HOCHSCHULE MITTWEIDA
University of Applied Sciences

Master's Thesis

Mr.
Julius Voigt

Prototype-based learning for sequences in molecular biology

Sensor Response Principle

2023

Faculty of **Applied Computer Sciences & Biosciences**

Master's Thesis

Prototype-based learning for sequences in molecular biology

Sensor Response Principle

Author:
Julius Voigt

Study Programme:
Genomic Biotechnology (M. Sc.)

Seminar Group:
GB20wB-M

First Reviewer:
Prof. Dr. rer. nat. habil. Thomas Villmann

Second Reviewer:
Dr. Marika Kaden

Mittweida, January 2023

"As Psmith would have said, he had confused the unusual with the impossible, and the result was that he was taken by surprise."

- P.G. Wodehouse, *Mike*

Bibliographic data

Voigt, Julius: Prototype-based learning for sequences in molecular biology,
University of Applied Sciences Mittweida,
Faculty of Applied Computer Sciences & Biosciences

Master's Thesis, 2022

Written with vim, typeset with LaTeX.

Abstract

Sequences are an important data structure in molecular biology, but unfortunately it is difficult for most machine learning algorithms to handle them, as they rely on vectorial data. Recent approaches include methods that rely on proximity data, such as median and relational Learning Vector Quantization. However, many of them are limited in the size of the data they are able to handle. A standard method to generate vectorial features for sequence data does not exist yet. Consequently, a way to make sequence data accessible to preferably interpretable machine learning algorithms needs to be found. This thesis will therefore investigate a new approach called the Sensor Response Principle, which is being adapted to protein sequences. Accordingly, sequence similarity is measured via pairwise sequence alignments with different sequence alignment algorithms and various substitution matrices. The measurements are then used as input for learning with the Generalized Learning Vector Quantization algorithm. A special focus lies on sequence length variability as it is suspected to affect the sequence alignment score and therefore the discriminative quality of the generated feature vectors. Specific datasets were generated from the Pfam protein family database to address this question. Further, the impact of the number of references and choice of substitution matrices is examined.

Acknowledgements

I would like to thank the following people for their support, knowing or unknowing:

Prof. Thomas 'Zwockl' 'Villy' Villmann for inviting me into his research group, through which I have met so many interesting people and because of which I am able to experience what scientific research is like. He has given me the chance and inspiration to work on many topics I would have never known about.

Marika Kaden for so much of her time, care, comprehensible insights and for holding the research group together.

Florian Heinke for sharing some of his vast knowledge with me over the last few years and for his enthusiasm for bioinformatics.

Julia for optimism and proofreading.

And everyone else in the group for the helpful discussions and a lot of fun. I have learned something relevant from each and every single one of you.

I would also like to thank friends and family for their encouragement.

Contents

Contents I

List of Figures II

List of Tables II

Abbreviations III

1 Introduction 1
- 1.1 Motivation . . . 1
- 1.2 Feature Generation . . . 2

2 Methods and basics 3
- 2.1 Sequence Alignment . . . 3
- 2.2 BLOcks SUbstitution Matrix . . . 7
- 2.3 Learning Vector Quantization . . . 8
- 2.4 Sensor Response Principle . . . 10

3 Experimental setup 13
- 3.1 Datasets . . . 13
- 3.2 Evaluation . . . 15

4 Experiments, results and discussion 17
- 4.1 The initial attempt . . . 17
- 4.2 Is the length of the sequences significant for classification? . . . 19
- 4.3 Choice of references . . . 25
- 4.4 Choice of substitution matrices . . . 30
- 4.5 Test with heterogeneous dataset . . . 31

5 Conclusions and future directions 35

Appendix 38

Bibliography 42

Glossary 47

See next page for abbreviations.

List of Figures

2.1 Structure of amino acid 4
2.2 Example for global alignment procedure 5
2.3 Example for local alignment procedure 6
2.4 Sequence identity example 8
2.5 SRP as schema for protein sequences 12

4.1 GNB accuracy on *Pfam8* 18
4.2 Length dependency in performance 20
4.3 NW alignment scores over protein families in *Pfam8_L200* 22
4.4 SW alignment scores over protein families in *Pfam8_L200* 22
4.5 NW alignment scores over protein families in *Pfam8_Lhybrid* . . . 23
4.6 SW alignment scores over protein families in *Pfam8_Lhybrid* . . . 24
4.7 Scatter plot of NW and SW scores for one reference 24
4.8 Accuracy depending on the number of references 26
4.9 Standard deviation of Needleman-Wunsch algorithm (NW) and Smith-Waterman algorithm (SW) scores in *Pfam8_L200* dataset . 28
4.10 Accuracy depending on the origin of references 29
4.11 Accuracy depending on the substitution matrices 31
4.12 Standard deviation of NW and SW scores in *Pfam8_L200* dataset . 31
4.13 Accuracy on *Pfam8* depending on the number of references. . . . 32
4.14 Accuracy on *Pfam8* with Generalized LVQ (GLVQ) and ?? 33

List of Tables

3.1 *Pfam8* dataset 14
3.2 *Pfam8_L200* and *Pfam8_Lhybrid* datasets 15

4.1 Accuracy with different number of references 27
4.2 Accuracy with different origins of references 30
4.3 Accuracy with different number of references for *Pfam8* 32

Abbreviations

ARS Attraction Repulsion Scheme

BLAST Basic Local Alignment Search Tool

BLOSUM BLOcks SUbstitution Matrix

CCM Classification Correlation Matrix

D -dimensional

DNA Deoxyribonucleic Acid

DTW Dynamic Time Warping

FASTA FAST-All algorithm

GLVQ Generalized LVQ

GMLVQ Generalized Matrix LVQ

GNB Gaussian Naive Bayes algorithm

indel **in**sertion/**del**etion

LVQ Learning Vector Quantization

MCC Matthews Correlation Coefficient

MSA Multiple Sequence Alignment

NW Needleman-Wunsch algorithm

RCSB Research Collaboratory for Structural Bioinformatics

RCSB PDB RCSB Protein Data Bank

RGLVQ Relational GLVQ

RLVQ Relational LVQ

rMIF resolved Mutual Information Function

RNA Ribonucleic Acid

SGD Stochastic Gradient Descent

SRP Sensor Response Principle

SW Smith-Waterman algorithm

Chapter 1

Introduction

1.1 Motivation

Machine learning methods are being applied to domain-specific data from an enormous range of fields, one of them being biology [Angra and Ahuja, 2017]. Biological data can be very complex and the underlying information quite difficult to extract [Zitnik et al., 2019]. Information often lies hidden behind relations of different elements in a graph-like structure [Ahmedt-Aristizabal et al., 2021]. Amino acids are one such type of element. They are the building blocks of life on the scale of proteins. Proteins are functional units that control the cell and carry out most of its intra- and intercellular functions [Zvelebil and Baum, 2007]. Proteins are therefore vital components of life that clearly must have an uncountable plurality of shapes and forms in order to be able to perform all kinds of different actions within and outside of our cells and those of every other species on earth that we know of.

The shape of a protein is associated with its function. The 3-dimensional (D) representation is not known for most proteins. The openly available RCSB Protein Data Bank (RCSB PDB) [Berman, 2000] contains 3D structural data for 171,077 proteins (PDB Stats) as of 18th October 2022. The UniProt Knowledgebase (UniProtKB) however contains 568,363 sequences of proteins that have been manually annotated and reviewed and a staggering 230,496,503 sequences of proteins that were automatically annotated [UniProt consortium, 2022]. This huge number of sequences is the result of billions of polynucleotides and polypeptides being sequenced by millions of researchers and lab technicians in laboratories all around the world. Sequencing is and always has been much easier hence cheaper and more accessible than protein structure determination. This will likely always be the case even with the recent development of AlphaFold [Jumper et al., 2021] as the structure prediction is not possible for all types of proteins and the folding of proteins depends on a plethora of different circumstances like the conditions of its environment to say the least. Further, protein sequences contain a lot of information already for they are not merely characters of an alphabet jumbled together but indeed collections in which order matters. The order in which amino acids are fused together with peptide bonds like a chain is crucial for the specific shape that this chain will take on.

The flow of information within cells from the genes to proteins, as described in the *Central Dogma of Molecular Biology*, is described through sequences of letters from different alphabets [Crick, 1970]. The sequence of proteins is essentially read off of the nucleic sequences within the genes. These nucleic

sequences have an alphabet of only four letters C, G, A and T for **C**ytosine, **G**uanine, **A**denine and **T**hymine whereas the basic alphabet of proteins has 20 letters, which stand for the different amino acids that are chained together to polypeptide chains, which in turn fold into proteins. The term *basic* is here meant to only include amino acids that are found to make up proteins in nature. The alphabet can be slightly larger, but is mostly still defined as those 20 amino acids. The folded polypeptide chain, finally, represents a functional unit within the cell with numerous functions and interactions with other biomolecules. During this flow, more information is required that goes into the translation of a gene to a protein sequence [Hanson and Coller, 2017], but this work will focus exclusively on protein sequences.

