-type proteins is not usually performed directly from sequence, as this procedure is usually based on additional structural information, or based on physicochemical models of amino acid interactions (Section 2.2.4). In this experiment, the *ProTherm* database of wild-type proteins proved to be unusable for the development of a ML model that uses the SeqVec features to predict protein thermostability directly from sequence. This can be a result of three steps of the process: inadequate data processing, inadequate features or inadequate regression models.

The preparation of the *ProTherm* data set of wild-type proteins and their respective free Gibbs energy of unfolding annotations confirmed the main problems highlighted with this database by other authors: incomplete annotations, wrong values and different experimental conditions. The multiple efforts to remove and/or incorporate the experimental conditions from the data set proved unsuccessful, because of the previously mentioned issues as well as because of the small amount of records available. Not much could have been done with only 119 different protein sequences.

Visualization of the SeqVec protein embeddings generated from the processed data set did not suggest that these were capable of capturing elements of protein thermostability. This should not be a result of the reduced data set size, because similar t-SNE projections during the development of the previous experiment, using reduced samples of the data available, produced similar results to those obtained using the entire secondary structure data set. Most likely, the SeqVec embeddings do not capture the thermostability information required for the development of a ML prediction model. The use of a larger data set is required to affirm this, as the correlation obtained by the polynomial regression with the SeqVec protein embeddings was positive, and although mostly horizontal, indicates that perhaps with more records a more accurate predictor could be developed.

Finally, using the experimental conditions as additional features for the model could, in fact, be more adequate than averaging the replicates. This method allows the use of more records, and should not have been discarded because of this, since the use of experimental conditions as predictive features is also frequently used in literature on thermostability engineering (Section 2.2.4).

5. Prediction of thermostability changes with the ProTherm single-mutants data set

Instead of modelling protein thermostability directly from sequence, the prediction of changes to protein thermostability as a result of point mutations is more frequently used in protein engineering. To assess the usefulness of the SeqVec embeddings in the development of such a model, a thermostability data set was compiled. The amino acid embeddings of the obtained protein sequences were explored for the development of predictive features for several ML models, from which the most promising were studied in detail.

5.1 Materials and Methods

5.1.1 ProTherm single-mutants thermostability data set processing

The *ProTherm* database also contains to date over 12000 records of single-mutant proteins and their thermostability properties, namely the change in free Gibbs energy of unfolding caused by a mutation, $\Delta\Delta G$, with detailed experimental conditions. With similar problems to the wild-type data set, this database has a reduced number of records with $\Delta\Delta G$ information, and most of those do not possess the temperature, pH or denaturing additives detailed correctly.

However, as the largest and most commonly used protein thermostability database, most prediction models published to date use data from *ProTherm*, some of which making their pre-processed data sets publicly available. Looking to use as much data as possible, we used the protein thermodynamic data sets made available by the iStable 2.0 [8] and the PremPS [31] prediction models. These data sets are described in Table 5.1, and to our knowledge are the largest and most updated data sets from *ProTherm*.

Table 5.1: Data sets of protein thermostability changes upon single mutations collected for this work. The data sets S3568 and S640 were obtained from [8] and the data sets S2648 and S921 were obtained from [31].

	Data set S3568		Data set S640		Data set 2648		Data set S921	
	Count	Percentage	Count	Percentage	Count	Percentage	Count	Percentage
Positive labels ($\Delta\Delta G > 0$)	898	25.2%	173	27.0%	568	21.5%	287	31.2%
Negative labels ($\Delta\Delta G < 0$)	2669	74.8%	467	73.0%	2080	78.5%	634	68.8%
Number of proteins	150	-	39	-	131	-	195	-

As all data sets originate from the same database, there is a large overlap between them. With this in mind, duplicate records were removed, and a unique $\Delta\Delta G$ value per record was calculated.

After this processing step (Appendix A.2.2) we are left with 3706 unique records from 305 different proteins, without redundancy from experimental conditions. This data set was named S3706, and its $\Delta\Delta G$ distribution is shown in Figure A.3.

The SeqVec model was used to process the wild-type and the mutant sequences of the records in this data set, from which the amino acid residue embeddings were obtained as the output of the middle layer of the model. These were used for the development of different combinations of features to describe each mutation record, that were then used to implement several ML algorithms to predict the thermostability change caused by the mutations.

5.1.2 Machine learning models implementation

The S3706 data set was manually split in a partition with 3272 mutation records, called S3272, for the training of sequence-based protein thermostability ML models, and another with 434 records, called S434, used for an unbiased evaluation of the models (Table 5.3). The test set was composed solely of mutation records originating from S921, as this was the most varied data set, with records from almost as many different proteins as the training set. Note that this split was done manually to avoid a random split, so that the models could be evaluated in entirely different wild-type protein sequences.

Table 5.3: Description of the data partitions of the data set S3706 prepared in Section 5.1.1, used for the separate training and evaluation of the machine learning algorithms, evidencing the imbalanced representation of the positive and negative classes of records.

	Entire data set (S3706)		Training subset (S3272)		Testing subset (S434)	
	Count	Percentage	Count	Percentage	Count	Percentage
Total number of records	3706	-	3272	-	434	-
Number of records with positive label ($\Delta\Delta G > 0$)	855	23.1%	648	19.8%	171	39.4%
Number of records with negative label ($\Delta\Delta G < 0$)	2851	76.9%	2588	79.2%	263	60.6%
Total number of proteins	305	-	155	-	150	-

Since this data set is related to single mutations in proteins, each mutation record is represented by a pair of protein sequence embeddings, and on a first approach a simple subtraction between the wild-type and the mutant embedding vectors was used to describe each mutation. However, a total of 10 different feature sets were generated, representing each record by different combinations of its pair of embeddings. These are described in detail in Table 5.5.

As in the previous experiment, due to the small amount of available data and its high-dimensionality, PCA was applied, where the first 250 principal components were chosen for each feature set. This choice was made with the first feature set, where over 90% of the data is explained with 250 dimensions,

and was maintained for the other feature sets, despite the different dimensionalities, with the objective of exploring better feature combinations without increasing the complexity of the models, at the cost of having a lower explained data variance percentage on the feature sets with more dimensions.

Table 5.5: Description of the different feature sets produced to represent the mutation records in the high-dimensional embedding space, with the full dimension size and the percentage of explained data variance in the PCA-reduced features. The *Diff_5* feature set produced the best results.

Feature set name	Wild-type and mutant protein sequences represented by	Mutation record represented by	Full dimension of each record	PCA reduction to 250 dimensions explained data variance (%)
Diff	Average of all residue embeddings	Subtraction of MT to WT protein representation	1024	91.14
Diff_0	Residue embedding at substitution location	Subtraction of MT to WT protein representation	1024	88.87
Diff_2	Average of residue embeddings in a window of 2 residues to each side of the mutation location	Subtraction of MT to WT protein representation	1024	84.49
Diff_5	Average of residue embeddings in a window of 5 residues to each side of the mutation location	Subtraction of MT to WT protein representation	1024	83.35
Diff_10	Average of residue embeddings in a window of 10 residues to each side of the mutation location	Subtraction of MT to WT protein representation	1024	83.48
Diff_20	Average of residue embeddings in a window of 20 residues to each side of the mutation location	Subtraction of MT to WT protein representation	1024	85.10
Concat	Average of all residue embeddings	Concatenation of MT to WT protein representations	2048	99.81
ConcatDiff	Average of all residue embeddings	Concatenation of Diff to WT protein representations	2048	99.58
MeanSTD	Mean and standard deviation of all residue embeddings	Subtraction of MT to WT protein representation	2048	92.36
Moments	Mean, standard deviation, variance, minimum and maximum value of all residue embeddings	Subtraction of MT to WT protein representation	5120	78.64

These features were explored for their usefulness with a base set of classifiers constituted by a logistic regression with predefined parameters, a SVM with linear kernel, polynomial kernels of degree 1, 2 and 3, and with a RBF kernel, and a k-NN algorithm with 1, 3, 5, 9, 15, 25 and 50 nearest neighbours. These models were chosen based on two decisive factors: model complexity and training time required, both due to the limited data available in order to avoid overly complex models that would risk overfitting the training set, and also because this allows a simpler hyper-parameter tuning.

With each feature set, the base set of classifiers was trained on the S3272 data set and tested on the S434 data set, from which the model with highest MCC was chosen to compare the feature sets. From this experiment, the *Diff_5* feature set showed the best overall results, with a high MCC and highest precision scores when used with the logistic regression, and was the only feature set used further. These

models were also evaluated in terms of Acc, recall (TP rate), specificity, F_1 score and ROC AUC, but these metrics were not analyzed in detail (Table A.1).

In parallel, attempting to improve the prediction performance further, the same comparison of features was performed with two different data pre-processing procedures. In the first procedure, aiming to remove unnecessary data and only train the model on relevant records, the mutations with an insignificant change to $\Delta\Delta G$ (between -1 and 1 kcal/mol) were removed from the training set. In the second procedure, aiming to balance the training data set, the reverse mutations were simulated and added to the data set, by switching the wild-type and mutant sequences, and inverting the $\Delta\Delta G$ value. By training the same basic classifiers on the filtered data set, evaluation of the models on the isolated testing set shows a general increase in precision but at the cost of a lower MCC (Table A.2), and when the balanced training set was used, a general increase in MCC was observed in the testing set, but at a lower precision (Table A.3). These processing procedures were not studied further.

With the original training data set S3272 and the *Diff_5* feature set, the two best performing classifiers were the logistic regression and the linear SVM models. A cross-validation procedure as described in Appendix A.1.2 was then applied using the training data set, with none of the additional pre-processing measures above, to determine the best hyper-parameter values for the two algorithms, based on the mean cross-validation MCC.

For the logistic regression, the l_1 and l_2 penalties were both attempted with different regularization strengths, from which a l_2 penalty with a C value of 1438.45 provided the best mean cross-validation MCC of 0.150 (Figure A.10).

For the linear SVM, the primal formulation and the dual formulation of the problem were both attempted as well as the standard SVM loss function and its squared formulation, and the l_1 and l_2 regularization types were both applied with different regularization strengths, from which a squared loss function with a l_1 penalty and a C value of 50 provided the best mean cross-validation MCC of 0.133 (Figure A.11).

The fine-tuned models were also studied in terms of the CM, ROC curve analysis, and by a PRC analysis. However, most of the analysis of the models was based on the MCC and precision score. The first one was chosen because it is the most suitable metric to handle the imbalance in the data sets, and because the MCC takes into account the four parameters of the confusion matrix, encompassing in a way the Acc, TP rate and TN rate, where a model that is capable of accurately predicting the two classes will have a MCC close to 1. The second one was chosen because the effect of mutations in protein stability is most frequently negative and the prediction of a negative record as positive is not desired. By evaluating the precision score of a model, which takes into account the FP predictions to calculate the proportion of positive predictions that are correct, its capacity to correctly label the rare positive records can be determined.

To contrast the previous classification algorithms, the MLP model was also implemented, in an attempt to model more complex relationships in the data. This model was implemented using 1 and 2 hidden layers, again due to the limited size of the data set. Another exploratory effort using the RF model was also attempted due to its feature selection capacity, tendency to resist overfitting and robustness to hyper-parameter changes, but this did not show promising results and will not be discussed further.

The MLP was implemented with the ReLU activation function on all neurons except those in the output layer. The output layer of the MLP classifiers consisted of a single neuron with sigmoid activation function. In this work, the binary cross-entropy loss function was chosen, a frequently used loss function for binary classifiers, and the optimization algorithm used for the MLP models was the *Adam* algorithm, a stochastic gradient descent method. This frequently used and memory efficient algorithm is based on an adaptive estimation of both first-order and second-order derivatives of the gradients of the loss function [62].

Several hyper-parameters could be chosen and fine-tuned. A batch size of 50 was used in all experiments, with which a small model with 20 neurons in a single hidden layer was used to experiment different learning rates and training epochs, where it was soon observed that using more than 3 epochs resulted in serious overfitting issues, which was followed by leaving out 15% of the training set in a random stratified split for validation of the training process (Figure A.12). However, by using just 1 or 2 training epochs, the models performed very poorly, so the predefined learning rate of 0.001 was used, which resulted in the least overfitting when 3 training epochs were performed. The choice of number of hidden layers and neurons was made by applying the cross-validation procedure described in Appendix A.1.2, from which an MLP architecture with two hidden layers of 128 neurons each achieved the best mean cross-validation MCC of 0.167 (Figure A.13).

A baseline model was also developed. This model uses very simple features to describe the protein mutations: for both the wild-type and the mutant amino acid, a one-hot-encoding label of the amino acid types and of the physicochemical properties (aliphatic, aromatic, polar neutral, acidic, basic or unique), their molecular weights and hydrophobicity values, and the *BLOSUM62* value for the substitution were used. The *BLOSUM* amino acid substitution matrix is frequently used in local sequence alignment tools such as the Basic Local Alignment Search Tool [82] to score a amino acid substitution according to the probability that this substitution happens in homologous protein sequences, and where a higher score is given to more biologically frequent mutations. Due to the extensive use of categorical features, the DT algorithm was chosen and applied with the predefined parameters (from which the most relevant is the *Gini* impurity calculation for information gain, which is more computationally efficient).

5.2 Results

5.2.1 Combining the embedding pairs to describe mutation records

Using *SeqVec* to embed the wild-type and the mutant sequence of each mutation record in the data set obtained in Section 5.1.1, we obtain two embeddings to describe each mutation. The effect of the mutations in the proteins was first described as the difference between the wild-type and mutant embeddings (named *Diff* in Table 5.5), which can be projected into two dimensions by t-SNE (Figure 5.1, a) (preceded by a PCA with 100 principal components, with an explained data variance of 74.7%). This projection can also be obtained by performing PCA to two dimensions (Figure 5.1, b). Upon observation of the t-SNE plot, some separation between mutations with a positive and a negative effect in thermostability can be observed, indicating that these features are somewhat representative of the $\Delta\Delta G$ value of the mutations, but this is not found in the PCA projection.

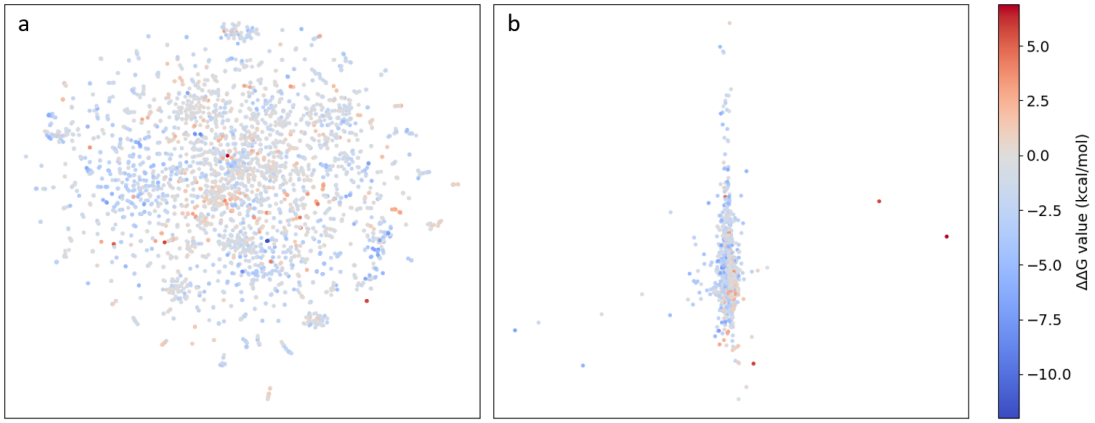

Figure 5.1: Projection to two dimensions of the mutation records, represented by the difference between the wild-type sequence embedding and the mutant sequence embedding by: (a) t-SNE (x-axis: t-SNE 1; y-axis: t-SNE 2); (b) PCA (x-axis: PCA 1; y-axis: PCA 2). The t-SNE projection reveals that there is some separation between mutations with a positive and a negative effect on thermostability.

A few preliminary experiments with the logistic regression, SVM and k-NN classifiers showed poor thermostability prediction performances when using this feature set (Section 5.1.2). The mutation records were then described by the *Diff_5* feature set, which represents each protein by the average of the embeddings of the amino acids in a window of 5 residues in each direction from the mutation, and then represents each mutation record by the difference between the wild-type and the mutant features.

With this feature set, the same two-dimensional projection procedure as above shows a better separation of classes in both the t-SNE (Figure 5.2, a) and the PCA (Figure 5.2, b) plots, which was not obtained before, and suggests that these features are more representative of the effect of the mutations in protein thermostability. Also, with these features, the base set of classifiers (Section 5.1.2) achieved better performances (Table 5.7).

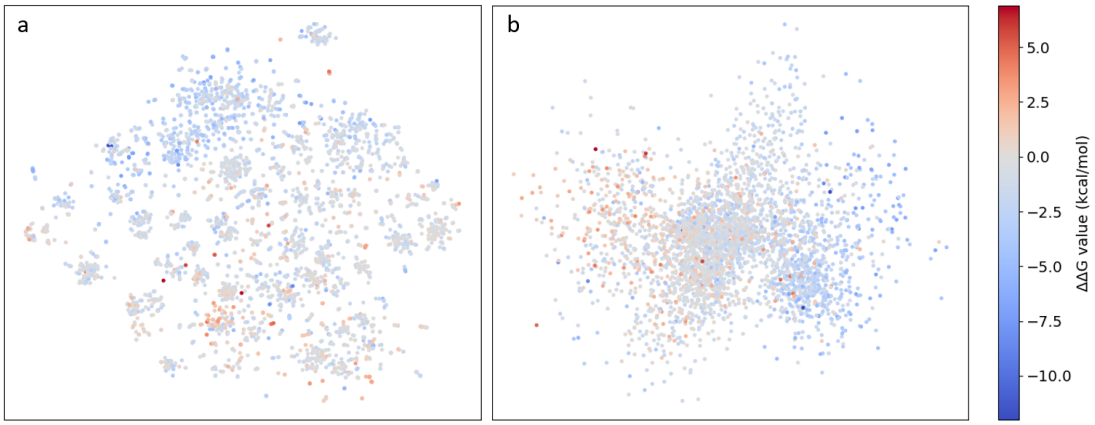

Figure 5.2: Projection to two dimensions of the mutation records, represented by the *Diff_5* feature set by: (a) t-SNE (x-axis: t-SNE 1; y-axis: t-SNE 2); (b) PCA (x-axis: PCA 1; y-axis: PCA 2). The t-SNE projection shows a better separation between mutations with a positive and a negative effect on thermostability, and only with this feature set could the same be observed with a PCA projection.