Summing up, protein sequences play an important role in bioinformatics. They symbolize one important way to store information about the components and the 2D structure of a protein.

1.2 Feature Generation

Machine learning algorithms often rely on linear algebra. Practically, this means that their input is required to be of a vectorial form. A lot of the information about biomolecules is, however, stored in various kinds of graphs, which are much more complex and fundamentally a different data structure. Protein sequences are a special kind of graph called a path graph where the vertices are elements of the set of amino acids $\mathcal{A}$. All elements in $\mathcal{A} = \{\text{A, R, N, C, Q, E, G, H, I, L, K, M, F, P, S, T, W, Y, V}\}$ stand for different amino acids, specified in [JCBN, 1984]. Data of this form can not readily be used as input for most machine learning algorithms, especially those that are based on distance measures. In the case of protein sequences, several attempts at a so-called feature generation have been proposed. Some do not rely on an alignment of the sequences, like resolved Mutual Information Function (rMIF) [Bohnsack et al., 2021], natural vectors [Wang et al., 2019] and bag of words (amino acid alphabet too large) [Blaisdell, 1989], some of which are systematized in [Bohnsack, 2020]. Other methods are based on the proximity of data and can make use of sequence alignments.

Two attempts to utilize alignment scores of protein sequences have been made as part of a 6-month research module. First, an integration of Dynamic Time Warping (DTW) into Learning Vector Quantization (LVQ) [Jain and Schultz, 2018] was investigated, but many challenges were encountered. Further, solutions involving Relational LVQ (RLVQ) [Hammer et al., 2014] were explored. Descriptions and discussions of both can be found in the report on the research module, upon request to the first referee or voigt5@hsmw.de. The main downside of both is the computational intensity. Both approaches require the calculation of *all* pairwise alignments, be it by DTW or any other alignment algorithm.

This work will examine yet another approach, called the Sensor Response Principle (SRP), which will be introduced in section 2.4.

Chapter 2

Methods and basics

2.1 Sequence Alignment

Sequences in molecular biology are a very important vehicle with which to convey information about a protein and its components (amino acids), as established above. Not only are all the amino acids described that are involved in shaping the protein, but their relation to one another is also described. To be more precise, the order in which they are chained together linearly with peptide bonds from beginning to end is specified.

It stands to reason that a comparison of protein sequences would yield some information about their structural relationship, their evolutionary relationship or even their functional relationship. There are a myriad of ways to compare sequences with each other. Some of the ones for nucleic sequences can be found systematized in [Bohnsack, 2020]. The methods described herein all have to do with sequence alignments.

Broadly, a set of sequences, be it polypeptides (protein) or nucleotides (Deoxyribonucleic Acid (DNA)/Ribonucleic Acid (RNA)), is being arranged in a way that adjoins similar subsequences. Many algorithms have been devised to align two or more sequences with each other in an optimal or a heuristic way [Zvelebil and Baum, 2007, p. 128]. Some of them will be applied and discussed here.

Whichever algorithm is employed to determine the alignment, the quality of it must be evaluated in order to assess its soundness as a measure of similarity. This is done by calculating a score of the alignment. An alignment of very similar and/or related sequences will then have a high score and an alignment of two random sequences a low score. Further, the optimal alignment(s) will have the highest score within the space of possible alignments. This score is important for the design of an alignment algorithm and is usually what it is being optimized for. A perfect scoring scheme likely does not exist because it has to take all evolutionary processes into account. Yet, the abovementioned assumptions are true most of the time for common scoring schemes.

One of the simplest ones is sequence identity. It describes the percentage of identical matches along the aligned sequences. This way, all matches are rewarded and all mismatches penalized equally. That does not, however, reflect the probabilities for mutations that are seen in nature. And an explanation is also not too far off: amino acids have properties. These properties are determined by the residue of each amino acid.

Some residues are more similar to each other than others, *i.e.* they all have their effect on the structure and function of the protein but when one amino

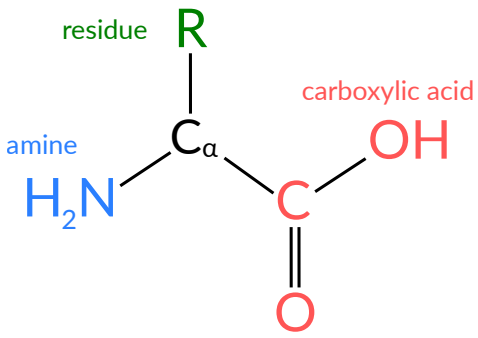


Figure 2.1: Schematic structure of an amino acid. Amino acids consist of an amino and a carboxylic acid with a so-called residue R attached to the alpha carbon C_α.

acid is switched out for another physico-chemically more similar amino acid, it does not result in so big a structural or functional change. Some residues, for example, are acidic and others are basic. Some are polar, others non-polar [Yoo et al., 2014]. The probability of a certain amino acid to be changed to another, owing to mutation, and producing a still functioning protein structure can be empirically observed with the help of statistics [Henikoff and Henikoff, 1992]. The resulting substitution matrices are an important resource for the scoring of alignments [Hess et al., 2016]. The set of sequences underlying the statistical analysis can be on a spectrum from almost identical and clearly homologous to almost wholly different and merely predicted homologous. The underlying dataset plays an important role as well [Keul et al., 2017]. The goal is to integrate as much background knowledge as possible into the process of alignment and that knowledge comes from the theory of evolution. This approach brings about a wide variety of substitution matrices that can be used for alignments of protein sequences in the context of different bioinformatic questions or hypotheses. The selection of the best substitution matrix for a given problem is naturally associated with expert knowledge, although there is a tendency to use the default, which is called BLOSUM62, for many cases.

The algorithm used to calculate the alignment presents an additional aspect. There are essentially two types of alignments: global and local. A global alignment attempts to align sequences over their whole length. Contrarily, a local alignment seeks to make out the parts of sequences that are related. A global alignment is particularly useful for sequences that are closely related and have approximately the same length, whereas a local alignment is better if the sequences are only partly similar, *e.g.* when a domain is present in both that might have been conserved in both proteins [Zvelebil and Baum, 2007, pp. 135-136]. The most basic way to calculate global and local alignments are the Needleman-Wunsch algorithm (NW) [Needleman and Wunsch, 1969] and the Smith-Waterman algorithm (SW) [Smith and Waterman, 1981] respectively. Both employ dynamic programming schemes to optimize the alignment of sequences by identifying matches between them and by inserting gaps into them. The reward of matches and the penalty cost of mismatches or **in**sertions/**del**etions (indels) influence the resulting alignment and score. The score in turn translates into a measure of similarity for sequences. The

performance of the similarity measure via this method, thus, depends strongly on the substitution matrix and gap costs.

As for the algorithms themselves, it can be said that both do the same, only with slightly different constraints. They are summing up all rewards and penalties along the alignment of two sequences. The way they work is that an optimization matrix is constructed, with the elements of sequence 1 along the first axis (as rows) and the elements of sequence 2 along the second axis (as columns). Then, starting from the first element of this matrix, *i.e.* top left, the rewards/costs of the element pair is evaluated. Based on what is best for only this position, the value of reward or cost for match or mismatch/indel is recorded in the matrix. Rewards or penalty costs for match or mismatch $s(x_i, y_j)$ are taken from the substitution matrix. The cost for introducing a gap in the case that an indel is preferable is defined by either a gap cost that is preset or by a more sophisticated gap cost model, such as *affine gap cost*. With the elected decision recorded in the optimization matrix, one moves an element further and the next element pair is evaluated, taking the previous decision into account. The alignment is consequently constructed from small optimal alignments step by step. A global alignment grows from the very beginning to the end of both sequences. The procedure can be seen acted out on a simple example in figure 2.2.