Table 5.7: Performance of the basic set of classifiers on the testing set S434, trained on the data set S3272 with the mutation records, represented by the *Diff_5* feature set. The models were evaluated for accuracy, recall, specificity, precision, MCC, F1 and ROC AUC scores. From this comparison, the logistic regression and the linear SVM algorithms show the most promising results, with overall best precision and MCC metrics.

Model	Acc	Sens	Spec	Prec	MCC	F1 score	ROC AUC
Log reg	0.66	0.15	0.99	0.93	0.28	0.25	0.78
Linear SVM	0.66	0.20	0.96	0.78	0.27	0.32	0.77
Poly 1 SVM	0.61	0.	1.	0.	0.	0.	0.71
Poly 2 SVM	0.61	0.	1.	0.	0.	0.	0.46
Poly 3 SVM	0.61	0.	1.	0.	0.	0.	0.8
RBF SVM	0.61	0.	1.	0.	0.	0.	0.72
1-NN	0.60	0.29	0.8	0.48	0.10	0.36	0.54
3-NN	0.63	0.29	0.86	0.57	0.18	0.39	0.63
5-NN	0.63	0.22	0.89	0.58	0.16	0.32	0.68
9-NN	0.64	0.22	0.92	0.63	0.20	0.33	0.68
15-NN	0.63	0.14	0.94	0.62	0.14	0.23	0.69
25-NN	0.63	0.09	0.98	0.75	0.16	0.16	0.70
50-NN	0.62	0.02	1.	1.	0.12	0.05	0.72

The logistic regression algorithm achieved the second best precision score (of 0.93 on the testing set), the third best MCC (of 0.28 on the testing set), and also one of the best ROC AUC scores (of 0.78 on the testing set) of all attempts. The linear SVM achieved the second best MCC of 0.27, while maintaining a reasonable precision score of 0.78. These models were chosen for an additional hyperparameter tuning.

5.2.2 The precision-MCC trade-off

Having chosen the best feature set and the most promising classifiers, the hyperparameters of the logistic regression and the linear SVM were tuned by cross-validation in the training set (Appendix A.1.2). The tuned logistic regression was able to achieve a MCC of 0.331, but at a cost on the precision score which decreased to 0.753. The tuned linear SVM also achieved a higher MCC of 0.318, but was capable capable of maintaining a similar precision score of 0.774. These models were also evaluated in terms of CM, PRC curve and ROC curve analysis (Figure 5.3 and Figure 5.4), with no large differences in performances except for the better PRC of the second model, with which the precision can be maintained at 1.0 for the largest decrease in the decision threshold.

These models were compared to a DT baseline model with basic features, which also performed better than random in the testing set, with an Acc of 0.64, TP rate of 0.15, precision of 0.73, MCC of 0.20, F_1 score of 0.24 and ROC AUC of 0.68 (and additional CM, ROC and PRC curves in Figure A.14). The two obtained models both outperformed this baseline, especially in terms of the PRC.

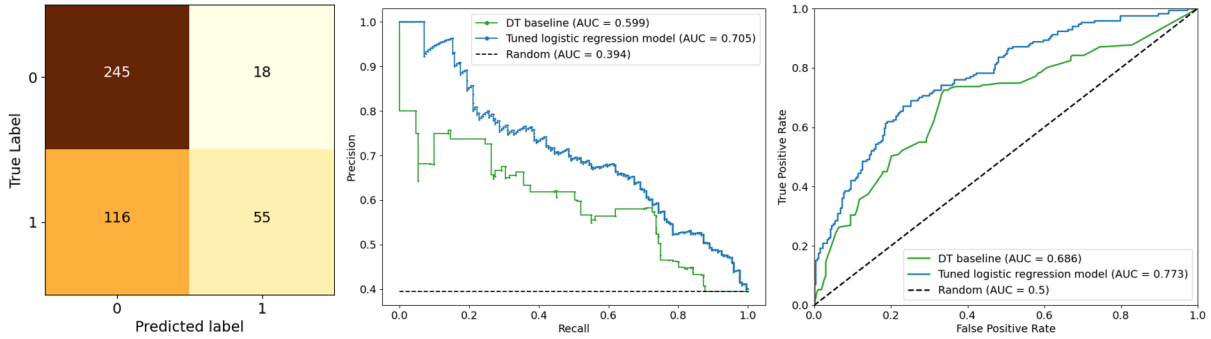

Figure 5.3: Confusion Matrix (CM) (left), Precision-Recall Curve (PRC) (middle) and Receiver Operating Characteristic (ROC) curve (right) of the tuned logistic regression, evaluated on the testing set S434 to predict the protein thermostability changes of single mutations. The CM shows a large bias towards negative mutations, but the PRC shows that the model is accurate in its most confident positive predictions. The ROC curve does not produce a meaningful analysis, although it shows that the model outperforms the DT baseline.

To contrast with the simple decision boundaries of the two previous linear models, an MLP was also implemented (Section 5.1.2). Compared to the previous models, the MLP shows the highest MCC found in all experiments (of 0.354 in the testing set), but shows a large decrease in precision (of 0.690 in the testing set). The very reduced positive predictive power of this model is evidenced by its PRC, which shows the inability of the model to accurately predict the positive records in which it has the most confidence (Figure 5.4, middle). This model, however, also outperforms the DT baseline.

The achieved MCC values are quite behind the state of the art *iStable 2.0* prediction model, which achieved a value of 0.708 for the same metric, but are better than the *PoPMuSiC* model with a MCC of 0.291 and the *MUpro* model, with a MCC of 0.248 [8].

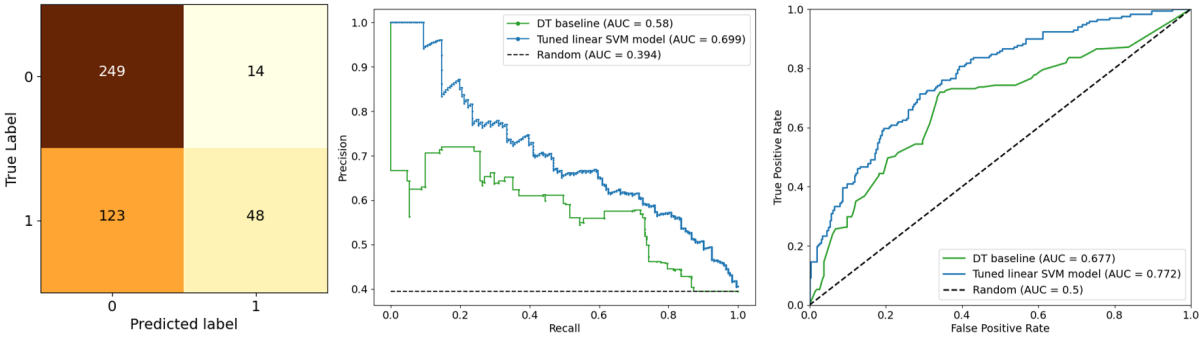

Figure 5.4: Confusion Matrix (CM) (left), Precision-Recall Curve (PRC) (middle) and Receiver Operating Characteristic (ROC) curve (right) of the tuned linear SVM, evaluated on the testing set S434 to predict the protein thermostability changes of single mutations. As with the logistic regression, this CM also shows a large bias towards negative mutations, while the PRC also shows that the model is accurate in its most confident positive predictions. The ROC curve does not produce a meaningful analysis, although it also shows that the model outperforms the DT baseline.

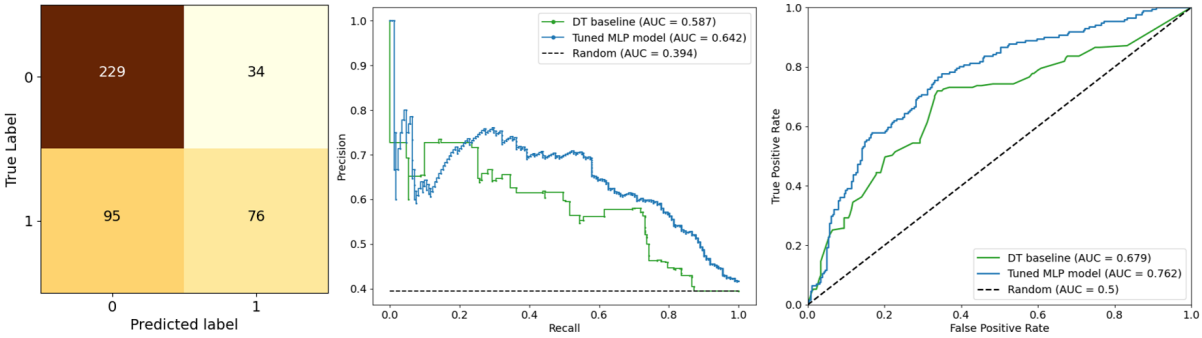

Figure 5.5: Confusion Matrix (CM) (left), Precision-Recall Curve (PRC) (middle) and Receiver Operating Characteristic (ROC) curve (right) of the tuned MLP, evaluated on the testing set S434 to predict the protein thermostability changes of single mutations. This model produced the less biased CM, but its PRC shows a very poor performance in predicting the positive class correctly, barely outperforming the DT baseline. This model also produced the worse ROC AUC.

5.2.3 Mutations to similar sequences are more accurately predicted

Given the importance of a correct classification of the positive records, the performance of the SVM model, which produced the best PRC, was studied as a function of protein sequence similarity. For this, the Basic Local Alignment Search Tool (BLAST) tool was used to query each of the wild-type protein sequences in the testing set against all the wild-type protein sequences in the training set. As expected from the diversity of protein sequences in the testing set, this procedure shows a varied percentage of sequence identities between the best matches (Figure A.15).

Using the result of the previous procedure, different subsets of the testing set were created, based on the sequence similarity percentage with the best match in the training set. A total of 4 subsets were generated, with sequence identities of 20 to 30%, 40 to 50%, 50 to 70% and 90 to 100%. The tuned linear SVM, which previously achieved the best precision, was evaluated on each of these subsets, and

the PRC curves were generated. These were re-scaled, from the range of values between 1 and the expected random prediction value, to a range of values between 1 and 0, since the representation of the classes was different in the subsets (Figure 5.6).

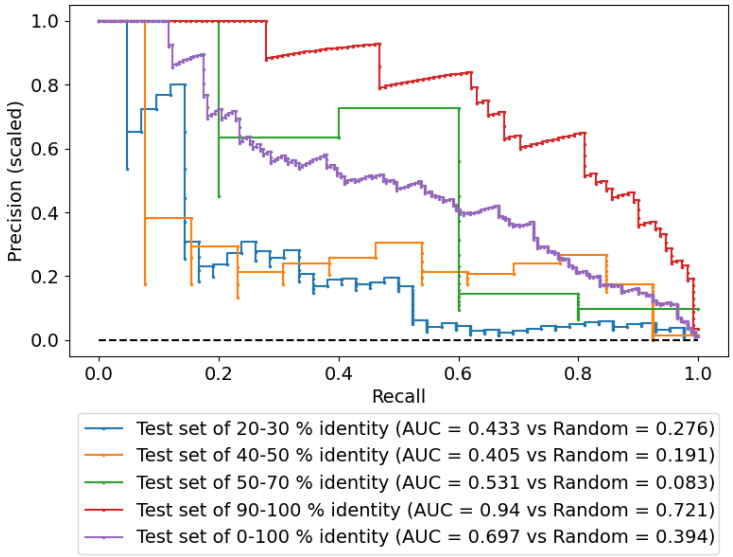

Figure 5.6: Precision-Recall Curves (PRCs) of the tuned linear SVM predictor of protein thermostability changes of single mutations, evaluated on different subsets of the testing set S434 with increasingly higher sequence identity percentages with the training set S3272. We can observe a trend of increasing precision performances when the model is tested on increasingly more similar sequences to those used during training.

A significant correlation between the sequence similarity and the precision of the model was observed, where a testing set with very similar sequences to the training set was more accurately predicted. This suggests that the model is more accurate in predicting mutations to protein sequences similar to those it was trained on.

5.3 Discussion

Unlike the prediction of the thermostability of wild-type proteins, the prediction of the effect that a mutation will have on a protein is not only more frequently studied, but also provides better results. This approach has more supporting literature, and the *ProTherm* database of mutation records is more extensive than for wild-type records. Overall, the SeqVec embeddings provided predictive features to develop ML models of $\Delta\Delta G$ changes upon single mutations that provide better performances than some well established models but still fall behind state of the art. This comparison is, however, not straightforward.

First, the data sets used in this experiment for the training and the testing of the machine learning models were different from those used by other models. Since different models frequently use different data sets, only the review papers that train and test each model with the same data sets present a valid comparison, and this was not the case since we compiled our own data sets. Even so, the data

sets presented in this work are the largest collection of thermostability records with no experimental condition redundancy, nor repeated records, and were thoroughly studied to prevent the testing set from containing proteins also present in the training set, which makes the performance of our models on independent data more reliable. On the other hand, considering the detailed data processing performed in this experiment, an additional processing step could have been useful where, as described by [30], mutant sequences with a $\Delta\Delta G$ above 5 kcal/mol were removed, as well as those involving a proline substitution, as these are likely to induce significant structural changes. This was not taken into account in the development of the data set.

The detailed performance evaluation of our models, namely the PRC, is also something that is not frequently seen in literature, and that allows the study of the model's most confident predictions. In protein engineering, it is desirable to perform as least mutations to a protein as possible, in order to avoid altering its fitness. Since each protein can be mutated in a wide number of different ways, from which only a few will result in real positive stability changes, a thermostability predictor does not need to accurately predict a lot of the records correctly (which would translate into a high MCC), as long as the ones it predicts as positive are correct (high precision score). If the most confidently predicted positive records are correct, such a model can still be interesting for application in a protein thermostability engineering procedure, and this was the reason for which the linear SVM was studied further.

The additional study of the Precision-Recall Curves (PRCs) in testing subsets of different sequence identities was also useful for this evaluation. Although it would have been ideal to see a model that can predict the positive class correctly independently of sequence identity to the training set, observing a correlation between sequence similarity and higher precision confirms that the model is learning biologically significant features that describe the effect of the mutations in the proteins, and can accurately generalize this knowledge to similar proteins.

It is also noteworthy that the models developed in this experiment used solely the SeqVec embeddings as features, with no additional structural features, nor even the explicit amino acid sequence of the protein. This approach, very different from the sequence or structure-based models seen in literature, skips the difficult step of generation of complex protein features of those models, which sometimes rely on other physicochemical models themselves. Capable of achieving MCC values that are relevant for recent literature, this opens a completely new approach to protein thermostability prediction with the helpful advantages associated with transfer-learning, namely the possibility to compare different proteins or mutations in the high-dimensional feature space.

A way to conclude about the usefulness of the embedding space for direct comparison of mutations could have been based on the two-dimensional projections generated for the features. The t-SNE plot showed some clustering that could be representative of specific amino acid substitutions, and an additional study of these could lead to useful conclusions about a positive or negative effect, depending on

their context in the entire protein sequence.

Finally, although several metrics were calculated, the comparison of the models was based on their MCC in the testing set. This proved to be adequate, as most models, with all feature sets, achieved an Acc score of over 60%, meaning that this metric would not have been useful with this data imbalance. The MCC was shown to successfully capture the most frequently used performance metrics recall, specificity and F_1 scores in a single value, making it the most useful metric to directly compare different models, where models with higher values of these also produced a higher value of MCC. In an additional note, the ROC curves of the models did not show significant differences between them, meaning that the ROC AUC measure is also not suitable for the comparison of the models by itself.

Overall, a trade-off between increasing the MCC at a decrease on the precision score, and vice versa, was observed. In no attempt could the models maintain both of these scores elevated, suggesting that, when using the SeqVec features for the prediction of protein thermostability upon point mutations, a choice needs to be made on a very precise model that misses a lot of potential positive mutations (to which the obtained linear SVM would be more appropriate), or a model that can overall separate the two classes of mutations but might suggest some negative mutations as positives (to which the logistic regression or the MLP model are more appropriate). The additional pre-processing steps attempted further proved this, as the filtering step increased the precision of the models but reduced their capacity to differentiate the two classes, and the balancing step increased the MCC values but at a cost on precision.

In general, it was expected that this experiment would find better results, as the prediction of the effect that a mutation can have in protein thermostability is more easily modelled than the prediction of protein thermostability directly from sequence. However, the competitive performances obtained by this implementation makes this experiment a success.

6. The effect of mutations in the SeqVec embeddings

Seeing that the effect of a mutation in the thermostability of a protein is better modelled by SeqVec when only the embeddings of amino acids close to the mutation are used to generate the protein embedding, it became interesting to study the effect that mutations have in the embeddings of the entire amino acid sequence.

6.1 Materials and Methods

6.1.1 Describing the effect of a mutation in the embeddings

To describe the effect of a mutation in the amino acid embeddings, the euclidean distance between the wild-type and the mutant sequence amino acid embeddings of each protein was used, originating a sequence of euclidean distance values with the same length as the protein, describing the effect of the mutation in the embeddings throughout the sequence.

This was used to study the mutation records in the S3706 data set (Section 5.1.1), as well as a mutagenesis data set, prepared from this data set to simulate a larger amount of mutations, with additional information about binding site locations in the proteins.

6.1.2 Mutagenesis data set preparation

A subset of 84 proteins from the S3706 (Section 5.1.1) data set was used to compose another single-mutants data set, used to study the capacity of SeqVec to model long-distance relationships between amino acids. These proteins were chosen arbitrarily from the S3706 data set with two criteria: availability of binding-site information in the PDB database, and sequence length smaller than 250 amino acids. For each protein, a single-mutant sequence was generated by changing each amino acid in the protein individually, generating the same number of single-mutant sequences as the protein's sequence length. For example, with protein *1ANK*, with a sequence length of 214 amino acids, 214 different single-mutant sequences were generated.

Although each amino acid could be mutated to any of the other 19 standard amino acids, only one mutation was simulated for each amino acid. The mutant amino acid was chosen according to the *BLOSUM62* matrix [83]. Each amino acid was mutated according to the highest scoring substitution, or randomly between the highest scoring substitutions, under the hypothesis that this would make the mutations as equivalent as possible, and allow us to focus entirely on the mutation location and its effect on the protein sequence as a function of the position. For each wild-type and mutant sequences, the

amino acid embeddings were obtained as the sum of the output of the three layers of SeqVec. The distribution of euclidean distances in this data set is detailed in Appendix A.2.3.

This effect was studied both in terms of distance in the amino acid sequence and also in terms of 3D distance between the amino acids in the wild-type protein conformation. This information was obtained, for each protein, from the *mmCIF* files obtained from PDB, which contain the Cartesian coordinates of each atom in the protein. By calculating the distance between the central carbon atoms of the amino acid residues, a distance matrix was produced for each protein and used to calculate the distances between each amino acid and the mutation location of all sequences.