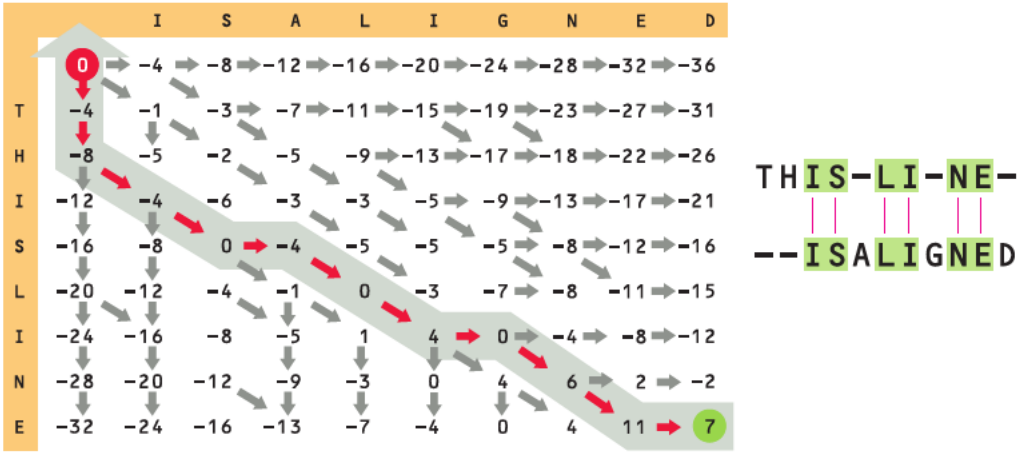


Figure 2.2: Simple example for global sequence alignment. A preset gap cost of −4 was used in conjunction with the BLOSUM62 substitution matrix. On the right is the finished optimal alignment, as determined by the traceback (big gray arrow) from the last matrix element all the way to the first. Adapted from [Zvelebil and Baum, 2007, p. 131].

A local alignment can begin anywhere in the optimization matrix. That is achieved by introducing the constraint that the lowest value is 0. This acts as a reset from where a new local alignment can begin. The local alignment procedure is shown on an example in figure 2.3.

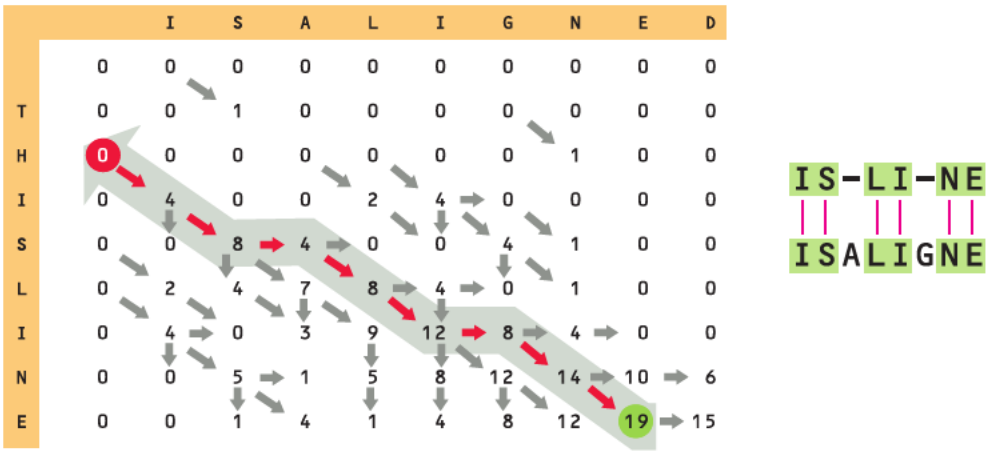


Figure 2.3: Simple example for local sequence alignment. A preset gap cost of -4 was used in conjunction with the BLOSUM62 substitution matrix. On the right is the finished optimal alignment, as determined by the traceback (big gray arrow) starting at the greatest matrix element and ending at the first encountered 0. Adapted from [Zvelebil and Baum, 2007, p. 136].

The algorithms as described above are often written as the following recursive formulas that construct the optimization matrix:

$$F_{i,j} = \max \begin{cases} F_{i-1,j-1} + s(x_i, y_j) \\ F_{i-1,j} + g \\ F_{i,j+1} + g \end{cases} \qquad \text{(Needleman-Wunsch)} \qquad (2.1)$$

$$F_{i,j} = \max \begin{cases} F_{i-1,j-1} + s(x_i, y_j) \\ F_{i-1,j} + g \\ F_{i,j+1} + g \\ 0 \end{cases} \qquad \text{(Smith-Waterman)} \qquad (2.2)$$

$F_{i,j}$ denotes the element in row i, column j of the optimization matrix. The reward/cost of the match/mismatch between the i^{th} element of sequence 1 and j^{th} element of sequence 2 is denoted $s(x_i, y_j)$. It is found in the selected substitution matrix, which represents a substantial set of parameters. The choice of gap penalty g is consequential and adds more parameters to those in the substitution matrices, which is why a static gap penalty was used in this thesis. Many optimizations have been proposed, that make calculations faster, but they yield potentially suboptimal alignments. Besides adaptations to NW or SW there exist heuristic algorithms, such as FASTA [Pearson and Lipman, 1988] and Basic Local Alignment Search Tool (BLAST) [Altschul et al., 1990] to name only the two most influential ones.

Only the alignment scores, not the alignments themselves, are of interest in this thesis. They serve as a similarity measure for sequences and are input for machine learning algorithms. One of the algorithms is introduced in the following section.

2.2 BLOcks SUbstitution Matrix

The comparison of protein sequences has long been an important tool for molecular biologists and bioinformaticians. The attempts to measure protein sequence similarity are aided by sequence alignment algorithms, such as the previously mentioned NW and SW algorithms. Those two and also many others align sequences with the help of substitution matrices. There are numerous ways to construct such substitution matrices, based on different rationales and datasets [Altschul, 1991] (for an overview, see [Trivedi and Nagarajaram, 2020]). One methodology, however, has produced a family of substitution matrices which is the most frequently used, the BLOcks SUbstitution Matrix (BLOSUM) [Hess et al., 2016].

It is based on counting the substitutions of pairs of amino acids in a specific set of aligned protein sequences, which are aligned in so-called *blocks*. Blocks are local, ungapped Multiple Sequence Alignments (MSAs) which have been derived from highly conserved regions from the BLOCKS database [Henikoff and Henikoff, 1991]. In order to avert the many redundancies in the blocks, which are likely to occur when the sequences are very similar, clustering is performed. Each cluster consists of all sequences that are identical to a certain degree to one or more other sequences in the same cluster. The clusters within the blocks are then weighed as a single sequence during the counting. For a block of width w and s sequences, a total of $ws(s-1)/2$ amino acid pairs can be counted. Note that the direction of the mutational events, *i.e.* the order of the protein sequences, are not considered with this method and the frequency of changes from an amino acid i to j is the same as for j to i. The substitution frequencies are then determined for all pairs of the 20 different amino acids. That means that for all $20 + 19 + \cdots + 1 = 210$ different amino acid pairs, the pairings of those two amino acid letters i and j across a column are counted, then summed up over all columns. This is repeated in all blocks and the frequencies f_{ij} are stored within a table. The observed probability of the occurrence for the pair i and j then amounts to

$$q_{ij} = \frac{f_{ij}}{\sum_{i=1}^{20}\sum_{j=1}^{i} f_{ij}}. \tag{2.3}$$

Next to the observed probabilities, the expected probabilities are required as well, because the scores for the amino acid pairs in BLOcks SUbstitution Matrixs (BLOSUMs) are logarithms of odds. The probability that any amino acid i is part of a pair is

$$p_i = q_{ii} + \sum_{j \neq i} q_{ij}/2 \tag{2.4}$$

The expected probabilities of occurrence e_{ij} can then be obtained as $p_i p_j$ if $i = j$ or as $p_i p_j + p_j p_i = 2p_i p_j$ if $i \neq j$. Finally, the log odds ratio can be calculated as

$$s_{ij} = \log_2(q_{ij}/e_{ij}) \tag{2.5}$$

which is subsequently often multiplied by the scaling factor 2 (to obtain half-bit units) and rounded to the nearest integer. [Henikoff and Henikoff, 1992].

Pairwise sequence identity is measured by counting all matching amino acids in an alignment of the two sequences divided by the total length of the alignment, as illustrated in figure 2.4.

Figure 2.4: Alignment example. Here, two short sequences were aligned optimally according to the sequence identity. With 10 matching (identical) loci and a total length of 16, sequence identity amounts to $10/16 = 62.5\%$. Figure from [Zvelebil and Baum, 2007, p. 80].

The most frequently used BLOSUM is BLOSUM62. It is generated, as described above, with a minimal sequence identity of 62% in each cluster. Additionally, 4 other BLOSUMs are used within this work and can be found in the appendix. They were chosen because they have been the substitution matrices available by default in the highly influential BLAST [Altschul et al., 1990]. They have been taken from the BLAST source code and can be found at https://ftp.ncbi.nlm.nih.gov/blast/matrices/.

2.3 Learning Vector Quantization

Learning Vector Quantization (LVQ) is a prototype-based learning scheme that was introduced by Teuvo Kohonen [Kohonen, 1986]. It may be categorized as supervised learning in the interest of learning a classification task.

Learning Vector Quantization 2.1

The most well-known variant of Learning Vector Quantization (LVQ) that adapts some improvements over the original approach is LVQ2.1. It works as follows:

Let $T = \{(\mathbf{x}_i, c(\mathbf{x}_i)) \in X \times C, i = 1, \dots, n_X\}$ be a labeled training dataset where the data $X \subset \mathbb{R}^d$ is labeled with class labels $c(\mathbf{x}_i) \in C$ from the set of classes C. A set of labeled prototypes
$W = \{(\mathbf{w}_i, c(\mathbf{w}_i)) \in W \times C, i = 1, \dots, n_W\}$ is then initialized randomly in the data space, *i.e.* $W \subset \mathbb{R}^d$. Prototypes can be used to classify data by assigning the label of the closest prototype. The closest prototype is determined by calculating the squared Euclidean distance between a data point and all prototypes, as in equation 2.6. The closest prototype to an $\mathbf{x}_i$ is called the winner and has the index $s(\mathbf{x}_i)$.

$$s(\mathbf{x}_i) = \underset{j}{\operatorname{argmin}} \|\mathbf{x}_i - \mathbf{w}_j\|^2 \tag{2.6}$$

The prototypes undergo updates during training, which is done iteratively. First, a data point $\mathbf{x}_i$ is selected randomly and the winning prototype $\mathbf{w}_{s(\mathbf{x}_i)}$ is

being determined. $\mathbf{w}_{s(\mathbf{x}_i)}$ is henceforth only called $\mathbf{w}_s$. If the label $c(\mathbf{w}_s)$ is the same as $c(\mathbf{x}_i)$ then $\mathbf{w}_s$ (also denoted as $\mathbf{w}^+$) gets pulled towards the data point. If, in turn, $c(\mathbf{w}_s) \neq c(\mathbf{x}_i)$, then $\mathbf{w}_s$ (also denoted as $\mathbf{w}^-$) gets pushed away from $\mathbf{x}_i$. This is called the Attraction Repulsion Scheme (ARS) and is described in the following equations:

$$\mathbf{w}_s(t+1) = \mathbf{w}_s(t) - \alpha(t)(\mathbf{x}_i - \mathbf{w}_s(t)) \quad \text{if } c(\mathbf{w}_s) = c(\mathbf{x}_i) \tag{2.7}$$

$$\mathbf{w}_s(t+1) = \mathbf{w}_s(t) + \alpha(t)(\mathbf{x}_i - \mathbf{w}_s(t)) \quad \text{if } c(\mathbf{w}_s) \neq c(\mathbf{x}_i) \tag{2.8}$$

Here, t signifies the time step and $\alpha(t)$ is a learning rate which can be changed over time. The prototypes are updated in this manner until they converge or until the iteration is aborted manually or after a specified number of repeats.

Generalized LVQ

A generalized variant, the Generalized LVQ (GLVQ), was proposed in 1995 that minimizes a differentiable cost function S that is approximating the classification error through Stochastic Gradient Descent (SGD) [Sato and Yamada, 1995]. As with LVQ2.1, a data point $\mathbf{x}_i$ is selected randomly and the squared Euclidean distance d is calculated to every prototype. In contrast to LVQ2.1, two prototypes are then chosen:

- $\mathbf{w}_i^+$, which represents the winning prototype from the set of prototypes of the same class as $\mathbf{x}_i \quad c(\mathbf{w}_i^+) = c(\mathbf{x}_i)$
- $\mathbf{w}_i^-$, which represents the winning prototype from the set of prototypes with a different class from that of $\mathbf{x}_i \quad c(\mathbf{w}_i^-) \neq c(\mathbf{x}_i)$

The cost function consists of the classification term

$$\mu(\mathbf{x}_i) = \frac{d_i^+ - d_i^-}{d_i^+ + d_i^-} \tag{2.9}$$

which is wrapped in a monotonously increasing function f and summed up over all data points:

$$S_{GLVQ} = \sum_{i=1}^{N} f(\mu(\mathbf{x}_i)) \tag{2.10}$$

$f(\mu(\mathbf{x}_i))$ is the local error regarding to the datapoint x_i. d_i^+ and d_i^- denote the squared Euclidean distances of $\mathbf{x}_i$ to $\mathbf{w}^+$ and $\mathbf{w}^-$ respectively. It therefore holds that $\mu(\mathbf{x}_i) < 0$ whenever the classification is correct for a given data point ($d_i^+ < d_i^-$) and $\mu(\mathbf{x}_i) > 0$ Whenever the classification is wrong ($d_i^+ > d_i^-$). Overall $\mu(\mathbf{x})$ stays between -1 and 1.

The update of the prototypes $\mathbf{w}_i \in W$ is realized with SGD as follows:

$$\mathbf{w}_i^{\pm} \leftarrow \mathbf{w}_i^{\pm} - \alpha \frac{\partial f(\mu(\mathbf{x}_i))}{\partial \mathbf{w}_i^{\pm}} \tag{2.11}$$

α denotes, once again, a learning rate that can be changed over time. The prototypes are updated in this manner, minimizing the cost function S_{GLVQ}, until the algorithm converges or the procedure is aborted manually.

The design of Generalized LVQ with the ARS and using SGD makes it a robust [Saralajew et al., 2019], fast and flexible [Villmann et al., 2016] machine learning algorithm. The fact that the prototypes live in the data space makes it interpretable by domain experts. A plethora of variations on the concept of LVQ and GLVQ have been developed to adapt it to different problems and requirements.

One important change to the algorithm is the way prototypes are initialized. There are numerous ways besides the random initialization. One strategy that is often used is to initialize the prototypes on the class means. Thus, if the data is structured in clusters, prototypes already have a good starting point.

Generalized Matrix LVQ

Another adaption concerns the distance function of GLVQ. The squared euclidean distance $d^2(\mathbf{x}, \mathbf{w}) = \|\mathbf{x} - \mathbf{w}\|^2$ that was introduced in equation 2.6 gets replaced by the squared omega distance

$$d^2_{\Omega}(\mathbf{x}, \mathbf{w}) = (\mathbf{x} - \mathbf{w})^T \mathbf{\Omega}^T \mathbf{\Omega} (\mathbf{x} - \mathbf{w}) \tag{2.12}$$

with the relevance matrix $\mathbf{\Omega}$, which gives the algorithm the name Generalized Matrix LVQ (GMLVQ) [Schneider et al., 2009]. The elements of matrix $\mathbf{\Omega} \in \mathbb{R}^{m \times n}$ with $m \leq n$ are updated during training as well. The multiplication with matrix $\mathbf{\Omega}$ represents a linear transformation of the feature input space to $\mathbb{R}^m$ [Bunte et al., 2012]. The so-called *Classification Correlation Matrix (CCM)* $\mathbf{\Lambda}$ can then be derived by $\mathbf{\Lambda} = \mathbf{\Omega}^T \mathbf{\Omega}$ with $\mathbf{\Lambda} \in \mathbb{R}^{n \times n}$. From [Biehl et al., 2016]: "Diagonal entries of $\mathbf{\Lambda}$ control the importance of single feature dimensions in the distance and can account for their potentially different scaling. Off-diagonal elements relate to the contribution of pairs of features". That means that it is possible to interpret $\mathbf{\Lambda}$ to gain insights into the amount of influence each feature has on classification and therefore how important the feature is for the GMLVQ model.

All GLVQ and GMLVQ models in this thesis have been trained with the help of ProtoTorch [Ravichandran, 2020], ProtoTorch Models, PyTorch Lightning and were evaluated with scikit-learn [Pedregosa et al., 2011].

2.4 Sensor Response Principle

The just-mentioned GLVQ algorithm, like many other machine learning algorithms, requires input in vectorial form, as indicated briefly in section 1.2. A common and abundant data format in molecular biology is that of a sequence, be it a nucleic sequence or a protein sequence. Protein sequences are aligned to each other frequently in bioinformatics in order to measure their similarity and detect evolutionary relationships. Looking at protein sequence similarity from a machine learning perspective, those similarity measurements span a proximity space. There are machine learning algorithms that can exploit proximity data or (dis-)similarity data, such as Median GLVQ [Nebel et al., 2015] or Relational GLVQ (RGLVQ) [Hammer et al., 2014], but with serious limits on the size of

the dataset. As discussed in the report on the research module, both need a lot of time to calculate dissimilarities and also a lot of memory to store them. A much more sparse approach is demonstrated in [Bohnsack et al., 2022] on graph kernels, which are inner products for graphs that can be used as measurements to compare their similarity to each other. The so-called Sensor Response Principle (SRP) is inspired by sensor fusion, as in [Zoghlami et al., 2021]. It only calculates the proximity measure to a small subset of the original data in order to approximate it and is therefore many times faster, smaller and more efficient. The original data is then only represented by their proximity to the small subset, which is henceforth called *references*. Sequence alignment scores can be used as the proximity measure in the case of protein sequences to which the SRP may, thus, be adapted in the following way.