6.1.3 The Wilcoxon rank-sum statistical test

Also called the Mann–Whitney U test, the Wilcoxon rank-sum statistical test was used to verify the statistical relevance of differences in the euclidean distances between the embeddings. It tests the null hypothesis that two sets are uniformly mixed ($P(x > y) = P(x < y)$), against the alternative hypothesis that samples from one of the sets are more likely to be larger than the other, and is adequate for the comparison of continuous variables [84].

This test assumes that both sets of variables are from the same distribution, and after arranging the two sets in order, counts the number of times that a sample from one set is larger than a sample from the other set. By comparing this value with the expected value of this count, obtained from the distribution assumption of each of the sets, this test returns a U test statistic that represents this difference and the associated p-value, which measures the significance level with which the null hypothesis is rejected [66].

6.2 Results

6.2.1 Distant amino acid embeddings capture the effect of a mutation

Firstly, the mutation records in the S3706 data set were explored, from which a few arbitary records were plotted in terms of euclidean distance between the wild-type and the mutant sequence embeddings (Section 6.1.1), showing that the highest difference was usually concentrated in and close to the mutation location, but that in some cases there was also an effect in distant amino acids (Figure 6.1, a). This prompted an effort to visualize all of the records (Figure 6.1, b), showing that quite frequently there is also a strong effect in amino acids distant from the mutation. Note that except for the mutated amino acid in the center of the x axis, all other amino acids are of the same type in the wild-type and in the mutant sequences, showing the capacity of the SeqVec model to capture the different contexts, as a consequence of a mutation in another amino acid of the protein.

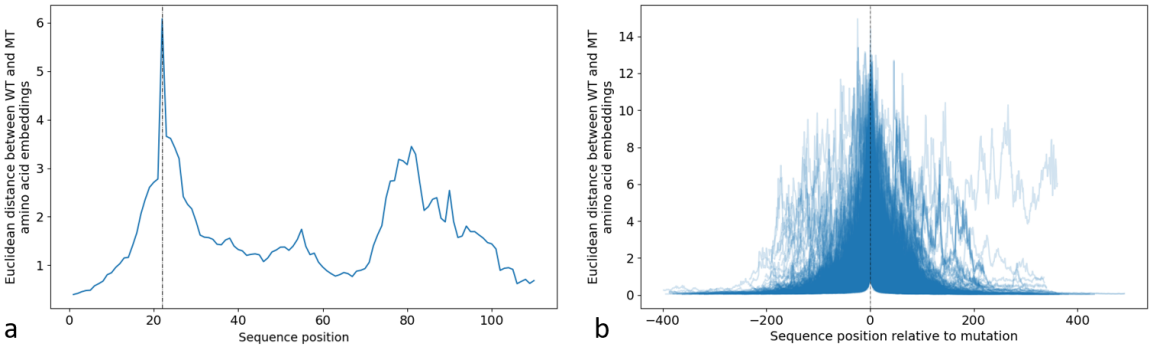

Figure 6.1: The euclidean distance between the wild-type sequence amino acid embeddings and the mutant sequence amino acid embeddings (y axis) as a function of the sequence position of the mutation (x axis). (a) The mutation W22F in protein 1AJ3 shows a strong effect in amino acids close to this location, but also an effect in another area of the sequence. (b) By plotting all mutation records of the data set S3706, this effect becomes more disperse but still relevant.

6.2.2 Mutations to binding sites cause a stronger effect in the embeddings

In an attempt to evaluate this effect in an unbiased form, a mutagenesis data set was generated (Section 6.1.2), in which 84 proteins with binding sites information were individually mutated in each amino acid. This experiment was performed under the hypothesis that this mutagenesis approach does not induce bias towards different substitutions, making the mutation effect only dependent on its position in the sequence. As such, a mutation to a binding site is expected to have a stronger effect in the embeddings, and is also expected to affect amino acids throughout the sequence more strongly.

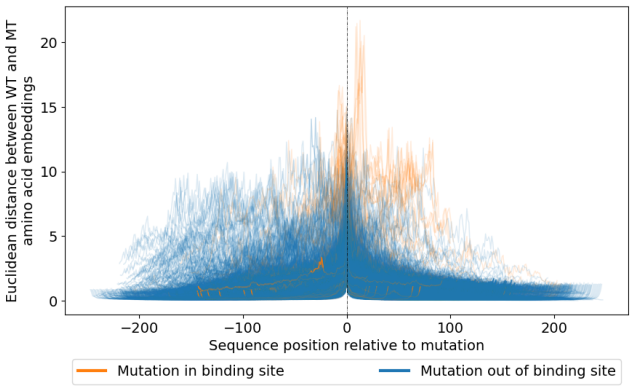

Figure 6.2: The euclidean distance between the wild-type sequence amino acid embeddings and the mutant sequence amino acid embeddings (y axis) as a function of the sequence position of the mutation (x axis) of the mutagenesis data set prepared in Section 6.1.2, centered horizontally in the location of the mutation and coloured orange if the mutation was performed in a binding site, and blue otherwise. Quite often, there is a widespread effect throughout the entire amino acid sequence.

By using the same euclidean distance procedure to describe the effect that a mutation in the sequence has in each of the amino acid embeddings, this result was plotted in terms of relative sequence

distance to the mutation, in which a widespread effect is seen throughout the amino acid embeddings (Figure 6.2). This effect is clearly stronger in amino acids closer to the mutation, and gradually weaker with an increasing distance, except for certain cases were the mutation caused a disruption throughout the sequence.

Although some of the records in the previous figure seem to indicate that a mutation in a binding site causes a higher euclidean distance between embeddings, this was investigated further by comparing this effect in distance segments in both directions (Figure 6.3). This allows the study of the statistical relevance of this hypothesis (Section 6.1.3), where a Rank-Sums (RS) test shows that mutations in binding sites indeed cause a higher euclidean distance between embeddings throughout the sequences, as evidenced by the positive RS test statistic in every segment, and the p-values very close to zero obtained in all segments.

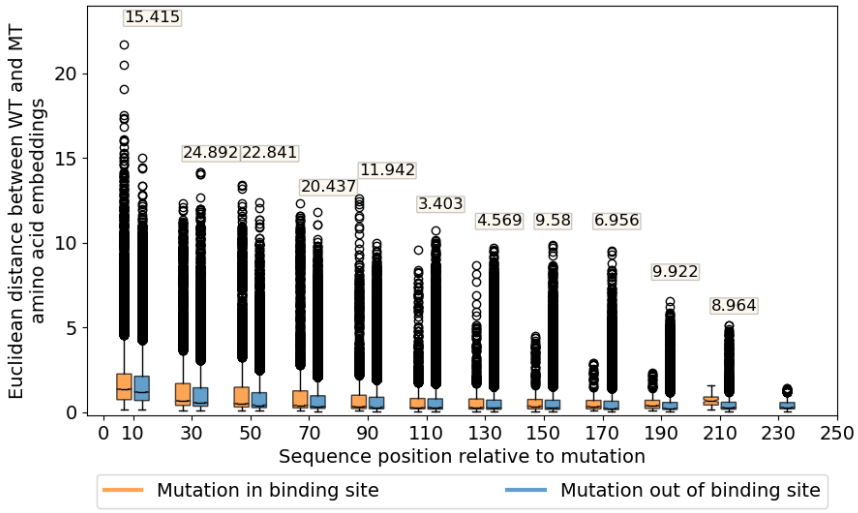

Figure 6.3: Boxplot representation of the euclidean distance value between wild-type sequence amino acid embeddings and mutant sequence amino acid embeddings (y axis) in segments of 20 amino acids of sequence distance to the mutation (x axis). For each segment, the RS test statistic is shown, showing that the effect of a mutation in a binding site (orange) is stronger than that of mutations out of the binding sites (blue).

By performing a collective RS test, over all segments, to compare the distributions of the euclidean distance between the embeddings of the binding site mutations and the non-binding site mutations, a factor of 53.237 with a p-value close to zero is obtained, further confirming the hypothesis that binding site mutations cause a significantly stronger effect in the embeddings.

6.2.3 3D distance is captured by the amino acid embeddings

Additionally, these results can also be plotted in terms of three-dimensional distance between the amino acids (Figure 6.3), were we can now see that the stronger effect of the binding site mutations is only

noticed by embeddings of amino acids close in space to the mutation. This is evidenced by the fact that the RS test statistic becomes negative for larger 3D distances, with p-values close to zero in all segments.

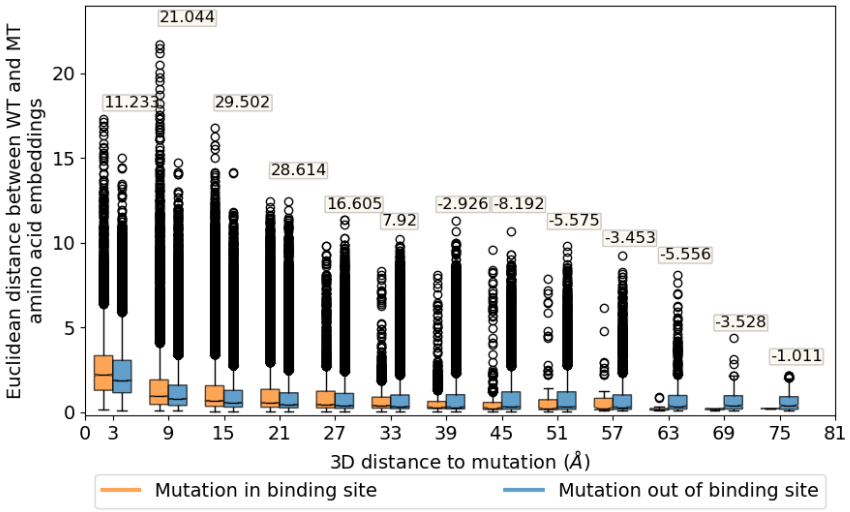

Figure 6.4: Boxplot representation of the euclidean distance value between wild-type sequence amino acid embeddings and mutant sequence amino acid embeddings (y axis) in segments of 6 angstrom of distance in space to the mutation (x axis). For each segment, the rank-sum test statistic is shown, indicating that the effect of a mutation in a binding site (orange) is stronger than that of mutations out of the binding sites (blue) only if the amino acid is close to the mutation.

Since amino acids of the protein binding site are usually close in space in the final conformation of the protein, and since it was previously concluded that a mutation to a binding site causes a stronger effect in the embeddings, seeing this effect concentrated in amino acids that are close together suggests that the SeqVec embeddings capture the protein conformation and the three-dimensional distance between amino acids.

6.3 Discussion

In this experiment we studied the effect of mutations in the SeqVec embeddings. Under the hypothesis that the simulated mutations were equivalent, and did not induce bias to specific amino acid changes, the euclidean distance between the amino acid residue embeddings of the wild-type and the mutant sequences suggests that the SeqVec embeddings can both capture long-distance effects of the mutation in the protein sequence, but also the three-dimensional conformation of the protein and the interactions between amino acids in the protein, as evidenced by the difference observed when mutations were performed to binding sites.

However, given the diversity of three-dimensional protein structures observed in nature, the assump-

tion that the simulated mutations are equivalent in amino acid type changes is far from ideal. Proteins are known to be highly divergent in sequence and highly similar in function, and sometimes the opposite is also observed, where highly similar proteins perform far from identical functions [18]. The effect of a mutation can range from completely disrupting the protein structure and incapacitating the protein from executing its function, to barely affecting the protein, and although simulating the amino acid substitutions based on the *BLOSUM* concept for biologically significant amino acid changes was the best solution for this experiment, the use of a curated deep mutational scanning data set would have been more suitable. This was, however, out of the scope of this thesis, and the experiment was performed with the limitation that performing mutations to very similar amino acids might not result in large effects in the context of the other amino acids.

Using the euclidean distance between embeddings to represent their differences could also induce some wrong conclusions, as it does not capture the notion of closeness between two high-dimensional vectors. The cosine similarity could have been used instead but even with this measure capturing distances in high-dimensional data is difficult [85]. On the other hand, the secondary structure prediction experiment applied a k-NN algorithm using the euclidean distance with success, so the chosen metric is expected to be accurate.

Although the SeqVec model was not used for protein engineering nor was it studied for the prediction of mutational effects, the *UniRep* model [14] was applied to predict the stability of naturally occurring proteins and of *de novo* designed proteins using deep mutational scanning data sets, achieving better results than well established methods such as *Rosetta* [43], and was also able to predict the functional effects of mutations to proteins, as well as modelling the fitness landscapes of diverse proteins. This model was also capable of predicting mutations that increase certain properties such as protein fluorescence, which was also observed with the *D-SPACE* model [10]. Additionally, the bidirectional transformer from [15] was used to successfully predict amino acid residue contact points as well as to predict enzyme activity changes upon mutations. Our results with the SeqVec model are in agreement with the observations that such unsupervised models can compete with state of the art models of protein biology.

7. Thermostability prediction with the Meltome Atlas wild-type data set

With the publication of the *Meltome Atlas*, an extensive and uniform data set of protein thermostability is made available [37]. Considering the obstacles faced in the prediction of thermostability directly from sequence with the *ProTherm* database, another attempt to develop such a model was performed, using SeqVec to embed the protein sequences of the *Meltome Atlas*, and studying the embeddings for their capacity to capture the melting temperature of the proteins.

7.1 Materials and Methods

7.1.1 The cross-species Meltome Atlas data set processing

The *Meltome Atlas* is the largest and most recent effort to map the thermal stability of the proteome of multiple organisms across the tree of life. Using a proteomic approach based on mass spectrometry, this data set contains the melting curves of 48691 proteins from 13 organisms ranging from *archaea* to humans and that have melting temperatures from 30 to 90 ℃ [37].

From this data, we used all proteomes except those coming from human cell lines, choosing to use only proteins that were clearly identified with a *UniProt* database entry code [9], with a total of 34501 unique protein sequences. However, some of these did not have a melting temperature annotation. These so-called non-melter proteins are annotated only by a melting curve AUC value, and although the AUC information was found to be correlated with the melting temperature information of each proteome (Figure A.6), this value depends on the temperature range of each experiment, as a non-melter at a low-temperature range could perhaps melt at a higher temperature, and as such would require a normalization or inference pre-processing step to allow the comparison of different proteomes directly. We decided not to use these proteins because only the melting temperature is independent across each experiment, allowing a direct comparison between all proteomes.

To deal with sequences with more than one record, only data referring to cell lysate experiments was used, and for all other repeated sequences, across different tissues, strains and organisms, the mean melting temperature value was used. After these processing steps, a data set with 27354 unique protein sequences with a unique melting temperature value was obtained, with a melting temperature distribution shown in Figure A.7. The protein sequences were processed by SeqVec, from which the protein-level embeddings were generated as the sequence average of the amino acid residue embeddings of each protein, obtained as the output of the middle-layer of the SeqVec model.

7.1.2 Machine learning models implementation

The cross-species data set of wild-type protein melting temperatures obtained in Section 7.1.1 was split in two random, shuffled and stratified partitions (obtained by dividing the original data set in 10 quantiles, and performing a random, shuffled split on each), where 85% of the data was used for training of ML models that use the SeqVec protein embeddings to predict the melting temperature values, and the remaining 15% was left out for an independent testing of the performance of the models on previously unseen data. As with previous experiments, PCA was also used to reduce the dimensionality of the features, to 100 principal components, fit to the training set with an explained variance of 86.4%, and used to reduce both data set partitions.

In this experiment, five different machine learning regression models were applied: the *Lasso* linear regression, the quantile linear regression, the k-NN regression model, the SVM regression model and an MLP regression model, following the same ideas as in Section 5.1.2. These models were evaluated in terms of RMSE, r^2 score, EVS, and in terms of PCC and SCC.

The *Lasso* linear regression model was applied both directly and with polynomial powers of degree 2 of the features. A few manual attempts showed that a high α value for regularization strength decreased the performance of the model, so this parameter was set to 0.001 in all experiments. The quantile regression was implemented using an iteratively reweighted least squares method to minimize the sum of absolute errors of the estimation of the mean of the melting temperature of the proteins [86], in an attempt to handle the tail with high melting temperature values in the data distribution (A.2.4) while maintaining the simplicity of a linear estimator [58].

The k-NN regression model was implemented with a k of 5, the predefined parameter value, since the performance of this model was not improved after several attempts with different values.

The SVM regressor was implemented with a linear kernel, with which a regularization strength C of 100 was chosen manually. This value was later used with a polynomial kernel of degree 2 and a RBF kernel, and was not studied further because the following MLP model outperformed even the best SVM model, which was obtained with the RBF kernel.

The MLP regression model was implemented as described in Section 5.1.2, but with a different output layer. This model possesses a single neuron in the output layer without activation function, which outputs the sum of the inputs from the previous layer. This output was optimized to predict the melting temperature of the proteins, using the MSE loss function.

Similar to the previous application for classification, a small architecture with a single hidden layer with 20 neurons was used to determine the best learning rate and number of training epochs. However, since this application has more training data available, the model could be trained for more epochs, and so a learning rate of 0.001 was established and an early stopping callback was implemented with a patience setting of 20 epochs. By isolating a random and stratified partition of 15% of the training data,

a validation set was constructed to follow the validation loss of the training of the model, with which a number of epochs of 23 was generally found to result in the least overfitting (Figure A.16).

After choice of learning rate and number of training epochs, the architecture of the MLP model was tuned with a cross-validation procedure as described in Appendix A.1.2, from which a MLP with 2 hidden layers of 256 and 20 neurons each resulted in the best mean cross-validation MSE of 48.759 (Figure A.17).

A baseline feature set was also developed. Based on the results obtained by the authors of the *Meltome Atlas*, five protein features were generated to describe the melting temperature of the records: sequence length, amino acid count by type, amino acid frequency by type, polar amino acid frequency and hydrophobic amino acid frequency. A linear regression model was fit to the training data using each of these features, from which the amino acid frequency by type vector produced the best results (based on the RMSE and PCC in the testing set, in Table A.4). During the evaluation of the previously mentioned ML models with the SeqVec embeddings, their performance with this feature set was used for comparison.

7.2 Results

7.2.1 Thermophile protein embeddings differ from mesophile embeddings

The protein embeddings generated from the protein sequences in the melting temperature data set (Section 7.1.1) were projected to two dimensions by t-SNE (preceded by PCA to 157 dimensions, with 90% explained data variance) (Figure 7.1, a), and by PCA directly (Figure 7.1, b). Although the PCA projection does not show any separation between proteins with different melting temperature ranges, the t-SNE projection reveals that proteins from thermophile organisms are different from the mesophile proteins and are grouped together in well-defined clusters, while the mesophile proteins are dispersed throughout the embedding space.