Let there be a set of protein sequences $\mathcal{P}$, with each element in $\mathcal{P}$ being a potential reference $r \in \mathcal{P}$. Accordingly, one sequence r is picked and the alignment score calculated to all $x \in \mathcal{P}$. Multiple alignment algorithms may be used with multiple sets of parameters, *e.g.* the substitution matrices used. They represent different kinds of sensors that measure the proximity/similarity $d(x, r)$ between all the data points x and the reference r. All data points are thereupon described by the sensor response vector $\mathbf{x} = \mathbf{d}(x, r)$, which consists of the different sensor responses $d(x, r)$ to the reference r:

$$\mathbf{d}(x, r) = (d_{AS_1}(x, r), d_{AS_2}(x, r), \dots, d_{AS_n}(x, r))^T \tag{2.13}$$

where $d_{AS_1} \dots d_{AS_n}$ stand for the different alignment scores calculated by various algorithms and/or parameter sets. Consequently, a vectorial sensor response space $\mathcal{X}$ emerges in which every protein sequence $x \in \mathcal{P}$ is represented by a feature vector $\mathbf{x} \in \mathcal{X}$ that relates it to reference r. Figure 2.5 shows a schematic summary of the approach on an example that aligns a sequence x and the reference r with NW, SW and 3 different BLOSUMs. The result is the 4D $\mathbf{x}_{sequence}$ which defines x in the vectorial feature space.

Besides multiple alignment algorithms and different parameters, the set of references may also be added to in order to increase the dimensionality of the feature space $\mathcal{X}$. But, because the choice of the best reference is difficult to estimate and may depend on the kind of sensor, finding an optimal set of multiple references is even more difficult. In this work, multiple references will be chosen at random and the distribution of their performance in aiding classification will be evaluated. Strategies to find optimal references or sets of references may include carrying out the sensor measurement for all of them, which is, depending on the dataset, very time-extensive and would neutralize one of the advantages of the SRP.

Given a set of protein sequences $\mathcal{P}$ with $|\mathcal{P}| = N$ sequences, N_{AS} different combinations of alignment algorithms and their parameters/substitution matrices, and N_r references, then the time complexity amounts to $O(N \cdot N_{AS} \cdot N_r)$ with $N_r \ll N$. Both mentioned alignment algorithms, NW and SW, have a time complexity of $O(mn)$ depending on the lengths m and n of the sequences. The SRP therefore saves a lot of time compared to, *e.g.* RGLVQ and Median GLVQ, that need *all* pairwise alignment scores which corresponds to $O(\frac{N^2}{2} \cdot N_{AS})$. Space complexities behave accordingly and all the advantages are passed on to the training of the classifier.

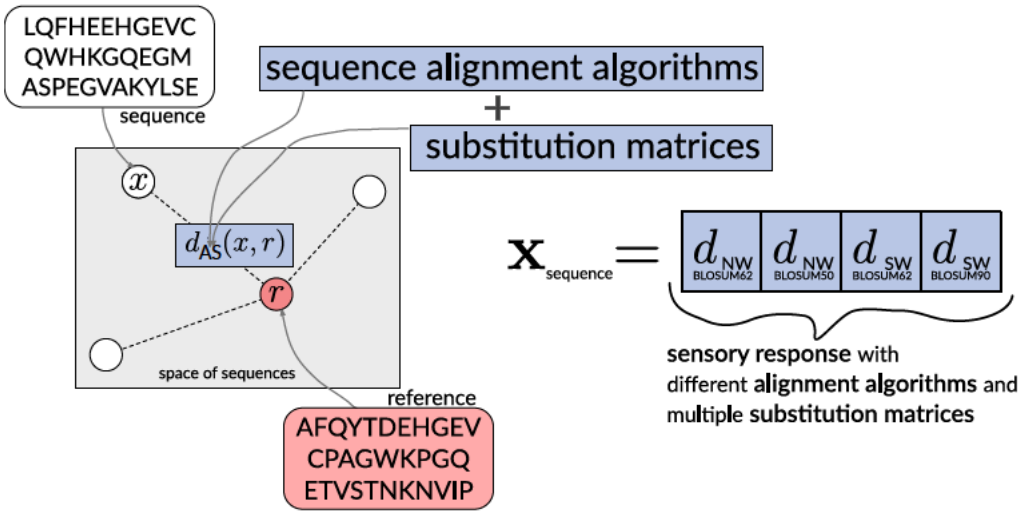


Figure 2.5: Schematic summary of the SRP for protein sequences. Space of sequences with a reference sequence r and another sequence x are shown. Choice of alignment algorithm and substitution matrix influence the sensor measurement $d_{AS}(x, r)$. In this example, NW, SW and 3 different BLOSUMs are used. Adapted from [Bohnsack et al., 2022].

Chapter 3

Experimental setup

The following chapter is intended to highlight the most important aspects of the experimental setup. Code and data will be made available under https://github.com/si-cim/SRPMastersThesis.

3.1 Datasets

The objective was to find a dataset containing labeled protein sequences to test the SRP on. Several datasets were generated in a non-deterministic manner by parsing the Pfam database [Mistry et al., 2020]. Pfam is a database dedicated to the classification of protein sequences into protein families and domains. Most of the entries in Pfam are manually annotated with functional information from literature. All data was only picked from the manually curated part. Pfam announced that "The Pfam website will be decommissioned in January 2023." and they have recently started to forward all users to its new home, the InterPro [Blum et al., 2020]. All Pfam links within this thesis will therefore lead to InterPro directly. At the time of retrieval (Pfam 35.0, February 2022), 19632 protein domains were present with many different protein sequences each (over 44 million in total). No expert knowledge was taken into account during the compilation of the datasets. This means that the domains that were chosen have not been looked at closely to check whether they are particularly close or far from each other in the space of all domains. The decision to stay away from such considerations was made because it was unclear whether it would be necessary for the experiments that domains should have a distinct relationship to each other. It is however conceivable that the chance to pick either very similar or dissimilar domains is very small. They are likely also no perfect representatives of the space of protein domains, but rather a naturally random collection.

The first dataset is entitled *Pfam8*. Powers of 2 were used throughout the composition process of all datasets to make handling easier. *Pfam8* is described in table 3.1 and consists of $2^3 = 8$ randomly chosen domain families. 2^{10} protein sequences were randomly chosen for each of these families. Sequences within a family are homologous and likely quite similar. The dataset has a total of 2^{13} sequences and classes are uniformly distributed. None of the families in *Pfam8* is a member of the same clan as any other. It is possible, that more than one classified domain is present in a single region of a protein sequence and that this protein is then a member of two families. This is, however, prevented during curation, unless the domains are in the same clan [Punta et al., 2011].

Since this is not the case, all chosen sequences will be member of only one class. Since the sequences in *Pfam8* were picked randomly from Pfam, they are of different lengths (175.0 ± 108.9 amino acids).

domain ID	clan ID	domain description
PF00310	CL0052	NTN GATase_2 Glutamine amidotransferases class-II
PF00370	CL0108	Actin_ATPase FGGY_N FGGY family of carbohydrate kinases, N-terminal domain
PF01454	CL0123	HTH MAGE MAGE homology domain
PF02230	CL0028	AB_hydrolase Abhydrolase_2 Phospholipase/Carboxylesterase
PF07179	-	SseB SseB protein N-terminal domain
PF07694	CL0315	Gx_transp 5TM-5TMR_LYT 5TMR of 5TMR-LYT
PF13290	CL0159	E-set CHB_HEX_C_1 Chitobiase/beta-hexosaminidase C-terminal domain
PF13415	CL0186	Beta_propeller Kelch_3 Galactose oxidase, central domain

Table 3.1: The *Pfam8* dataset. IDs of each protein family and their clan membership are shown, along with a short description. 1024 sequences were sources at random from each of the protein families.

Another two datasets were generated by almost the same procedure. In order to investigate the role that sequence length plays for the proposed feature extraction method, a constraint was set on the length of the sequences from Pfam. The first dataset consists, again, of 8 domain families with 1024 sequences each, but with the condition that all sequences are of length L with $190 < L < 210$. It is called *Pfam8_L200*. For the second dataset 4 families from *Pfam8_L200* were copied over and 4 more families were randomly picked with sequences of length L with $390 < L < 410$. It is called *Pfam8_Lhybrid*. The specific domain families can be seen in table 3.2. All data files will be made available under https://github.com/si-cim/SRPMastersThesis. Once again, none of the chosen domain families are classified under the same clan and inter-class similarity is therefore likely not very high. Within each class sequences are of very similar length, according to the standard deviations. The overall sequence length in *Pfam8_L200* has a very low standard deviation as well (200.8 ± 5.7).