Colouring this same t-SNE projection by the organism of each protein, we find that the thermophile organisms are the only ones that are in well-defined clusters in the embedding space, while all other organisms are spread throughout the feature space (Figure A.8). This suggests that SeqVec can identify thermophile proteins from a varied data set of protein sequences from multiple organisms.

7.2.2 SeqVec features are correlated with properties related to thermostability

Protein properties such as sequence length, frequency of polar amino acids and frequency of hydrophobic amino acids were found to be correlated with higher protein stability by the data set's authors [37]. Attempting to study if any of the 1024 SeqVec features encode for these propeties, an analysis of the

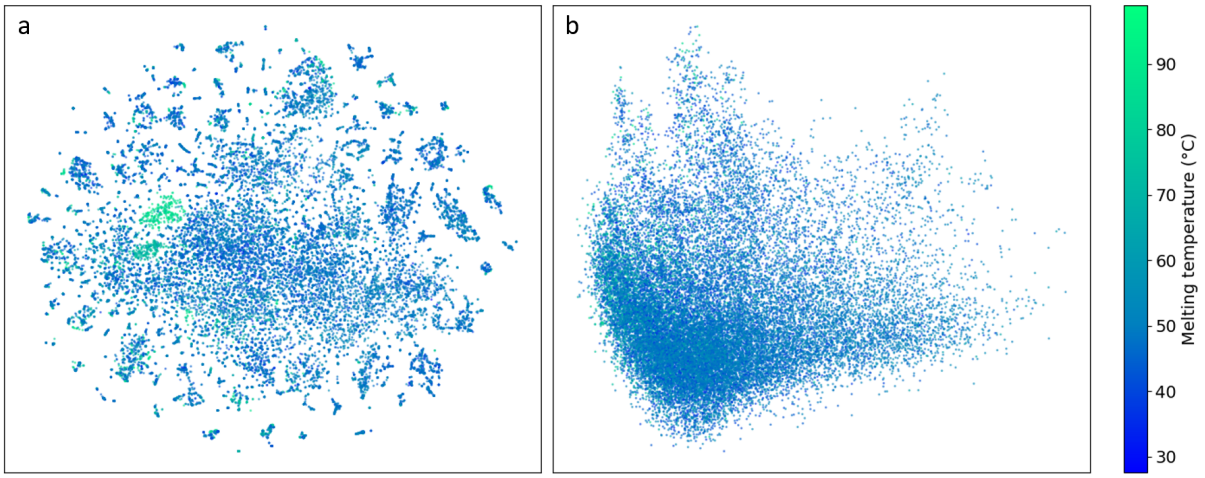

Figure 7.1: Projection to two dimensions of the protein embeddings obtained from the data set prepared in Section 7.1.1, coloured by melting temperature: (a) t-SNE (x-axis: t-SNE 1; y-axis: t-SNE 2); (b) PCA (x-axis: PCA 1; y-axis: PCA 2). Observation of the t-SNE projection shows that the embeddings of thermophile proteins are clustered together, and isolated from the remaining proteins.

protein embeddings was performed, by calculating the PCC between each of the SeqVec features and each of the three mentioned protein properties.

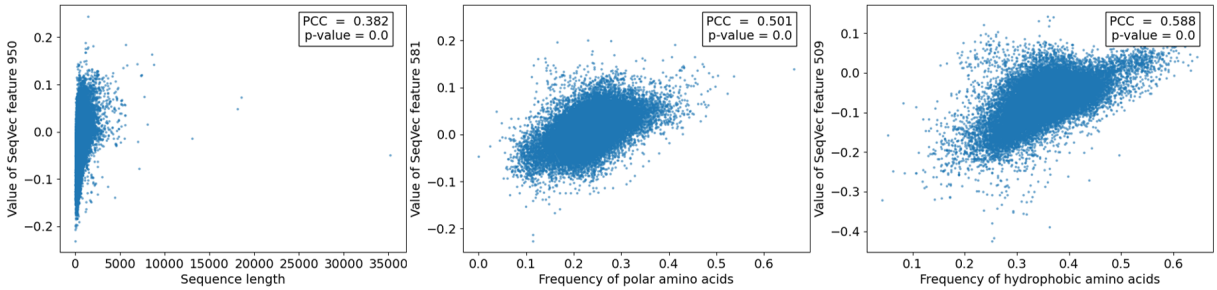

Figure 7.2: Scatter plot of three SeqVec features and three protein features known to be correlated with thermostability. The positive PCCs of some of the features from SeqVec and thermostability features indicate that the embeddings model aspects of protein thermostability. The SeqVec features were chosen as the ones with highest PCC with each of the three features: sequence length, frequency of polar amino acids and frequency of hydrophobic amino acids.

Several of the embedding features are found to be correlated with such protein properties, from which the highest correlations achieved PCC values of 0.382, 0.501 and 0.588 with sequence length, frequency of polar amino acids and frequency of hydrophobic amino acids, respectively (Figure 7.2). This result suggests that the SeqVec model learns meaningful properties of protein biology, further proposing that SeqVec embeddings can be used to model aspects of protein thermostability directly from its amino acid sequence.

7.2.3 The embeddings can be used to train a melting temperature predictor

After splitting the data set in a training and a testing partition, the protein embeddings of the training data set were used to train several machine learning regression models to predict the melting temperature of each protein sequence (Section 7.1.2). Their performance in the training and testing sets is shown in Table 7.1. The polynomial regression, as well as the RBF SVM and the MLP produced very positive results, with PCC values of over 0.70, and the most reduced test RMSE values.

Table 7.1: Performance of several machine learning regression models trained to predict the melting temperature of the protein embeddings. The models were evaluated for RMSE, r^2, EVS, PCC and SCC on the training and on the testing data, from which the PCC on the testing set was used to choose the MLP as the best performing model.

Model with *SeqVec* features	*RMSE* train	*RMSE* test	r^2 train	r^2 test	*EVS* train	*EVS* test	*PCC* train	*PCC* test	*SCC* train	*SCC* test
Lasso reg	8.35	8.14	0.33	0.36	0.33	0.36	0.58	0.60	0.31	0.33
Quantile reg	8.74	8.56	0.27	0.29	0.28	0.30	0.55	0.58	0.36	0.38
Poly 2 reg	7.28	7.24	0.49	0.49	0.49	0.49	0.70	0.70	0.43	0.43
5-NN	6.67	8.22	0.57	0.35	0.58	0.35	0.78	0.60	0.62	0.35
Linear SVR	8.75	8.56	0.27	0.29	0.28	0.30	0.55	0.58	0.36	0.38
Poly 2 SVR	10.3	10.3	-0.02	-0.02	0.00	0.00	0.13	0.13	0.27	0.27
RBF SVR	6.42	7.04	0.61	0.52	0.61	0.52	0.79	0.72	0.66	0.51
MLP	6.06	7.05	0.65	0.52	0.65	0.53	0.81	0.74	0.59	0.48

Although the MLP model shows some signs of overfitting, due to the decrease in performance between the training set and the testing set evaluation metrics, this model showed the highest PCC value of 0.74 in the testing set, the most widely used parameter to evaluate protein thermostability regression models. With a close to best test RMSE value of 7.04, this model was considered the best model obtained in this experiment and was studied further.

For this model, the r^2 and EVS scores show a positive predictive power, and their similar values suggest that the model is unbiased. The performance of this model was also studied by visualization of the predictions (Figure 7.3). From this figure we can observe that the SeqVec protein embeddings can be used to train a MLP to predict the melting temperature directly from sequence that is capable of modelling the stability of wild-type proteins, as this regression model has the capacity to predict the melting temperature of the most stable proteins in the data as significantly different from the melting temperature of the more frequent mesophile proteins, while also achieving a good PCC. This is observed in the training set as well as in the testing set.

To compare the SeqVec protein features to hand-crafted features, a baseline feature set that describes each protein by a vector with the frequency of each amino acid in the sequence was developed, and used to train and test the same MLP architecture. This baseline model produced very similar results (Table A.5), with barely any difference in performance metrics compared to using the SeqVec embeddings. With a test PCC value of 0.73 and a test SCC value of 0.43, as well as a similar test RMSE

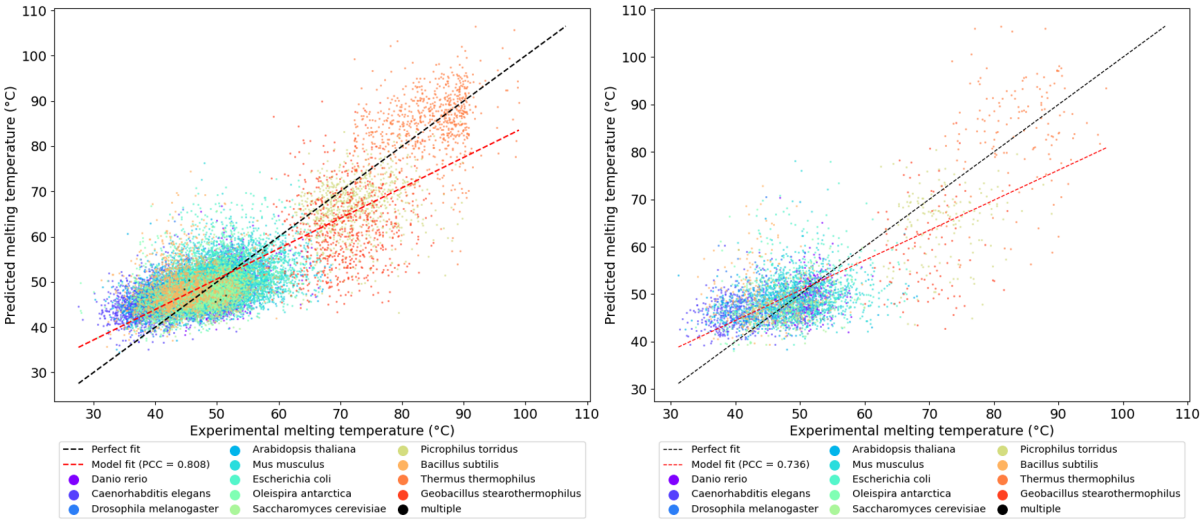

Figure 7.3: Scatter plots of the melting temperatures predicted by the MLP model developed with the protein embeddings (y axis), and their true values (x axis). The performance on the training set (left), as well as the performance on the testing set (right), show a very positive correlation and trend, indicating that the SeqVec protein embeddings can be used to develop a ML wild-type protein thermostability predictor.

of 7.14, the hand-crafted features provide very similar predictive performances, which suggest that the SeqVec features are not an improvement to protein thermostability prediction directly from sequence. An additional visualization of the predictions of the MLP model with the baseline features (Figure A.18) shows that the plot is quite similar, although slightly more dispersed.

7.2.4 The captured thermostability information is too general

To further explore the capacity of the SeqVec embeddings to capture thermostability information directly from the protein sequence, three additional experiments were performed using the previous MLP model. First, with the proteins of each individual organism, a train/test split was performed, and the model was trained on a single organism to predict the melting temperature of the remaining proteins of that same organism. Second, in a leave-one-out approach, the proteins of all except one organism were used to train the model, and the isolated organism was used to evaluate the model. Third, in a one-vs-all approach, each individual organism was used to train the model, which was then evaluated on all organisms individually. In all experiments, an early stopping callback was implemented with a patience setting of 10 (with a subset of the training data used for validation of the training) to take into account the different data set sizes in each experiment, and the performances were evaluated in terms of RMSE, r^2 score, EVS, PCC and SCC in the train and test data, from which the RMSE and the PCC were defined as the most important parameters.

The first experiment aims to see if the performance of the model is maintained on all organisms, and its results are found in Figure 7.4. There is a general decrease in performance between the training and

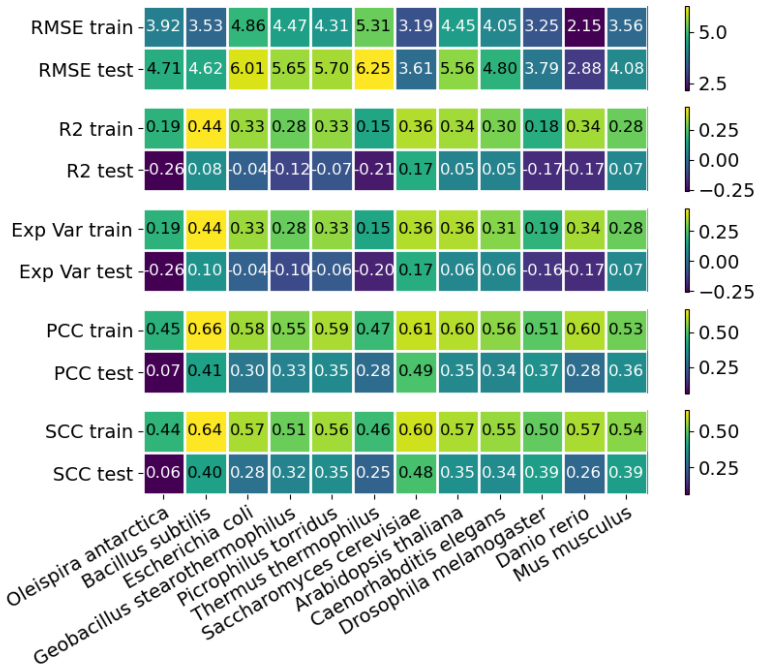

Figure 7.4: Train and test RMSE, r^2, EVS, PCC and SCC performance metrics of the MLP model with the protein embeddings for melting temperature prediction, upon training and evaluation on individual organisms. A general decrease in performance can be observed when individual organisms are used to develop the model, and a significantly worse performance was observed with the *O. antarctica* protein embeddings.

testing predictions, which was not observed in the previous experiment. This suggests that the use of a larger, more varied data set, produces a better training ground for the model and that, overall, the model is not capable of differentiating the melting temperatures found in the proteins of individual organisms. Additionally, while in most individual organisms the performance is comparable, the prediction of the melting temperature of proteins from *O. antarctica* was significantly worse than on the other organisms. In general, prediction on most organisms has a negative or very reduced r^2 score and EVS, but can maintain a RMSE close to the training data, and PCC and SCC values that show a moderate correlation between the predictions and their true values. This was not the case with proteins from *O. antarctica* because all the correlation coefficients were very close to zero, showing that although the RMSE follows the same pattern as with the other organisms, the predictions were very dispersed and uncorrelated to the experimental values.

The second experiment aims to determine the generalization capacity of a melting temperature predictor that uses the SeqVec features, by evaluating the performance of the model on a new organism, and its results are shown in Figure 7.5. This experiment showed that this prediction model is not accurate when applied to proteins from organisms it was not trained on. Ideally, the model would be capable

of maintaining the performance metrics when evaluated on a different organism, but this was not observed due to the negative r^2 score and EVS throughout all test organisms. The correlation coefficients were also both reduced in all cases, except for certain organisms such as *C. elegans*, *D. melanogaster* and *D. rerio* that show the best PCC and SCC values of this experiment, which are still lower than the correlations on the training data. Additionally, analysis of the RMSE, r^2 score and EVS values of the evaluation organisms shows a clear distinction between evaluating the model on thermophiles and mesophiles/psycrophiles, in which the model shows severely worse values when evaluated on the thermophiles.

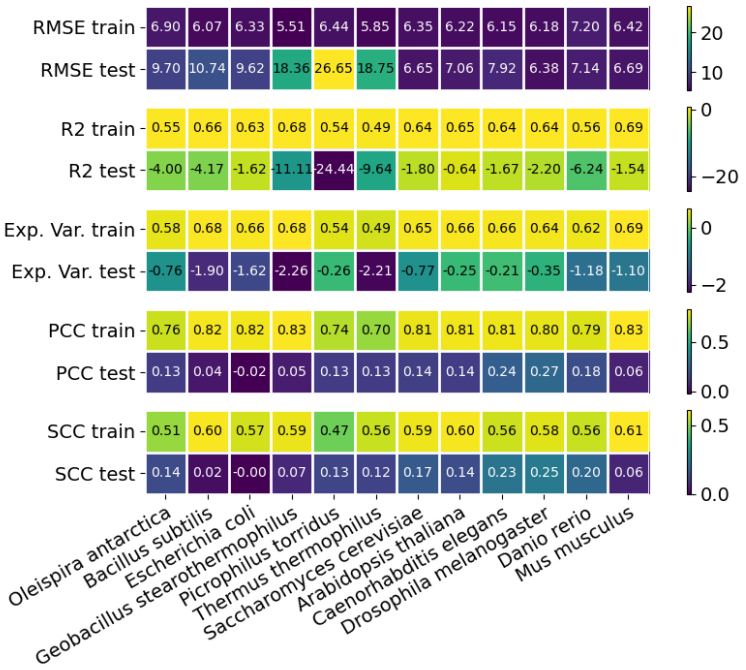

Figure 7.5: Train and test RMSE, r^2, EVS, PCC and SCC performance metrics of the MLP model with the protein embeddings for melting temperature prediction, upon training on all organisms except one, which was used for evaluation. The results show that the predictions are not accurate on organisms in which the model was not trained on.

The reduced performances in this experiment were studied further, and the prediction plots were generated, in which the reduced PCC in the left-out organisms can be clearly explained. As an example, the experiment where *M. musculus* proteins were isolated from the training and were used as the testing set are shown in Figure 7.6. Although centered in the line of perfect prediction, the predictions on the test proteins are severely dispersed, an unexpected result considering the good correlation in the training data. This suggests that the model is only learning how to identify the organisms.