DS	domain ID	clan ID	description	length in amino acids
⋈	PF01061	CL0181	ABC-2 ABC2_membrane ABC-2 type transporter	205.9±3.6
▷	PF01184	-	Gpr1_Fun34_YaaH GPR1/FUN34/yaaH family	200.7±5.6
⋈	PF02361	-	CbiQ Cobalt transport protein	199.9±6.0
▷	PF05013	CL0035	Peptidase_MH FGase N-formylglutamate amidohydrolase	206.4±2.6
⋈	PF06439	CL0004	Concanavalin 3keto-disac_hyd 3-keto-disaccharide hydrolase	200.7±5.4
⋈	PF06764	CL0172	Thioredoxin DUF1223 Protein of unknown function (DUF1223)	196.6±4.3
▷	PF07685	CL0014	Glutaminase_I GATase_3 CobB/CobQ-like glutamine amidotransferase domain	195.3±3.5
▷	PF20169	-	DUF6537 Family of unknown function (DUF6537)	200.7±3.0
◁	PF00450	CL0028	AB_hydrolase Peptidase_S10 Serine carboxypeptidase	402.6±4.8
◁	PF00464	CL0061	PLP_aminotran SHMT Serine hydroxymethyltransferase	396.9±4.9
◁	PF00999	CL0064	CPA_AT Na_H_Exchanger Sodium/hydrogen exchanger family	398.2±5.0
◁	PF02163	CL0126	Peptidase_MA Peptidase_M50 Peptidase family M50	399.6±5.1

Table 3.2: The *Pfam8_L200* and *Pfam8_Lhybrid* datasets. In the first column, dataset (DS), a "▷" is placed when the domain family belongs to *Pfam8_L200* a "◁" when it belongs to *Pfam8_Lhybrid* and a "⋈" if it belongs to both. All families contain 2^{10} sequences each, *i.e.* data are uniformly distributed among classes.

3.2 Evaluation

Evaluation methods such as accuracy, F1-Score or Matthews Correlation Coefficient (MCC) play an important role in judging the performance of machine learning methods. The aim, however, of the experimental investi-

gation in the following chapter is to ascertain the efficacy of applying the SRP in the context of sequences in molecular biology. Here, the machine learning model is not under examination but rather the feature generation. Nonetheless, the performance metric stays the same. To determine how well the SRP is channeling the information that is encoded in the sequences into the classification algorithm, *accuracy* is going to be employed as a metric. This happens under the assumption that the machine learning algorithm of choice learns to classify the featurized sequences as best as possible because the goal is to only judge the ability of the SRP to generate features, not to judge the effectiveness of the classifier.

Accuracy is the ratio of correct class predictions to the size of the whole dataset, as shown in the following equation (adapted from [scikit-learn developers, 2022] and [Margherita et al., 2020]):

$$Accuracy = \frac{\text{number of correct predictions}}{\text{total number of predictions}}. \tag{3.1}$$

One has to consider that if the dataset is imbalanced, accuracy as a performance measure can be misleading. That is because if a majority of data points are from one class, *e.g.* 90%, and the classifier does not learn at all and appoints classes to data points at random, the accuracy would nevertheless be at around 90%. This would signal a good performance although the machine learning model did not learn anything about patterns or relationships in the data, which is what one wants. In order to circumvent this fallacy without losing the ability to use this relatively straight-forward performance metric, one can check if the classes in the dataset are of about equal size. That is the reason why all classes are sized equally in the generated datasets (see section 3.1). This way, accuracy is an unbiased measure of how many correct predictions are made by the machine learning model.

To further reduce the bias of data, the model is cross-validated by randomly splitting the data 10-fold twice and taking the average of the resulting accuracies. To clarify, each run and subsequent evaluation will be repeated 20 times on different subsamples of data each time. The standard deviation of accuracies will also be shown in order to give a measure of robustness and confidence. While the machine learning model will be trained on the training datasets ($\approx 9/10$ of the data), only the accuracy of prediction on the validation dataset ($\approx 1/10$ of the data) is evaluated. This intends to reduce the risk that the model is overfit to the data, since its accuracy is judged on data that it has not been trained with.

Chapter 4

Experiments, results and discussion

This chapter will combine descriptions of experiments and their results. The results will then be discussed right away in order to inform the design of subsequent experiments. Default parameters will be assumed wherever possible for all software and algorithms unless it is specifically noted otherwise. This is due to the immense amount of places where parameter optimization is possible. Every part of the approach can be adjusted, from gap costs during alignment to the number of reference sequences during measurement. Adjustments have to be made one by one in order to be able to understand what each of the changes is doing. The most important and hopefully robust metric by which to judge the overall effect of each change is the performance of the classifier on the dataset. The performance heavily relies on the parametrization of the algorithm but the hope is that by starting with the simplest setup, strong biases can mostly be prevented. The first experiments apply the GLVQ classifier with only one prototype per class. Additionally, the following parameters are set:

- batch size is 64
- the monotonously increasing function f in equation 2.10 is the following sigmoid function: $f(x) = \frac{1}{1+e^{-\beta x}}$ with $\beta = 10$
- prototypes are initialized for each class in the class mean
- k-folds cross-validation with 10 random splits and 2 random repeats
- $2^8 = 256$ such runs with a different set of references each time
- BLOSUM62 is used as the default substitution matrix for both alignment algorithms unless specified otherwise

4.1 The initial attempt

The potential of the SRP in combination with a classifier is to be explored. As a first step, a simple experiment is conducted. As the classifier, Gaussian Naive Bayes algorithm (GNB) is used because it lacks parameters. The dataset is the unconstrained *Pfam8* dataset that was randomly sourced from the Pfam protein family database (see section 3.1). The dataset simulates a real world problem where sequences would likely come in all lengths, also within a single protein

family. Reference sequences are chosen randomly, one at a time. According to the SRP, similarity is measured between the reference and all other sequences. This is done with the score that is acquired by either of the two described sequence alignment algorithms, the NW and the SW. Then, with only this one feature as the input, the GNB algorithm is applied that tries to learn to classify the sequences on the basis of their similarity to the reference. The results may be observed in figure 4.1 in the form of a violin plot.

Violin plots show similar markers as box plots, *i.e.* the minimal, mean and maximal values of the data. Additionally, they show how the data is distributed as smoothed probability density curves. The areas under those curves are filled-out, wing-like and symmetrical. If a majority of the data is between the mean and the maximal value, then the wing will be wider there (as in figure 4.1 on the left. When little data is as low as the minimal value then this end of the violin will have a sharp tip (as in figure 4.1 on the right).

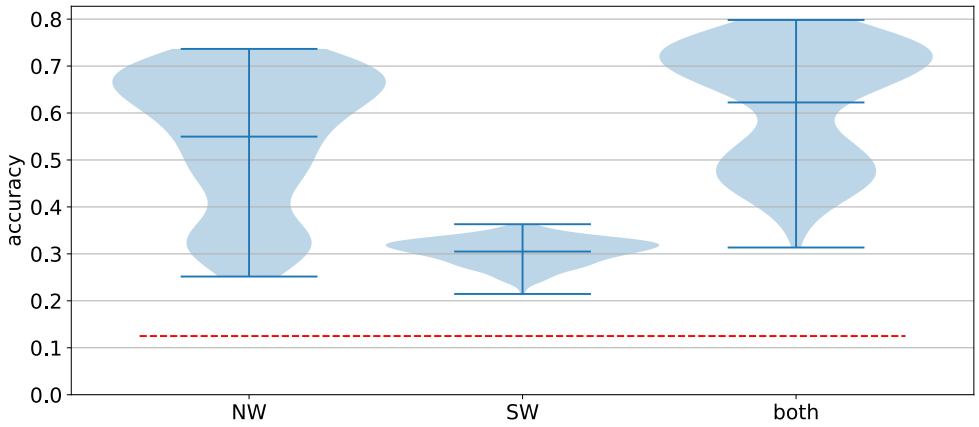


Figure 4.1: Violin plot of the accuracy of the GNB classifier for three different inputs. The inputs were the alignment scores from each the NW(1D) and the SW(1D) and both together (2D). $2^9 = 512$ runs with different references were conducted. The dashed red line signifies the random classification case, *i.e.* the worst possible performance for an 8-class problem.