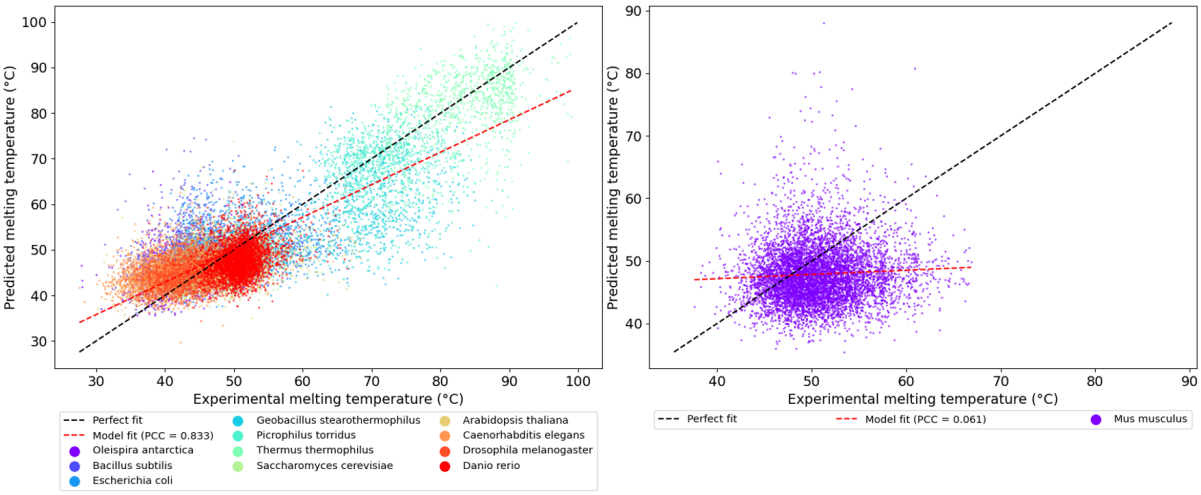

Figure 7.6: Scatter plots of the melting temperatures predicted by the MLP model developed with the protein embeddings (y axis) and their true values (x axis), on a training set without proteins from *M. musculus*, which were used as the testing set. The performance on the training set (left) suggests that the model can accurately differenciate between proteins with different melting temperatures, but its performance in the testing set (right) shows that it is not accurate enough to correctly predict the melting temperatures of proteins with similar melting temperatures belonging to an organism outside of the training set.

The third experiment aims to further evaluate the generalization capacity of this model, by training the model on a single organism and then evaluating it on the other organisms, and its results are shown in Figure 7.7. This experiment shows similar results to the previous experiment: there is a significant decrease in the PCC in the predictions on the training and testing sets, and there is a clear distinction between the RMSE of a model trained on mesophiles/psycrophiles and evaluated in thermophiles and vice versa.

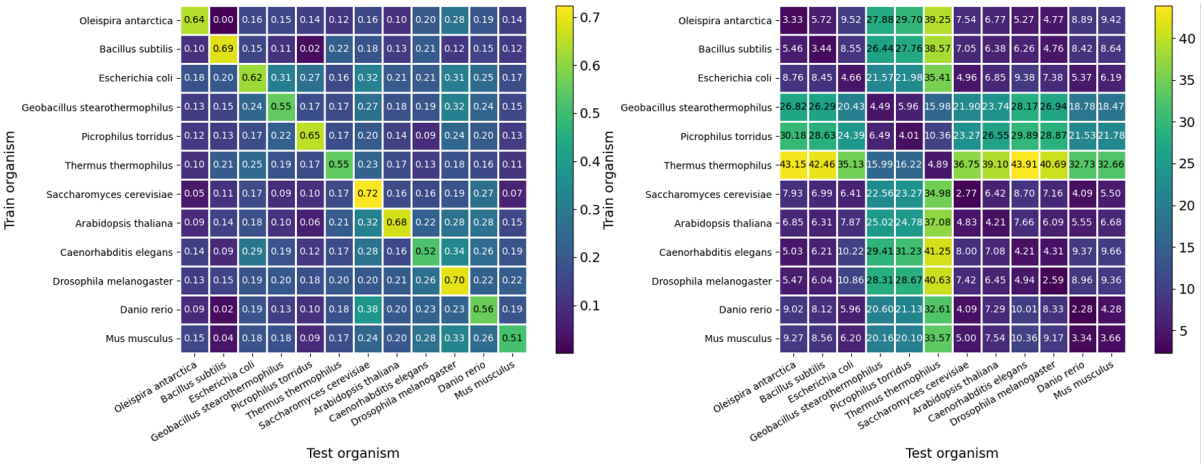

Figure 7.7: Train and test PCC and RMSE performance metrics of the MLP model with the protein embeddings for melting temperature prediction, upon training on individual organisms and testing on all the others. These results further prove that this model is not capable of adequate generalization, especially towards thermophile proteins.

The results of these experiments suggest that the SeqVec protein embeddings only superficially describe the thermostability of the proteins, because although the first regression model trained on all organisms shows very promising results, this model is incapable of generalization to different organisms. To further discuss this hypothesis, this model was compared to two baseline models.

The first baseline model uses the same protein baseline features as in Section 7.2.3, where the proteins are described by a vector with the frequency of each amino acid type in the protein sequence. The same MLP model was subjected to the same generalization experiments, where significantly worse performance metrics were observed (Appendix A.3.3). The second baseline is a naive linear regression that simply uses the average melting temperature of the organism of the protein as a single feature. When the entire data set is used for a train/test split, in the same procedure as in Section 7.2.3, this naive baseline shows a test RMSE of 4.50 and a test PCC of 0.895 (Table A.6), which are better metrics than those of either the baseline features model or the SeqVec features model. This naive model also shows lower RMSE values in the leave-one-out experiment (Figure A.22).

Although it was previously concluded that the SeqVec embeddings did not improve upon hand-crafted features, the first baseline model provided worse generalization capabilites. However, even a simple linear regression of the average melting temperature of the available proteomes provided better error and correlation metrics than all of the development models.

7.3 Discussion

The publication of the *Meltome Atlas*, with a large amount of coherent and well-annotated records from diverse organisms, provided the requirements that the *ProTherm* database previously failed to meet. The SeqVec protein embeddings generated from this data set proved to be capable of modelling aspects of protein thermostability directly from sequence, but their use in the development of ML resulted in poor performances, only capable of differentiating between different organisms. This information is too general and can not be applied in a protein engineering approach.

Although protein thermostability modelling is not usually performed on wild-type proteins, the obtained SeqVec protein features could be used to accurately differentiate proteins originating from thermophile organisms from other organisms less resistant to high temperatures, suggesting that this ELMo adaptation can identify the characteristics that make a protein resistant to high temperatures. Although it can be argued that thermophile organisms have different protein sequences and that this was the only difference captured by SeqVec, the positive correlation between some of the SeqVec features and other thermostability-related properties can be used to further confirm the previous conclusion.

The ML models trained with the protein embeddings to predict the melting temperature of previously unseen proteins produced generally positive results when using the entire data set. However, the best performing MLP predictor with the SeqVec features was only slightly better than an identical model using

a simple baseline feature set that describes each protein by an amino acid frequency vector, and this model was also found to lack in generalizability. Since this experiment used a large enough data set, the reduced performance of the models can only be explained by the features used. It is noteworthy, however, that this experiment was based on the PCA-transformed protein embeddings, with only 100 dimensions. This was the only experiment performed in this thesis where enough data was available to discard this step. Additionally, since the algorithm chosen was the MLP, the feature reduction could have been performed by an initial layer of the perceptron. This was not studied, and could be somewhat responsible for the reduced performances obtained. Performing a filtering step with CD-HIT to remove sequences of high similarity from the data set could also have produced better results, but was not attempted.

The issues with this model were not unexpected due to the difficulty of the task at hand, but with the surprisingly similar success of the baseline features we are forced to conclude that the SeqVec features do not provide a revolutionary approach for the development of new protein thermostability engineering methods that can predict the melting temperature directly from protein sequence. This is further evidenced by the high dispersion of the predictions around the average melting temperature of each organism, where simply predicting this value resulted in lower prediction errors. However, this high dispersion was not studied in detail.

By developing a model that uses the SeqVec embeddings together with the hand-crafted features that were here only used as baselines, more meaningful conclusions could have been drawn, perhaps showing that with additional features, a useful ML prediction model could be developed with the SeqVec embeddings. But at the end of this experiment, with a model that predicts a general melting temperature value, randomly spread around the organism's average melting temperature, the delicate task that is protein design becomes impossible, and a more specific model that can accurately differentiate between very similar proteins, and not just isolate the thermophile proteins, is still to be developed.

8. Conclusions and future work

The objective of this thesis was to study the application of deep language models for protein sequence modelling. For this, the SeqVec model was used to produce the protein features that were then used to implement several ML models for the prediction of protein thermostability properties.

The first experiment with the SeqVec model produced successful protein secondary structure models, and in all experiments the protein embeddings showed a capacity to model aspects of protein thermostability. The positive literature results on the application of deep language models for protein engineering, coupled with the advantages that these models have over conventional protein engineering methods, prove that this is an approach to biological sequence modelling with a lot of potential. It is, however, still behind state of the art performance, and would undoubtedly benefit from the preparation of larger and more well-curated databases of protein properties.

The results obtained in the development of ML models for protein thermostability prediction directly from wild-type sequences using the SeqVec protein embeddings revealed that such models are capable of identifying features related with thermostability, but are not yet adequate for the prediction of melting temperatures. The use of the *ProTherm* database confirmed its frequently mentioned issues, but not even with the larger *Meltome Atlas* could this prediction achieve performances useful for protein engineering. This is a difficult prediction task, with various factors influencing protein thermostability, that is not expected to be improved by introducing the use of transfer-learning with language models trained on protein sequences.

More frequently used in protein thermostability engineering, is the prediction of the effect that a mutation will have on the protein's free Gibbs energy of unfolding. Using the SeqVec features, we developed ML models that achieved higher Matthews correlation coefficients than some well established models. Our models also achieved useful precision metrics, proving that transfer-learning methods can compete with current literature. Our approach to training and evaluation of the models was built on top of a detailed processing of the data, which included several steps to guarantee that there was no data leakage, a problem that is frequently observed in the development of some of those models, and used a sequence-based approach that bypasses the hand-crafting of features for this task. The application of these models to guide mutagenesis studies is expected to see an increase in use and a substantial increase in performance if more data is made available for their training.

However, an accurate comparison of the developed models with state of the art literature in protein engineering can only be performed if the models are trained and tested on the same data sets. In addition to a comparison between the models, an application of the predictors developed in this thesis to case-studies should be performed, to evaluate their performance in different protein families and dismiss frequent issues in ML such as overfitting and bias that are usually observed in protein thermostability predictors. Our models were only broadly evaluated in the testing sets, with no detailed observation of

their behaviour on proteins for which very detailed information exists, and this could be valuable to guide further development of the use of transfer-learning for protein engineering.

The mutagenesis experiment developed in this thesis was also very superficial, and more meaningful conclusions are expected to be drawn if such an experiment would be performed with deep mutagenesis studies data sets of protein mutations. The use of these data sets, coupled with the positive results we observed when SeqVec embeddings were used for single mutations, suggest that this model can be used successfully for other protein engineering tasks for which extensive data is available, but that is still unfortunately not available for protein thermostability. This experiment could also have been taken a step further and simulated particular mutations that are frequently performed to increase the thermostability of proteins, as is the case of mutations that introduce disulfide bonds. A study of this specific effect in the embeddings could have lead to more meaningful conclusions on the applicability of the SeqVec embeddings for other protein engineering approaches, as well as its use to predict protein tertiary structures for other applications.

The use of transfer-learning also allows the use of the learned protein representations for different purposes. One of which is the protein fitness, of high interest for protein engineering. By coupling a thermostability predictor and a function predictor, the expensive and time-consuming process of assessing the suggested protein designs could be reduced to only assessing mutations that are expected to increase thermostability while also maintaining protein function.

Finally, in this thesis only the SeqVec model was studied for its potential in the development of models of protein thermostability. SeqVec is based on the ELMo language model, which is already falling behind transformer-based models such as BERT in language tasks. Already the *UniRep* pre-trained LSTM-based model is available for public use, as well an adaptation of BERT for protein sequences, and if the application of these models for biological data follows their trend in human language processing, the use of transformer-based networks for these tasks will see an improvement in performance. Additional procedures that were discussed but not implemented were the fine-tuning of the SeqVec model for specific protein families or organisms, or an additional fine-tuning step of the entire model for thermostability prediction (instead of using separate ML algorithms), which is how the previously mentioned papers perform the modelling tasks. The use of more complex ML models to handle the high-dimensionality of the representations learned by these language models is also worthy of mention, where the use of convolutional or recurrent neural networks to find better representations of a protein from their sequence of amino acid embeddings could produce more meaningful features than a simple average.

Bibliography

[1] C. B. Anfinsen, "Principles that govern the folding of protein chains," *Science*, vol. 181, no. 4096, pp. 223–230, 1973.

[2] S. Mazurenko, Z. Prokop, and J. Damborsky, "Machine learning in enzyme engineering," *ACS Catalysis*, vol. 10, no. 2, pp. 1210–1223, 2019.

[3] K. K. Yang, Z. Wu, and F. H. Arnold, "Machine-learning-guided directed evolution for protein engineering," *Nature methods*, vol. 16, no. 8, pp. 687–694, 2019.

[4] M. Musil, H. Konegger, J. Hon, D. Bednar, and J. Damborsky, "Computational design of stable and soluble biocatalysts," *ACS Catalysis*, vol. 9, no. 2, pp. 1033–1054, 2018.

[5] G. Qu, A. Li, C. G. Acevedo-Rocha, Z. Sun, and M. T. Reetz, "The crucial role of methodology development in directed evolution of selective enzymes," *Angewandte Chemie International Edition*, 2019.

[6] P. Larranaga, B. Calvo, R. Santana, C. Bielza, J. Galdiano, I. Inza, J. A. Lozano, R. Armañanzas, G. Santafé, A. Pérez *et al.*, "Machine learning in bioinformatics," *Briefings in bioinformatics*, vol. 7, no. 1, pp. 86–112, 2006.

[7] H. P. Modarres, M. Mofrad, and A. Sanati-Nezhad, "Protein thermostability engineering," *RSC advances*, vol. 6, no. 116, pp. 115 252–115 270, 2016.

[8] C.-W. Chen, M.-H. Lin, C.-C. Liao, H.-P. Chang, and Y.-W. Chu, "istable 2.0: Predicting protein thermal stability changes by integrating various characteristic modules," *Computational and structural biotechnology journal*, 2020.

[9] U. Consortium, "Uniprot: a worldwide hub of protein knowledge," *Nucleic acids research*, vol. 47, no. D1, pp. D506–D515, 2019.

[10] A. S. Schwartz, G. J. Hannum, Z. R. Dwiel, M. E. Smoot, A. R. Grant, J. M. Knight, S. A. Becker, J. R. Eads, M. C. LaFave, H. Eavani *et al.*, "Deep semantic protein representation for annotation, discovery, and engineering," *BioRxiv*, p. 365965, 2018.

[11] M. Heinzinger, A. Elnaggar, Y. Wang, C. Dallago, D. Nechaev, F. Matthes, and B. Rost, "Modeling aspects of the language of life through transfer-learning protein sequences," *BMC bioinformatics*, vol. 20, no. 1, p. 723, 2019.

[12] M. Remmert, A. Biegert, A. Hauser, and J. Söding, "Hhblits: lightning-fast iterative protein sequence searching by hmm-hmm alignment," *Nature methods*, vol. 9, no. 2, pp. 173–175, 2012.

[13] M. Steinegger and J. Söding, "Mmseqs2 enables sensitive protein sequence searching for the analysis of massive data sets," *Nature biotechnology*, vol. 35, no. 11, pp. 1026–1028, 2017.

[14] E. C. Alley, G. Khimulya, S. Biswas, M. AlQuraishi, and G. M. Church, "Unified rational protein engineering with sequence-based deep representation learning," *Nature methods*, vol. 16, no. 12, pp. 1315–1322, 2019.

[15] A. Rives, S. Goyal, J. Meier, D. Guo, M. Ott, C. L. Zitnick, J. Ma, and R. Fergus, "Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences," *bioRxiv*, 2019. [Online]. Available: https://www.biorxiv.org/content/early/2019/04/29/622803

[16] D. B. Duong, L. Gai, A. Uppunda, D. Le, E. Eskin, J. J. Li, and K.-W. Chang, "Annotating gene ontology terms for protein sequences with the transformer model," *bioRxiv*, 2020.

[17] R. Rao, N. Bhattacharya, N. Thomas, Y. Duan, P. Chen, J. Canny, P. Abbeel, and Y. Song, "Evaluating protein transfer learning with tape," in *Advances in Neural Information Processing Systems*, 2019, pp. 9689–9701.

[18] D. L. Nelson, A. L. Lehninger, and M. M. Cox, *Lehninger principles of biochemistry*. Macmillan, 2008.

[19] J. G. V. Donald Voet, *Biochemistry (Fourth Edition)*, 4th ed. Wiley, 2010.

[20] T. E. Creighton, *Biophysical Chemistry of Nucleic Acids and Proteins*. Helvetian Press, 2010.

[21] R. Harada, N. Tochio, T. Kigawa, Y. Sugita, and M. Feig, "Reduced native state stability in crowded cellular environment due to protein–protein interactions," *Journal of the American Chemical Society*, vol. 135, no. 9, pp. 3696–3701, 2013.

[22] J. Stourac, J. Dubrava, M. Musil, J. Horackova, J. Damborsky, S. Mazurenko, and D. Bednar, "Fireprotdb: database of manually curated protein stability data," *Nucleic Acids Research*, 2020.

[23] G. Bruylants, J. Wouters, and C. Michaux, "Differential scanning calorimetry in life science: thermodynamics, stability, molecular recognition and application in drug design," *Current medicinal chemistry*, vol. 12, no. 17, pp. 2011–2020, 2005.

[24] N. J. Greenfield, "Using circular dichroism collected as a function of temperature to determine the thermodynamics of protein unfolding and binding interactions," *Nature protocols*, vol. 1, no. 6, p. 2527, 2006.

[25] M. S. Kumar, K. A. Bava, M. M. Gromiha, P. Prabakaran, K. Kitajima, H. Uedaira, and A. Sarai, "Protherm and pronit: thermodynamic databases for proteins and protein–nucleic acid interactions," *Nucleic acids research*, vol. 34, no. suppl_1, pp. D204–D206, 2006.

[26] D. H. Corrêa and C. H. Ramos, "The use of circular dichroism spectroscopy to study protein folding, form and function," *African Journal of Biochemistry Research*, vol. 3, no. 5, pp. 164–173, 2009.

[27] S. N. Baker, T. M. McCleskey, S. Pandey, and G. A. Baker, "Fluorescence studies of protein thermostability in ionic liquids," *Chemical communications*, no. 8, pp. 940–941, 2004.

[28] M. Schneider, S. Walta, C. Cadek, W. Richtering, and D. Willbold, "Fluorescence correlation spectroscopy reveals a cooperative unfolding of monomeric amyloid-β 42 with a low gibbs free energy," *Scientific reports*, vol. 7, no. 1, pp. 1–8, 2017.

[29] Y. Yang, S. Urolagin, A. Niroula, X. Ding, B. Shen, and M. Vihinen, "Pon-tstab: protein variant stability predictor. importance of training data quality," *International journal of molecular sciences*, vol. 19, no. 4, p. 1009, 2018.

[30] Y. Dehouck, A. Grosfils, B. Folch, D. Gilis, P. Bogaerts, and M. Rooman, "Fast and accurate predictions of protein stability changes upon mutations using statistical potentials and neural networks: Popmusic-2.0," *Bioinformatics*, vol. 25, no. 19, pp. 2537–2543, 2009.

[31] Y. Chen, H. Lu, N. Zhang, Z. Zhu, S. Wang, and M. Li, "Premps: Predicting the effects of single mutations on protein stability," *bioRxiv*, 2020.

[32] E. Capriotti, P. Fariselli, and R. Casadio, "I-mutant2. 0: predicting stability changes upon mutation from the protein sequence or structure," *Nucleic acids research*, vol. 33, no. suppl_2, pp. W306–W310, 2005.

[33] C. Y. Wang, P. M. Chang, M. L. Ary, B. D. Allen, R. A. Chica, S. L. Mayo, and B. D. Olafson, "Protabank: A repository for protein design and engineering data," *Protein Science*, vol. 27, no. 6, pp. 1113–1124, 2018.

[34] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, "The Protein Data Bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235–242, 01 2000. [Online]. Available: https://doi.org/10.1093/nar/28.1.235

[35] H. Pezeshgi Modarres, M. R. Mofrad, and A. Sanati-Nezhad, "Protdatatherm: A database for thermostability analysis and engineering of proteins," *PloS one*, vol. 13, no. 1, p. e0191222, 2018.

[36] H. M. Hussain, H. Seker, and M. Gorania, "Bioinformatics approach to classification of four classes of organism in relation to their optimal growth temperature," *International Journal of Pharma Medicine and Biological Sciences*, 2018.

[37] A. Jarzab, N. Kurzawa, T. Hopf, M. Moerch, J. Zecha, N. Leijten, Y. Bian, E. Musiol, M. Maschberger, G. Stoehr *et al.*, "Meltome atlas—thermal proteome stability across the tree of life," *Nature methods*, vol. 17, no. 5, pp. 495–503, 2020.

[38] M. Lehmann, L. Pasamontes, S. Lassen, and M. Wyss, "The consensus concept for thermostability engineering of proteins," *Biochimica et Biophysica Acta (BBA)-protein structure and molecular enzymology*, vol. 1543, no. 2, pp. 408–415, 2000.

[39] V. Potapov, M. Cohen, and G. Schreiber, "Assessing computational methods for predicting protein stability upon mutation: good on average but not in the details," *Protein engineering, design & selection*, vol. 22, no. 9, pp. 553–560, 2009.

[40] N. Pokala and T. M. Handel, "Energy functions for protein design: adjustment with protein–protein complex affinities, models for the unfolded state, and negative design of solubility and specificity," *Journal of molecular biology*, vol. 347, no. 1, pp. 203–227, 2005.

[41] A. Benedix, C. M. Becker, B. L. de Groot, A. Caflisch, and R. A. Böckmann, "Predicting free energy changes using structural ensembles," *Nature methods*, vol. 6, no. 1, pp. 3–4, 2009.

[42] J. Schymkowitz, J. Borg, F. Stricher, R. Nys, F. Rousseau, and L. Serrano, "The foldx web server: an online force field," *Nucleic acids research*, vol. 33, no. suppl_2, pp. W382–W388, 2005.

[43] C. A. Rohl, C. E. Strauss, K. M. Misura, and D. Baker, "Protein structure prediction using rosetta," in *Methods in enzymology*. Elsevier, 2004, vol. 383, pp. 66–93.

[44] J. Cheng, A. Randall, and P. Baldi, "Prediction of protein stability changes for single-site mutations using support vector machines," *Proteins: Structure, Function, and Bioinformatics*, vol. 62, no. 4, pp. 1125–1132, 2006.

[45] L.-T. Huang, M. M. Gromiha, and S.-Y. Ho, "iptree-stab: interpretable decision tree based method for predicting protein stability changes upon mutations," *Bioinformatics*, vol. 23, no. 10, pp. 1292–1293, 2007.

[46] D. E. Pires, D. B. Ascher, and T. L. Blundell, "Duet: a server for predicting effects of mutations on protein stability using an integrated computational approach," *Nucleic acids research*, vol. 42, no. W1, pp. W314–W319, 2014.

[47] X. Fang, J. Huang, R. Zhang, F. Wang, Q. Zhang, G. Li, J. Yan, H. Zhang, Y. Yan, and L. Xu, "Convolution neural network-based prediction of protein thermostability," *Journal of Chemical Information and Modeling*, vol. 59, no. 11, pp. 4833–4843, 2019.

[48] J. Fang, "A critical review of five machine learning-based algorithms for predicting protein stability changes upon mutation," *Briefings in bioinformatics*, vol. 21, no. 4, pp. 1285–1292, 2020.

[49] F. Pucci, K. V. Bernaerts, J. M. Kwasigroch, and M. Rooman, "Quantification of biases in predictions of protein stability changes upon mutations," *Bioinformatics*, vol. 34, no. 21, pp. 3659–3665, 2018.

[50] S.-H. Liao, P.-H. Chu, and P.-Y. Hsiao, “Data mining techniques and applications–a decade review from 2000 to 2011,” *Expert systems with applications*, vol. 39, no. 12, pp. 11 303–11 311, 2012.

[51] J. P. Jiawei Han, Micheline Kamber, *Data Mining: Concepts and Techniques*, 3rd ed. Elsevier, 2012.

[52] E. Alpaydin, *Introduction to Machine Learning*, 3rd ed. The MIT Press, 2014.

[53] S. Wold, K. Esbensen, and P. Geladi, “Principal component analysis,” *Chemometrics and intelligent laboratory systems*, vol. 2, no. 1-3, pp. 37–52, 1987.

[54] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[55] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint arXiv:1802.03426*, 2018.

[56] J. F. Trevor Hastie, Robert Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference and Prediction*, 2nd ed. Springer, 2009.

[57] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, “Scikit-learn: Machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[58] R. Koenker and K. F. Hallock, “Quantile regression,” *Journal of economic perspectives*, vol. 15, no. 4, pp. 143–156, 2001.

[59] P. Domingos, *The Master Algorithm: How the Quest for the Ultimate Learning Machine Will Remake Our World*. USA: Basic Books, Inc., 2018.

[60] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[61] B. Xu, N. Wang, T. Chen, and M. Li, “Empirical evaluation of rectified activations in convolutional network,” *arXiv preprint arXiv:1505.00853*, 2015.

[62] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-scale machine

learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: https://www.tensorflow.org/

[63] W. M. J. Mohammed J. Zaki, *Data Mining and Analysis: Fundamental Concepts and Algorithms*. Cambridge University Press, 2014.

[64] L. A. Jeni, J. F. Cohn, and F. De La Torre, "Facing imbalanced data–recommendations for the use of performance metrics," in *2013 Humaine association conference on affective computing and intelligent interaction*. IEEE, 2013, pp. 245–251.

[65] J. Davis and M. Goadrich, "The relationship between precision-recall and roc curves," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 233–240.

[66] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, and SciPy 1.0 Contributors, "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python," *Nature Methods*, vol. 17, pp. 261–272, 2020.

[67] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[68] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[69] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[70] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[71] M. E. Peters, M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer, "Deep contextualized word representations," in *Proc. of NAACL*, 2018.

[72] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[73] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.

[74] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.

[75] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," *arXiv preprint arXiv:1412.3555*, 2014.

[76] M. Spellings, "Agglomerative attention," *arXiv preprint arXiv:1907.06607*, 2019.

[77] N. Kitaev, Ł. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," *arXiv preprint arXiv:2001.04451*, 2020.

[78] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," 2020.

[79] A. Villegas-Morcillo, S. Makrodimitris, R. van Ham, A. M. Gomez, V. Sanchez, and M. Reinders, "Unsupervised protein embeddings outperform hand-crafted sequence and structure features at predicting molecular function," *bioRxiv*, 2020.

[80] Y. Huang, B. Niu, Y. Gao, L. Fu, and W. Li, "Cd-hit suite: a web server for clustering and comparing biological sequences," *Bioinformatics*, vol. 26, no. 5, pp. 680–682, 2010.

[81] J. M. Dana, A. Gutmanas, N. Tyagi, G. Qi, C. O'Donovan, M. Martin, and S. Velankar, "SIFTS: updated Structure Integration with Function, Taxonomy and Sequences resource allows 40-fold increase in coverage of structure-based annotations for proteins," *Nucleic Acids Research*, vol. 47, no. D1, pp. D482–D489, 11 2018. [Online]. Available: https://doi.org/10.1093/nar/gky1114

[82] S. McGinnis and T. L. Madden, "Blast: at the core of a powerful and diverse set of sequence analysis tools," *Nucleic acids research*, vol. 32, no. suppl_2, pp. W20–W25, 2004.

[83] S. Henikoff and J. G. Henikoff, "Amino acid substitution matrices from protein blocks," *Proceedings of the National Academy of Sciences*, vol. 89, no. 22, pp. 10 915–10 919, 1992.

[84] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Ann. Math. Statist.*, vol. 18, no. 1, pp. 50–60, 03 1947. [Online]. Available: https://doi.org/10.1214/aoms/1177730491

[85] L. Ertöz, M. Steinbach, and V. Kumar, "Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data," in *Proceedings of the 2003 SIAM international conference on data mining*. SIAM, 2003, pp. 47–58.

[86] S. Seabold and J. Perktold, “statsmodels: Econometric and statistical modeling with python,” in *9th Python in Science Conference*, 2010.

[87] G. Van Rossum and F. L. Drake, *Python 3 Reference Manual*. Scotts Valley, CA: CreateSpace, 2009.

[88] W. McKinney *et al.*, “Data structures for statistical computing in python,” in *Proceedings of the 9th Python in Science Conference*, vol. 445. Austin, TX, 2010, pp. 51–56.

[89] T. E. Oliphant, *A guide to NumPy*. Trelgol Publishing USA, 2006, vol. 1.

[90] P. J. Cock, T. Antao, J. T. Chang, B. A. Chapman, C. J. Cox, A. Dalke, I. Friedberg, T. Hamelryck, F. Kauff, B. Wilczynski *et al.*, “Biopython: freely available python tools for computational molecular biology and bioinformatics,” *Bioinformatics*, vol. 25, no. 11, pp. 1422–1423, 2009.

[91] J. D. Hunter, “Matplotlib: A 2d graphics environment,” *Computing in Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[92] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, “Pytorch: An imperative style, high-performance deep learning library,” in *Advances in Neural Information Processing Systems 32*, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. Curran Associates, Inc., 2019, pp. 8024–8035. [Online]. Available: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[93] D. Ulyanov, “Multicore-tsne,” https://github.com/DmitryUlyanov/Multicore-TSNE, 2016.

A. Appendix

A.1 Additional implementation details

A.1.1 Software used

All computational work was done using the *Python* software [87] version 3.6.10. For data manipulation, the libraries *pandas* [88] and *NumPy* [89] were used. The library *Biopython* [90] was also used to work with the protein sequence files. All plots were generated using the *Matplotlib* [91] library. The library *SciPy* [66] was used to perform all statistical tests. SeqVec is implemented in *PyTorch* [92] version 1.2.0, and the implementation of all machine learning models was done using *Python's* machine learning library *Scikit-learn* [57], except for the MLPs that were implemented with the *Keras* module from *TensorFlow* [62] version 1.15.0 and the quantile regression that was implemented with the *Statsmodels* [86] library. All models were applied with the predefined parameters except when explicitly noted otherwise.

t-SNE was used exclusively for data visualization, and was implemented using the *Multicore t-SNE* [93] library, using the predefined parameters (namely, a perplexity of 30.0). PCA was used for data visualization and for feature reduction in the training of the ML models, and was implemented using the *Scikit-learn* [57] library. Linear Discriminant Analysis (LDA) was also tried for feature reduction but did not provide good results, and was not studied further. Other feature selection methods such as filters and wrappers were not attempted due to their long application times and to avoid a complex feature engineering step, in order to provide an unbiased exploration of the *SeqVec* embeddings.

A.1.2 Cross-validation hyper-parameter tuning procedure

Cross-validation was implemented with three objectives: to maximize the use of the available training data, to prevent overfitting to the training data set, and to avoid using the testing data set for a biased choice of models. The models with pre-defined parameters that showed the most potential were fine-tuned with this method and then evaluated in the unbiased testing set data.

In this work, the hyper-parameter tuning of each model was performed using an exhaustive grid search method. Upon training and evaluation of all models in all folds, this method chooses the best hyper-parameter combination as the one with the best average performance over all the folds. After choosing the best model hyper-parameters, this model is trained on the entire training data set, and is then evaluated in terms of its performance in the previously unused testing data set. The PCA feature reduction step was also fit and performed individually on each fold. For all implementations of this procedure in this work, the entire cross-validation process was done using the training data set, which was split in 7 random, shuffled, and stratified folds.

A.2 Additional data processing details

A.2.1 Wild-type protein thermostability data set processing

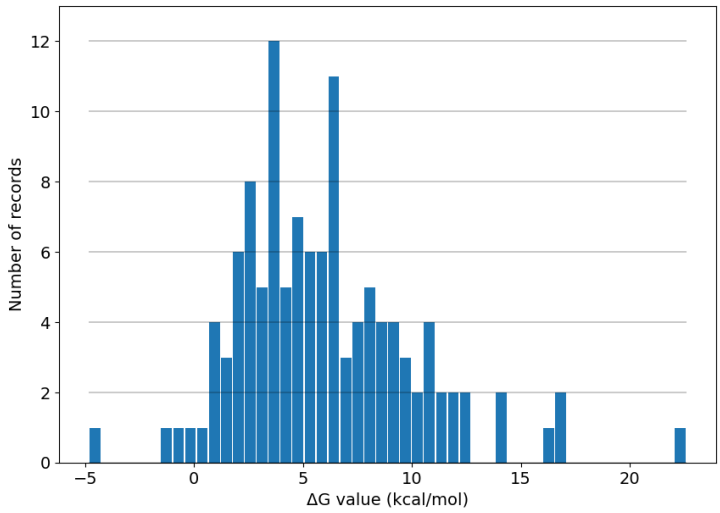

Figure A.1: Histogram of the distribution of free Gibbs energy of unfolding in the processed *ProTherm* wild-type data set obtained in Section 4.1.1.

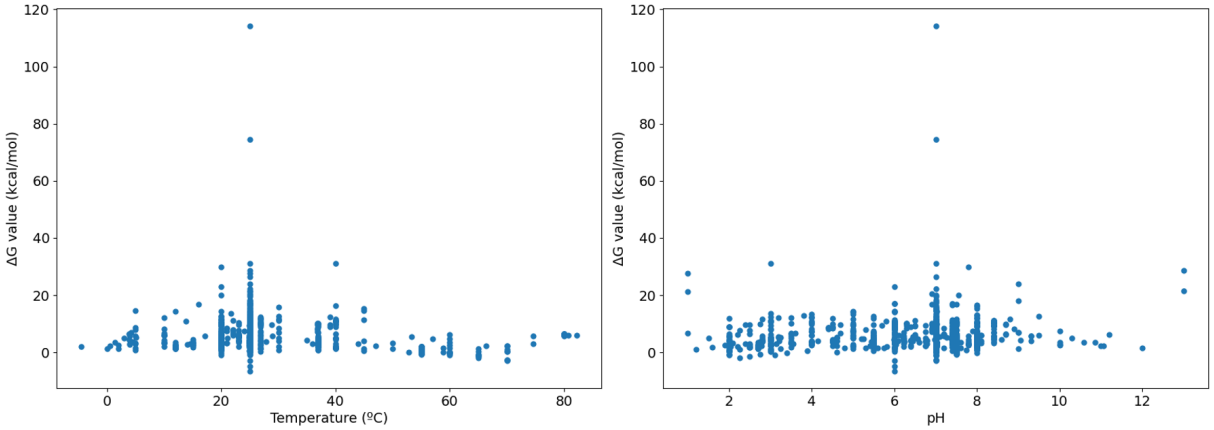

Figure A.2: Free Gibbs energy of unfolding values as a function of temperature (left) and pH (right) in the processed ProTherm wild-type data set obtained in Section 4.1.1. The ΔG values in the data set are not correlated with any of the experimental conditions.

A.2.2 Single-mutants protein thermostability data set processing

The protein thermostability prediction models *PoPMuSiC-2.0* [30] and *I-Mutant 2.0* [32], published the data sets S1948 and S2648, respectively, named after the number of records in the set. These data sets are frequently used by more recent models, and although both claim to be redundancy-free, S1948 contains multiple T and pH values for some mutations. The S2648 data set has a single $\Delta\Delta G$ value for each different mutation, calculated with Equation (A.1) as a weighted mean value where records obtained closer to 25 ºC and pH 7 were given higher weights.

$$\Delta\Delta G_M = (\sum_{i=1}^{n} w_i^{pH} w_i^T w_i^{add} \Delta\Delta G_i)/(\sum_{i=1}^{n} w_i^{pH} w_i^T w_i^{add}) \tag{A.1}$$

where w_i^{pH}, w_i^T and w_i^{add} are the weights given to each record i for a given protein, according to pH, temperature and denaturing additives, respectively, and are obtained by Equation (A.2).

$$w_i^{pH} = 1 - \frac{|pH_i - 7|}{7} \quad , \quad w_i^T = max(0; 1 - \frac{|T_i - 25|}{25}) \quad , \quad \begin{cases} w_i^{add} = \prod_{j=1}^{m} (1 - \frac{C_{ij}}{C_j^{max}}) & \text{if } m > 0 \\ w_i^{add} = 1 & \text{if } m = 0 \end{cases} \tag{A.2}$$

where n is the number records of $\Delta\Delta G$ for a given mutation, m is the number of additives in solution in experiment i, C_{ij} is the concentration of additive j in experiment i, and C_j^{max} is the maximal concentration of additive j in all experiments in the dataset.

Two mutation records were considered duplicates if they have the same PDB identifier, PDB chain, wild-type amino acid, mutation location in the PDB sequence and mutant amino acid, and overall we found 4000 unique records. To calculate a unique $\Delta\Delta G$ value per record that is independent of experimental conditions, Equation (A.1) was used ignoring the information about denaturing additives. However, for each mutation, if one of the duplicates was from S2648 or S921, this value was used directly, as these were already processed accordingly. Overall, 2579 records came directly from S2648, 911 records came directly from S921, 186 records came directly from S630 and 168 records came directly from S3568. 65 records were obtained with Equation (A.1), and the remaining 91 records were labelled as conflicting, because throughout the data sets had both positive and negative $\Delta\Delta G$ values. After removing duplicate and conflicting records, we obtained a data set with 3909 unique records.