Pfam8 is no benchmark dataset, *i.e.* information about the performance of other methods is missing. However, as a first proof of concept for feature generation from sequences, results look promising. The mean accuracy over all reference choices for NW alignment scores is $55.0\% \pm 14.4\%$. The maximal accuracy that was achieved for NW scores is even $73.7\% \pm 0.6\%$. This is high considering that the data set has 8 uniformly distributed classes. If there was no information in the generated 1D feature, the accuracy would only amount to $1/8 = 12.5\%$ which is signified by the dashed red line. The runs using the SW alignment score was not able to perform as well for classification. The mean accuracy was only at $30.5\% \pm 2.8\%$ and the maximal accuracy at $36.3\% \pm 1.7\%$. Although that is higher than 12.5%, it is not equally as reassuring. Taking both the global and the local alignment score as input leads to a slight improvement over using only NW with a mean accuracy of

62.3% $\pm$ 12.5%. The best references in this scenario lead to 79.8% $\pm$ 0.7% accuracy.

Evidently, information from sequences in molecular biology can be translated into a vectorial form that is accessible to a machine learning algorithm. Unfortunately, the standard deviation over all achieved accuracies is quite high, which raises doubts. Furthermore, the difference between the two alignment scores in terms of accuracy could be problematic if not understood. When both scores were used, some references seemed to produce a similarity measure that was distinctive enough to discriminate well between sequences from the 8 classes. Others yielded comparatively poor results. The mixture of apparent potential and uncertainty warrant additional research into some of the factors that might influence the SRP approach to sequences in molecular biology and motivate the following sections.

4.2 Is the length of the sequences significant for classification?

To start off, one question should be addressed right away. Both sequence alignment algorithms that have been mentioned in chapter 3 add up the rewards and costs of matches, mismatches and indels along the optimal sequence alignment. Given the parameters, such as substitution matrix and gap cost model, different alignments will be found optimal by the algorithm. The sum of the reward/cost values along the optimal path in the optimization matrix is what determines the score by which the similarity of the two sequences is judged. Two sequences are presumably more similar, the higher this score is.

The NW, as the first example, seeks the optimal global alignment. The amount of gaps that have to be introduced when trying to align a very short sequence with a very long sequence will likely have a consequential effect on the alignment score. Even if there are many matches and favored mismatches ($s(x_i, y_j) > 0$), the gaps will outweigh them. It is ultimately a question of whether the sequences in the dataset have similar lengths or not. If they do not, scores of alignments between very long and very short sequences will be very low. Accordingly, if the reference is a very long sequence, the classifier will only be able to discriminate well between other long sequences and fail to distinguish the short sequences. Is the reference a very short sequence, then it is the other way round because all long sequences will have a bad score and will not be distinguishable, other than by how big the difference in length is. Undoubtedly, the magnitude of the effect of that presumption depends strongly on the gap penalty model in connection with the substitution matrix.

The SW on the other hand does not, by design, yield negative scores. As soon as the impact of gaps and/or unfavoured mismatches gains the upper hand over the score, the algorithm will end the traceback of this path in the optimization matrix. Thus, depending on the gap penalty model, not many gaps and also unfavoured mismatches are allowed within the local alignment. To clarify, if there is a dominant domain in both a very long and a very short sequence, the score will be high nonetheless. Just as high, in fact, as if the sequences were of equal length.

To investigate, two datasets with length constraints were generated, as described in section 3.1: *Pfam8_L200* and *Pfam8_Lhybrid*. The mean accuracy, using different sequences as the reference, is visualized in figure 4.2.

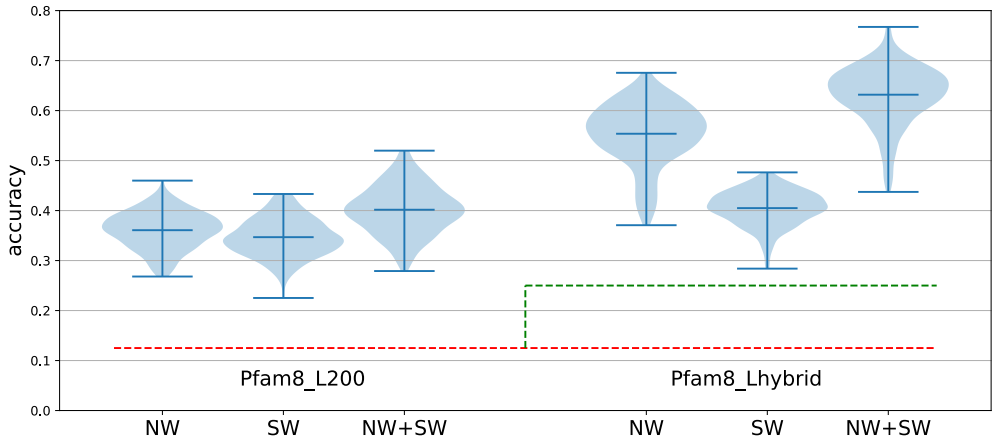


Figure 4.2: Violin plot of the average accuracies across datasets *Pfam8_L200* and *Pfam8_Lhybrid*. NW and SW were used as the similarity measures between sequence and reference. The dashed red line is at exactly $1/8 = 12.5\%$ which signifies the accuracy when the classifier might as well be guessing randomly. The dashed green line is at 25% for reasons described in the text. The horizontal lines of each violin plot indicate the minimal, mean and maximal values of each configuration.

Overall, NW performed better than SW. Both alignment methods performed worse on the *Pfam8_L200* dataset, where all sequences are of similar length, than on the *Pfam8_Lhybrid* dataset. With 8 uniformly distributed classes, no high accuracy is to be expected. To be precise, with no prior knowledge, a correct prediction has a probability of $P(\mathit{correct}) = 12.5\%$. However, among the random sample set of $2^8 = 256$ references, which is equivalent to $2^9/2^{13} = 3.125\%$ of the dataset, a maximal accuracy of $46.0\% \pm 1.0\%$ was achieved with NW. The mean accuracy was $36.1\% \pm 3.7\%$. Use of the SW score lead to a comparable mean accuracy of $34.1\% \pm 3.8\%$. Although that is approximately almost three times the probability that would be achieved by random assignment of classes, it is still quite low. Accuracy did not change much compared to only NW or SW when both scores where taken into account. The scores seemingly held more or less the same information about sequence similarity. The mean accuracy rose to $40.2\% \pm 4.8\%$. Some reference sequences delivered an alignment score that made it possible for the classifier to predict over half of the data in the validation dataset correctly. The maximal accuracy that was achieved was at $52.0\% \pm 1.4\%$.

The three violins on the right show the mean accuracy distributions over 256 different references for the *Pfam8_Lhybrid* dataset. The performance of the SRP-driven GLVQ model was much better than it was on the *Pfam8_L200* dataset. The NW score resulted in a mean accuracy of $55.4\% \pm 6.2\%$, a considerable 19.3% improvement over the *Pfam8_L200* dataset, and a remarkable maximal accuracy of $67.6\% \pm 0.8\%$. This means, that with only the NW alignment score as the 1D feature input, the model was able to predict more

than 2/3 of the validation data correctly. The mean accuracy when using the SW score was merely increased by 5.8% to 40.5% $\pm$ 3.5%. Even so, it is an improvement and at least a few references resulted in almost 50% correct predictions, at a maximum of 47.6% $\pm$ 1.7%. Despite the claim that the SW algorithm would produce the same score regardless of the length of each sequence, it performed better on the dataset with differently sized sequences. The claim only applies as long as SW aligns the same local domain.

It can not be said that the only difference between *Pfam8_L200* and *Pfam8_Lhybrid* is the length of its sequences, albeit the only intended difference. The fact of the matter is that it is impossible to gage the impacts of changing half of *Pfam8_L200*, besides the gain in information regarding the length of its data. This length increase in half of the data (4 classes with 2^{10} sequences each) is assumed to be the principal contrast between the two datasets. It is however also very possible that the increased performance on *Pfam8_Lhybrid* does not come from the sequences length but the interplay between the protein families. Perhaps the newly introduced sequences from protein families PF00450, PF00464, PF00999 and PF02163 are inherently easier to discriminate, also from the original sequences from *Pfam8_L200*. There is no easy way of canceling out this potential effect, even if expert knowledge of the specific protein families and their presumed inter-family similarity is available, because it is undoubtedly difficult to comprehend. One thing that could be done is to increase the sample size of protein families from the database and randomly pool them. This process would generate datasets with different compositions. While such a scheme was planned out, it has not yet been put into practice due to lack of time for implementation and running of the experiments.

It should be noted that the gap penalty model plays a large role during alignment and strongly influences the alignment score. However, all experiments in this work use a static gap penalty (the default for each of the substitution matrices) to reduce complexity.