The PDB identifier of each protein was used to obtain the protein sequences. To generate the mutant sequence, the mutation location was used to change the wild-type sequences in the data set. To take into account several exceptions with the PDB amino acid numbering scheme, the *mmCIF* files of each protein, which contain all the protein information in the form of a *PDBx* dictionary file, were used to find missing amino acids, the sequence start index and gaps to locate the amino acid to mutate as shown in

Equation (A.3). The information about the wild-type amino acid was used to double check this process, but still a few exceptions needed to be solved manually, as these are related to heterogeneous amino acids, missing residues not accounted for, incoherent sequence start indexes and extra amino acids that are not numbered.

$$Sequence\ position = PDB\ position + missing\ amino\ acids - start\ index - gaps \tag{A.3}$$

After obtaining the wild-type and mutant protein sequences, the data set revealed 203 repeated sequence pairs, which come from different PDB identifiers and chains that correspond to an equal amino acid sequence, which were also removed.

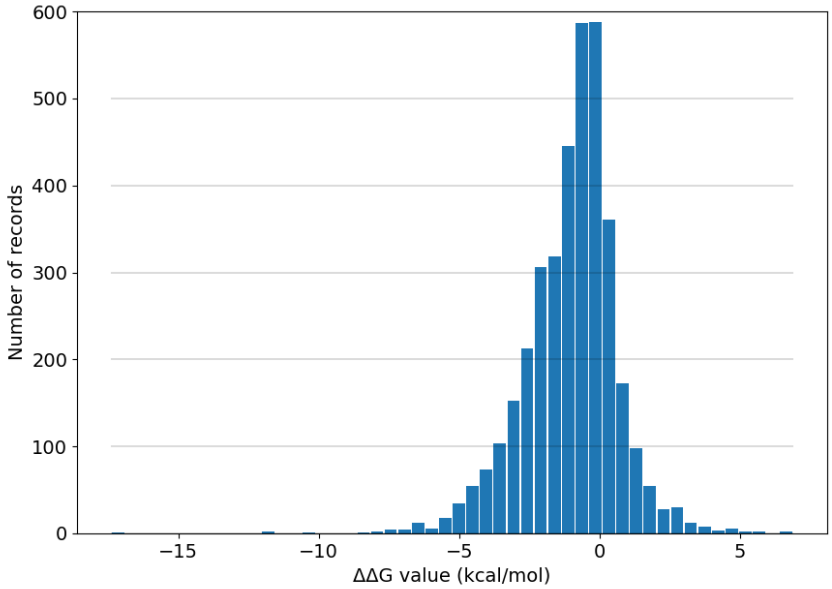

Figure A.3: Histogram of the distribution of free Gibbs energy of unfolding changes upon single mutations in the processed data set S3706 obtained in Section 5.1.1.

A.2.3 Mutagenesis data set processing

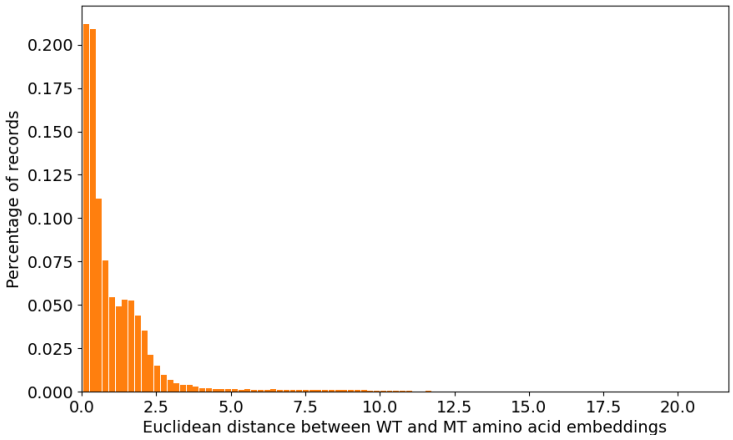

Figure A.4: Histogram of the distribution of the euclidean distances between amino acid embeddings of wild-type and mutant sequences of mutations to binding sites in the mutagenesis data set obtained in Section 6.1.2.

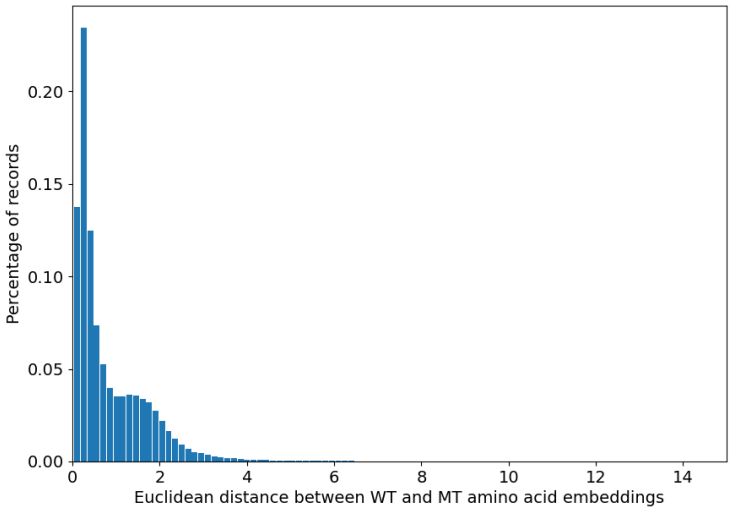

Figure A.5: Histogram of the distribution of the euclidean distances between amino acid embeddings of wild-type and mutant sequences of mutations outside binding sites in the mutagenesis data set obtained in Section 6.1.2.

A.2.4 Protein melting temperatures data set processing

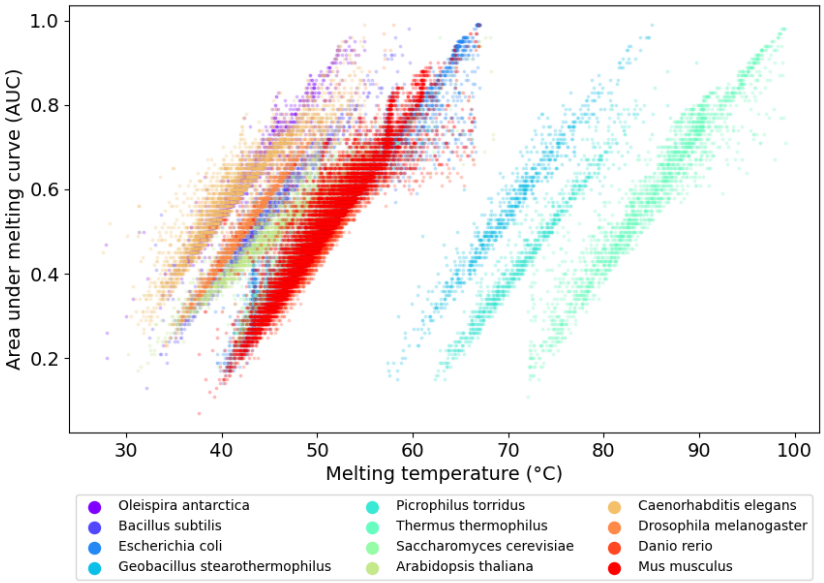

Figure A.6: Area under the melting curve as a function of the melting temperature of each protein in the *Meltome Atlas* data set before processing in Section 7.1.1. From an analysis of this plot, a correlation between the AUC and the melting temperature is observed, but this correlation is individual for each organism, as it depends on the temperature range at which the experiments were performed.

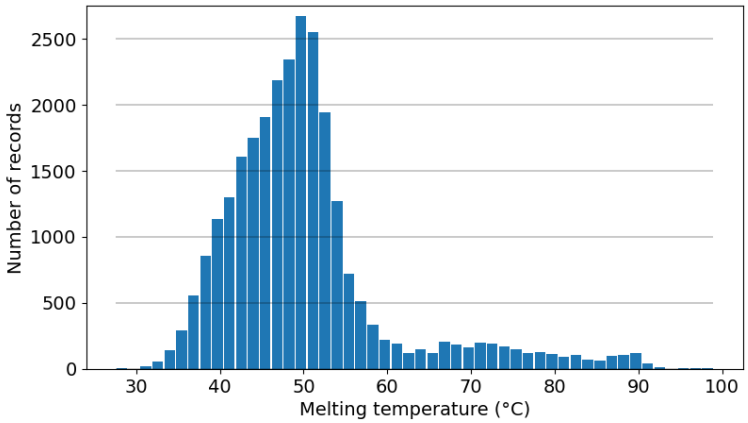

Figure A.7: Histogram of the distribution of melting temperature values in the processed *Meltome Atlas* data set obtained in Section 7.1.1.

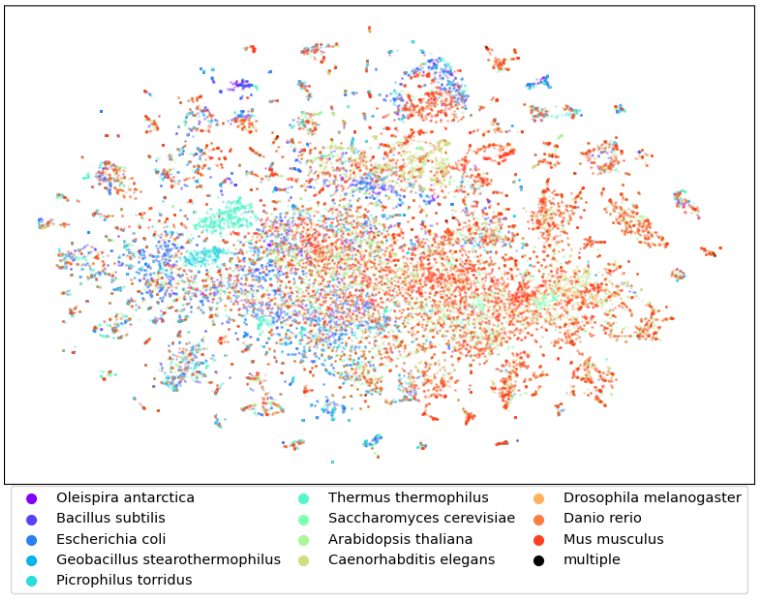

Figure A.8: t-SNE projection to two dimensions of the protein embeddings obtained from the data set prepared in Section 7.1.1, coloured by organism. Observation of this projection shows that the proteins of most organisms are dispersed through the space, except those from thermophile organisms that are mostly clustered together, indicating that the embeddings are not different from organism to organism, but different according to their thermostability.

A.3 Additional model development information

A.3.1 Secondary structure prediction

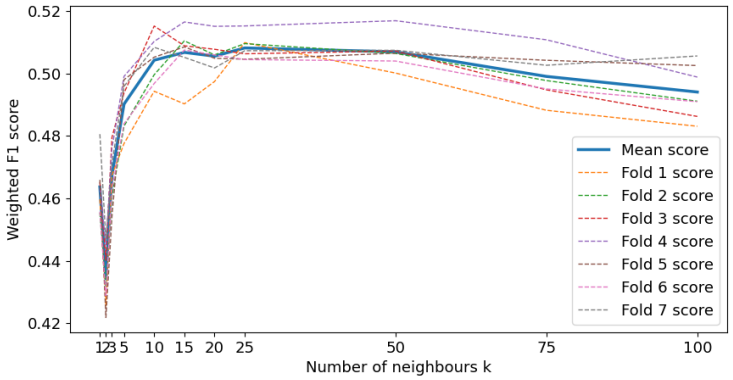

Figure A.9: Cross-validation hyper-parameter tuning process of the k-NN secondary structure predictor. The optimal value for the parameter k was determined to be 25, from the possible values 1, 2, 3, 5, 10, 15, 25, 50, 75, 100. This choice was based on the mean test set F_1 score average of all classes, weighted according to the number of records with each label, which was 0.508, the best obtained in all experiments.

A.3.2 Thermostability change upon mutation prediction

Table A.1: Performance of the basic set of classifiers on the testing set S434, trained on the data set S3272 as obtained in Section 5.1.2, using the different features generated to represent the mutation records. The classifiers were evaluated for accuracy, recall, specificity, precision, MCC, F1 and ROC AUC scores on the testing set S430. For each feature set, the predictor with the highest MCC was chosen, and shown here.

Feature set	Best model	Acc	Sens	Spec	Prec	MCC	F1 score	ROC AUC
Diff	3-NN	0.65	0.27	0.89	0.62	0.21	0.38	0.64
Diff_0	Log reg	0.69	0.26	0.97	0.85	0.34	0.39	0.76
Diff_2	Linear SVM	0.68	0.26	0.96	0.80	0.32	0.40	0.76
Diff_5	Log reg	0.66	0.15	0.99	0.93	0.28	0.25	0.78
Diff_10	Linear SVM	0.65	0.15	0.98	0.86	0.26	0.25	0.79
Diff_20	Linear SVM	0.65	0.12	1.	0.95	0.27	0.22	0.79
Concat	50-NN	0.61	0.02	1.	1.	0.10	0.03	0.29
ConcatDiff	9-NN	0.60	0.06	0.95	0.43	0.02	0.10	0.45
MeanSTD	15-NN	0.62	0.08	0.98	0.68	0.13	0.14	0.63
Moments	Linear SVM	0.66	0.20	0.96	0.78	0.27	0.32	0.73

Table A.2: Performance of the basic set of classifiers on the testing set S434, trained on the data set S3272 without insignificant mutation records, using the different features generated to represent the mutation records. The classifiers were evaluated for accuracy, recall, specificity, precision, MCC, F1 and ROC AUC scores on the testing set S430. For each feature set, the predictor with the highest MCC was chosen, and shown here. A general increase in precision is observed with all features, but at a cost of reduced MCC performances.

Feature set	Best model	Acc	Sens	Spec	Prec	MCC	F1 score	ROC AUC
Diff	5-NN	0.63	0.06	0.99	0.85	0.16	0.12	0.61
Diff_0	Log reg	0.66	0.15	0.99	0.93	0.28	0.25	0.76
Diff_2	Linear SVM	0.64	0.11	0.98	0.79	0.20	0.19	0.76
Diff_5	Linear SVM	0.65	0.11	1.	1.	0.27	0.2	0.77
Diff_10	Linear SVM	0.62	0.05	1.	1.	0.17	0.09	0.77
Diff_20	3-NN	0.62	0.10	0.97	0.65	0.13	0.17	0.62
Concat	Linear SVM	0.60	0.14	0.89	0.46	0.05	0.22	0.38
ConcatDiff	25-NN	0.62	0.04	0.99	0.78	0.11	0.08	0.34
MeanSTD	9-NN	0.61	0.01	1.	1.	0.08	0.02	0.58
Moments	Log reg	0.63	0.07	1.	1.	0.21	0.13	0.74

Table A.3: Performance of the basic set of classifiers on the testing set S434, trained on the data set S3272 balanced with the reverse mutations, using the different features generated to represent the mutation records. The classifiers were evaluated for accuracy, recall, specificity, precision, MCC, F1 and ROC AUC scores on the testing set S430. For each feature set, the predictor with the highest MCC was chosen, and shown here. A general increase in most metrics, including the MCC values is observed with all features, but at a cost of reduced precision metrics.

Feature set	Best model	Acc	Sens	Spec	Prec	MCC	F1 score	ROC AUC
Diff	1-NN	0.75	0.82	0.70	0.64	0.51	0.72	0.76
Diff_0	5-NN	0.76	0.78	0.75	0.67	0.52	0.72	0.81
Diff_2	9-NN	0.75	0.75	0.75	0.66	0.49	0.70	0.81
Diff_5	Poly 3 SVM	0.75	0.73	0.77	0.67	0.49	0.70	0.82
Diff_10	Poly 3 SVM	0.76	0.75	0.78	0.68	0.52	0.72	0.83
Diff_20	Poly 3 SVM	0.77	0.73	0.79	0.70	0.52	0.71	0.82
Concat	Poly 3 SVM	0.73	0.49	0.88	0.72	0.41	0.59	0.77
ConcatDiff	Linear SVM	0.71	0.62	0.76	0.63	0.38	0.62	0.75
MeanSTD	1-NN	0.71	0.78	0.66	0.60	0.43	0.68	0.72
Moments	25-NN	0.76	0.68	0.81	0.70	0.50	0.69	0.80

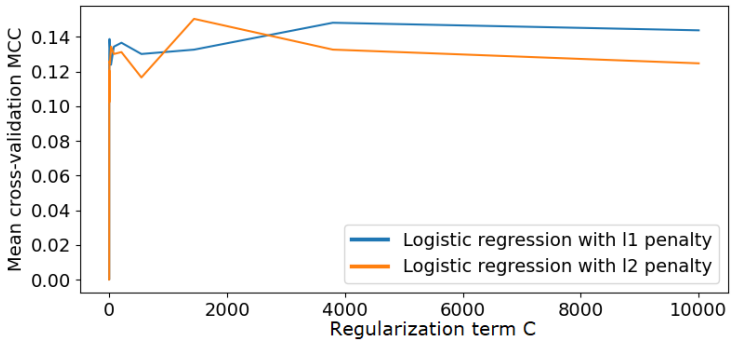

Figure A.10: Cross-validation hyper-parameter tuning process of the logistic regression predictor of protein thermostability changes upon mutations. The optimal mean test MCC of 0.150 was obtained with an l_2 penalty and a C value of 1438.45.

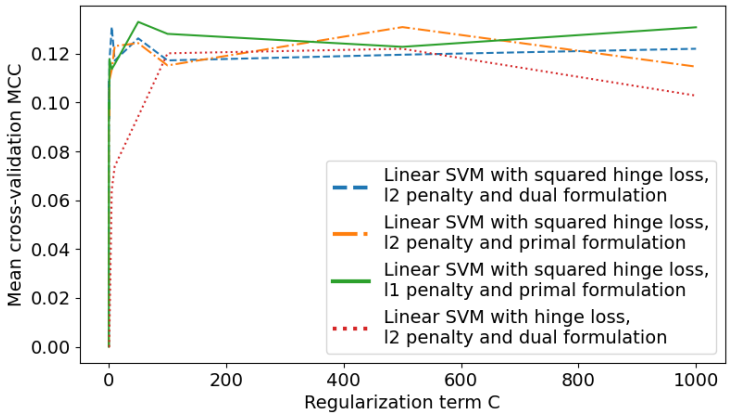

Figure A.11: Cross-validation hyper-parameter tuning process of the linear SVM predictor of protein thermostability changes upon mutations. The optimal mean test MCC of 0.133 was obtained with a squared hinge loss, an l_1 penalty and a C value of 50.

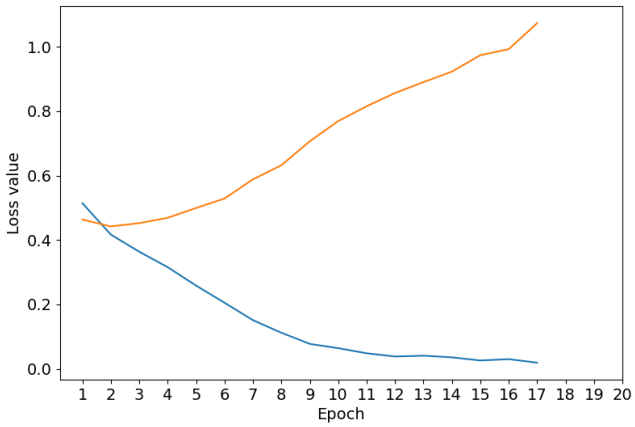

Figure A.12: Training history of the MLP predictor of protein thermostability changes upon mutations. This experiment trained the model for 17 epochs, and followed the training loss (blue) and the validation loss (orange) of the model at the end of each epoch. Serious overfitting issues can be observed when more than 3 training epochs were applied.

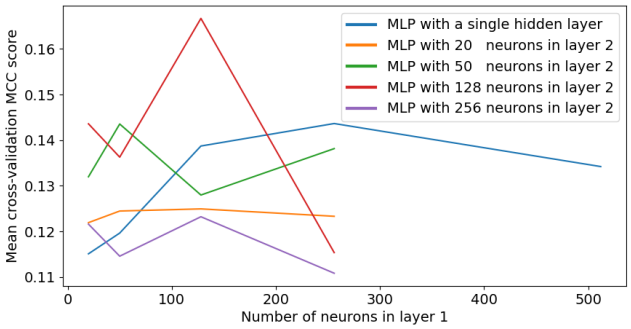

Figure A.13: Cross-validation hyper-parameter tuning process of the MLP predictor of protein thermostability changes upon mutations. Different number of neurons and layers were attempted, and the optimal mean test MCC of 0.167 was obtained with two hidden layers of 128 neurons each.

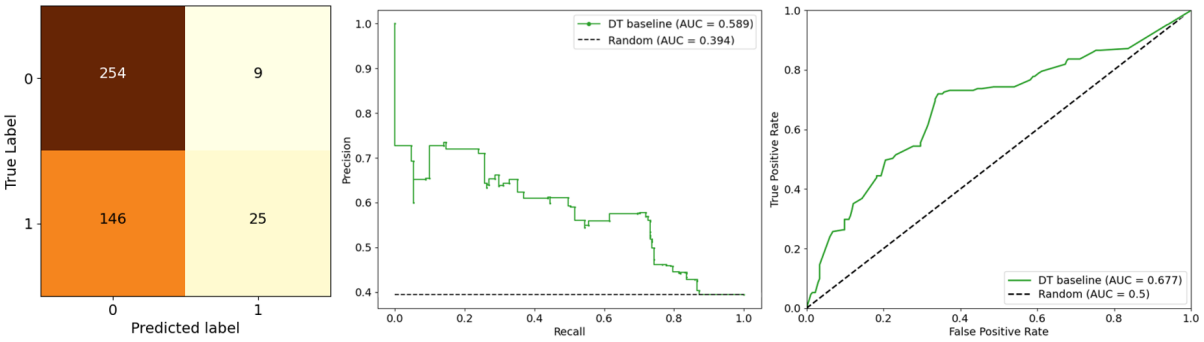

Figure A.14: Performance of the DT model with baseline features on the testing set S434. The confusion matrix shows a large bias towards the negative class, and the PRC shows the reduced precision of the model. The ROC curve is also worse than most of the models that used the SeqVec features.

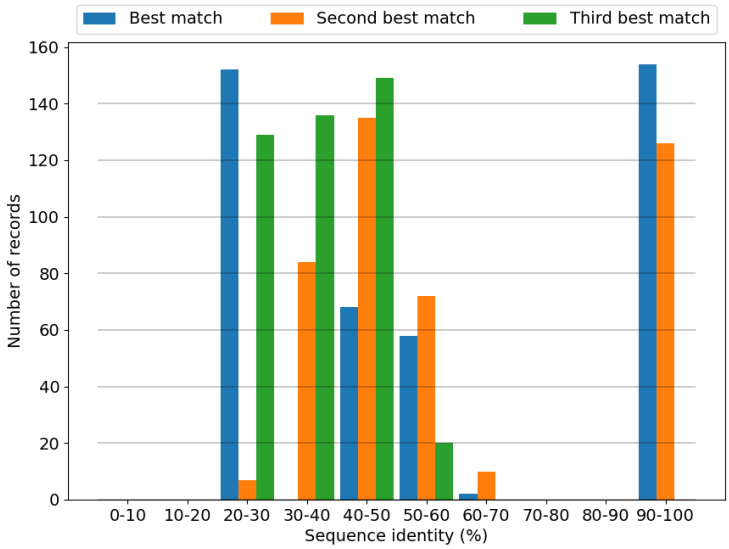

Figure A.15: Sequence identity between each of the sequences in the testing set S434 and their first, second and third best matches in the training set S3272, obtained by using the BLAST tool in Section 5.2.3.

A.3.3 Melting temperature prediction

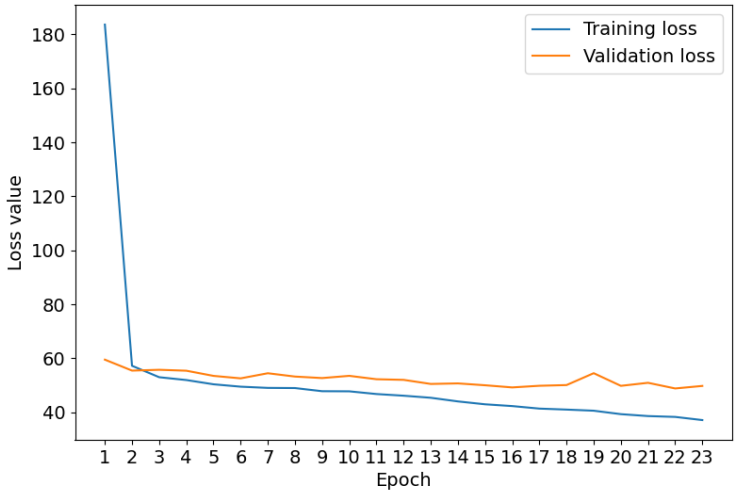

Figure A.16: Training history of the MLP predictor of protein melting temperature. This experiment trained the model for 23 epochs, and followed the training loss (blue) and the validation loss (orange) of the model at the end of each epoch. The availability of more data allowed the use of more training epochs before overfitting became an issue, and the value 23 was chosen after several attempts revealed that this value generally resulted in the least overfitting.

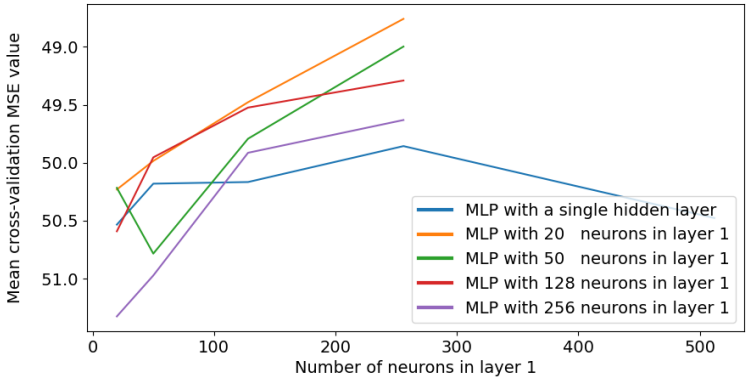

Figure A.17: Cross-validation hyper-parameter tuning process of the MLP predictor of protein melting temperature. Different number of neurons and layers were attempted, and the optimal mean test RMSE of 28.759 was obtained with two hidden layers of 256 and 20 neurons each.

Table A.4: Performance of a linear regression protein melting temperature model with different baseline features on the testing set. Representing each protein by an amino acid frequency vector produced the best baseline performance, with the lowest test RMSE value, and highest coefficient of determination r^2, EVS and correlation coefficients.

Baseline features	RMSE train	RMSE test	r2 train	r2 test	EVS train	EVS test	PCC train	PCC test	SCC train	SCC test
Sequence length	10.1	10.12	8.90e-3	1.12e-2	8.90e-3	1.12e-2	9.44e-2	1.06e-1	1.23e-1	1.39e-1
Amino acid count vector	9.62	9.53	1.14e-1	1.23e-1	1.14e-1	1.23e-1	3.36e-1	3.51e-1	2.26e-1	2.47e-1
Amino acid frequency vector	**9.07**	**8.88**	**2.13e-1**	**2.38e-1**	**2.13e-1**	**2.38e-1**	**4.61e-1**	**4.88e-1**	**2.00e-1**	**2.20e-1**
Polar amino acid frequency	9.75	9.66	8.86e-2	9.99e-2	8.86e-2	9.99e-2	2.98e-1	3.17e-1	1.24e-1	1.38e-1
Hydrophobic amino acid frequency	10.09	10.04	2.44e-2	2.68e-2	2.44e-2	2.68e-2	1.56e-1	1.64e-1	4.88e-2	4.41e-2

Table A.5: Train and test set performance of the melting temperature prediction MLP model implemented with the baseline feature set.

Model	*RMSE* train	*RMSE* test	r^2 train	r^2 test	*EVS* train	*EVS* test	*PCC* train	*PCC* test	*SCC* train	*SCC* test
MLP with amino acid freq. vector	7.22	7.14	0.50	0.51	0.53	0.53	0.73	0.73	0.43	0.43

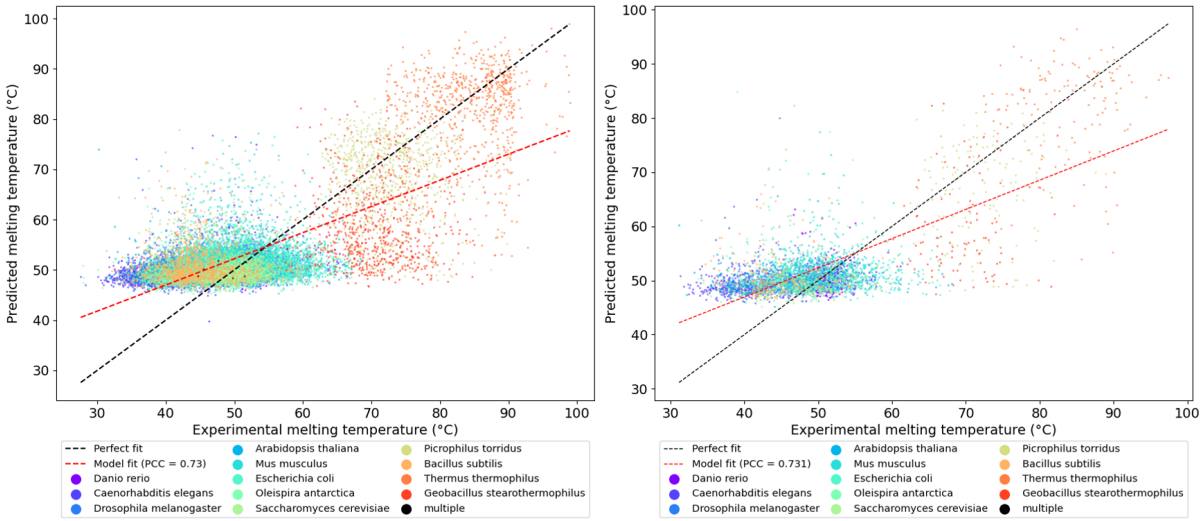

Figure A.18: Scatter plots of the melting temperatures predicted by the MLP model developed with the baseline feature set, and their true values. The performance on the training set (left) and on the testing set (right), show a worse prediction fit than when the SeqVec protein embeddings are used.

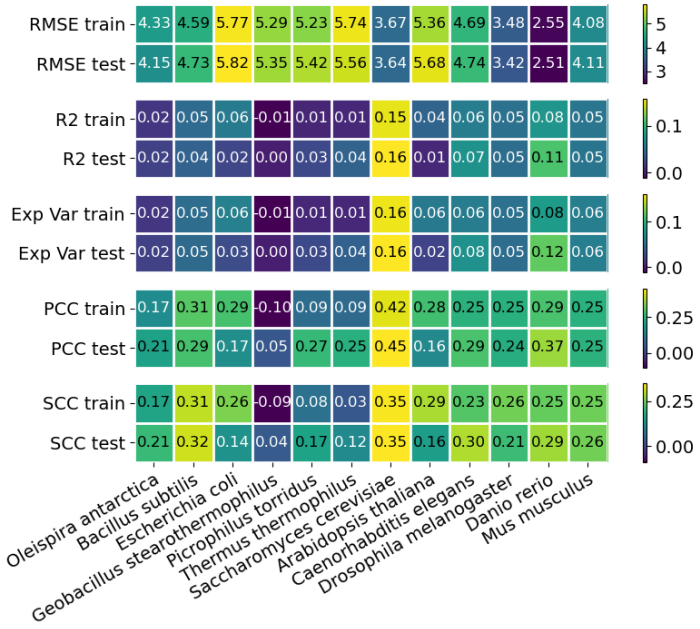

Figure A.19: Train and test RMSE, r^2, EVS, PCC and SCC performance metrics of the MLP model with the baseline feature set for melting temperature prediction, upon training and evaluation on individual organisms.

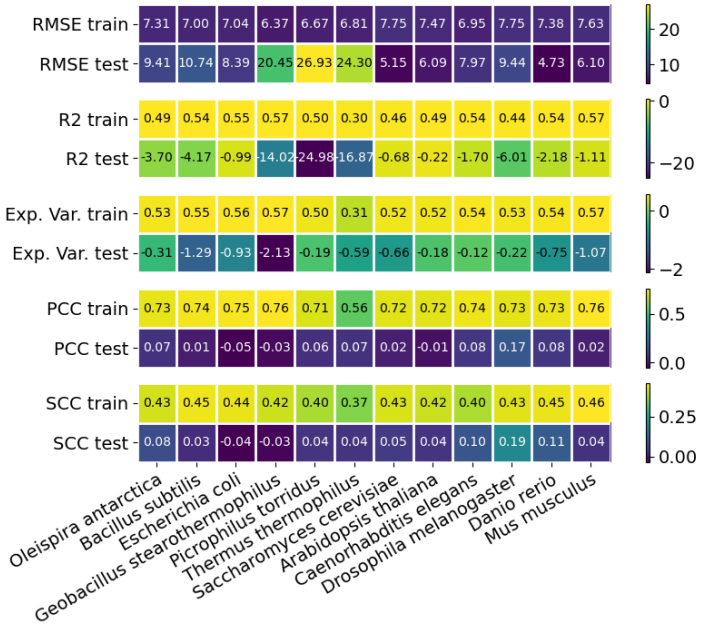

Figure A.20: Train and test RMSE, r^2, EVS, PCC and SCC performance metrics of the MLP model with the baseline features for melting temperature prediction, upon training on all organisms except one, which was used for evaluation.

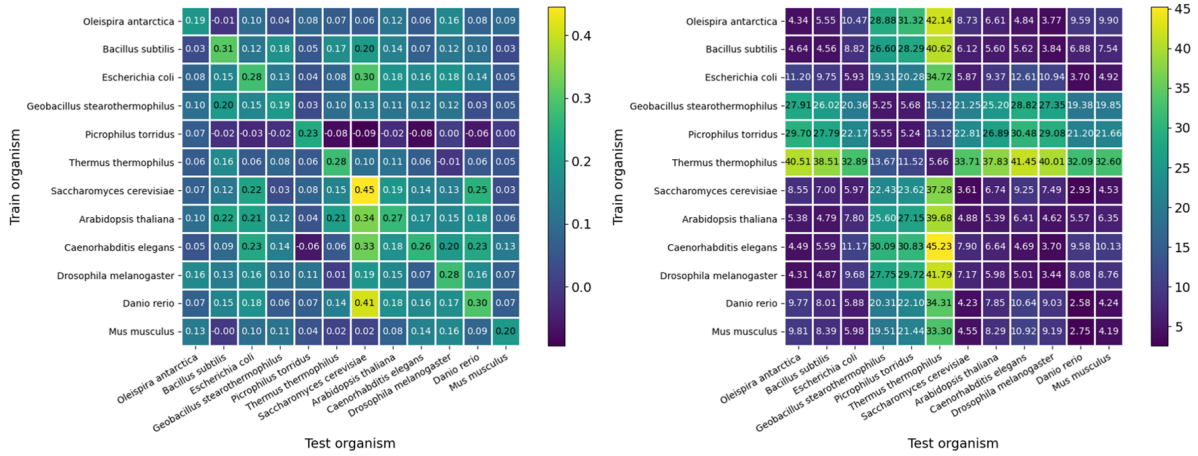

Figure A.21: Train and test PCC and RMSE performance metrics of the MLP model with the baseline features for melting temperature prediction, upon training on individual organisms and testing on all the others.

Table A.6: Performance of several naive baselines for melting temperature prediction on all species.

Naive baseline features	RMSE train	RMSE test	r2 train	r2 test	EVS train	EVS test	PCC train	PCC test	SCC train	SCC test
Species one-hot-encoding	4.53	4.50	0.80	0.80	0.80	0.80	0.90	0.90	0.72	0.72
Organism OGT	5.50	5.45	0.71	0.71	0.71	0.71	0.84	0.84	0.63	0.62
Organism average melting temperature	4.53	4.50	0.80	0.80	0.80	0.80	0.90	0.90	0.72	0.72
All of the above	4.53	4.50	0.80	0.80	0.80	0.80	0.90	0.90	0.72	0.72

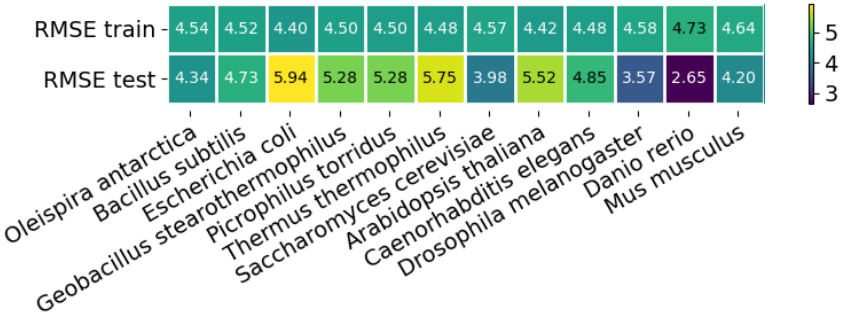

Figure A.22: Train and test RMSE values of a naive melting temperature linear regression baseline in the same leave-one-out experiment with the processed *Meltome Atlas* data set. The error values are better than those obtained by the MLP model with the SeqVec features.