Contrary to before on *Pfam8_L200*, accuracy was amplified when using both alignment scores together. The mean accuracy rose to 63.2% $\pm$ 5.8%. This is an increase of 7.8% over only using the NW score. The similarity scores to one reference in particular even enabled the resulting GLVQ models, to predict 76.7% $\pm$ 0.6% of the data correctly. This is remarkable given that the input only had 2 dimensions. The reference in that specific case is A0A095XZ51 from the protein family PF02361. Thus, it is present in both *Pfam8_L200* and *Pfam8_Lhybrid*.

Figure 4.3 shows what the SRP looks like for the 8 different protein families. Naturally, the highest scores can be found in the violin plot of the references own protein family. The overall maximum is the score of the alignment of the reference sequence with itself. The reference sequence generally has a score that is vastly greater than that of the majority of sequences because all amino acids are matching up in that alignment. It should be pointed out that many alignments with sequences from other protein families score higher than the ones with the sequences in its own protein family. This is because the membership in a protein family only requires the same protein domain to be present in all member proteins. Around those domains, the proteins typically look significantly different and, more importantly here, they have

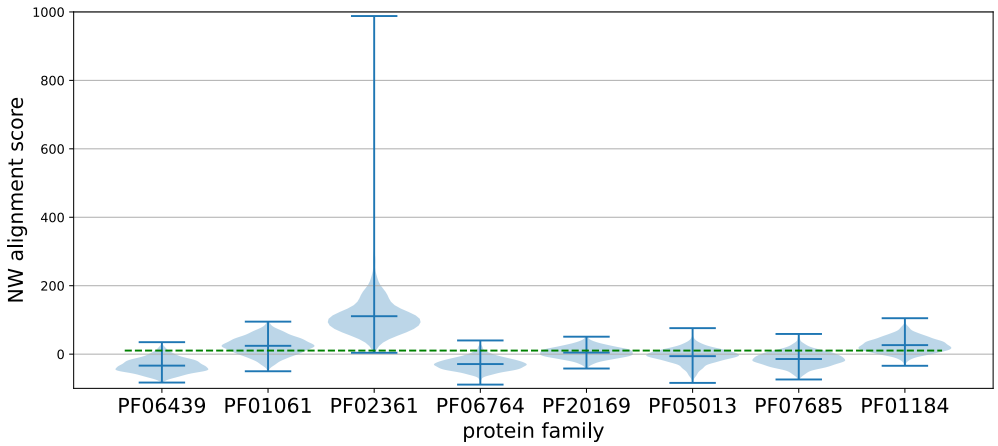


Figure 4.3: Violin plot of NW alignment scores over protein families in *Pfam8_L200*. Each violin visualizes the NW alignment scores of its sequences and the one particular reference (A0A095XZ51), which is from the PF02361 protein family. The dashed green line represents the global mean over all protein families.

different amino acids. The NW algorithm aligns the sequences globally and the many matches those mutual protein domains bring with them to increase the alignment score get easily outweighed by more numerous mismatches.

Yet, the NW alignment score for only one reference already looks like it could, for the most part, be used to separate one class from the others and apparently even more than that, judging by the maximal accuracy of $46.0\% \pm 1.0\%$. The same plot for the SW score looks partly similar (see figure 4.4). The highest score value belongs, again, to the alignment of the reference sequence with itself. It is equal to the NW alignment, because it exclusively consists of matches and BLOSUM62 was used as the substitution matrix in both algorithms.

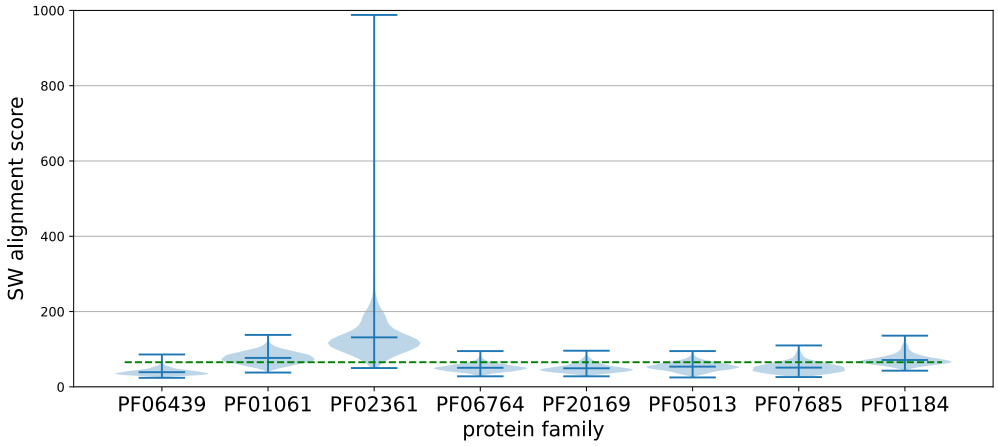


Figure 4.4: Violin plot of SW alignment scores over protein families in *Pfam8_L200*. Again, each violin visualizes the alignment scores, this time produced by the SW algorithm, class-wise over all sequences. The reference sequence belongs to PF02361 and is called A0A095XZ51. The dashed green line marks the global mean over all protein families.

What can be observed in both figures is that almost all of the alignment scores with protein sequences from the same protein family are greater than the overall mean. Over the entire *Pfam8_L200* dataset, this mean score value is at 10.5 ± 51.6 and 65.4 ± 35.2 for NW and SW respectively. The mean, however, for the alignment scores of alignments with sequences of the same protein family (PF02361) is 110.9 ± 57.6 and 131.5 ± 53.5 respectively. This is more than 2 standard deviations higher for both scores. This explains an observed intra-class accuracy that almost always exceeds 90%. It is the case for both alignment scores, that only 2 other protein families mean scores are above the overall mean. This likely adds to the discriminative power of the feature during classification. The violins that are representative of NW scores (figure 4.3) seem to be higher than the SW score violins, which corresponds to the standard deviations within each protein family.

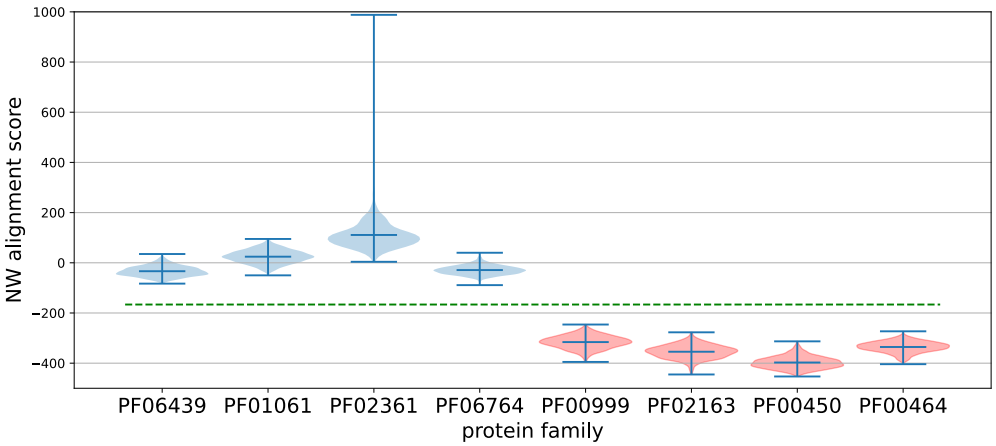


Figure 4.5: Violin plot of NW alignment scores over protein families in *Pfam8_Lhybrid*. The reference sequence, again, belongs to PF02361 and is called A0A095XZ51. The red violins are the ones belonging to the 4 newly substituted families. The dashed green line marks the global mean over all protein families.

The violin plot in figure 4.5 gives a strong hint as to why the performance with the NW score is better on the *Pfam8_Lhybrid* dataset. All alignments of the 207 amino acids long reference sequence and the sequences from the 4 classes with sequences, that are all about 400 amino acids long, are far below -200. That is because of all the gaps that need to be introduced in order to align sequences of such unequal length globally. They lower the scores substantially. The overall mean alignment score dropped to -166.3 ± 192.5. All alignment scores with the *short* sequences are well above that mean, while all alignments with *longer* sequences lie well below it. The standard deviation of intra-class scores did not change for the 4 original classes that are also in *Pfam8_L200*. The intra-class standard deviation of the 4 new classes is comparable. Nonetheless, overall standard deviation increased greatly, which explains why the discriminative power increased as well. This is not so much the case for the SW score, as can be interpreted from the violin plot in figure 4.6.

The alignment scores of the 4 new protein families are essentially not distributed differently to the ones that they are replacing. That is not surprising,

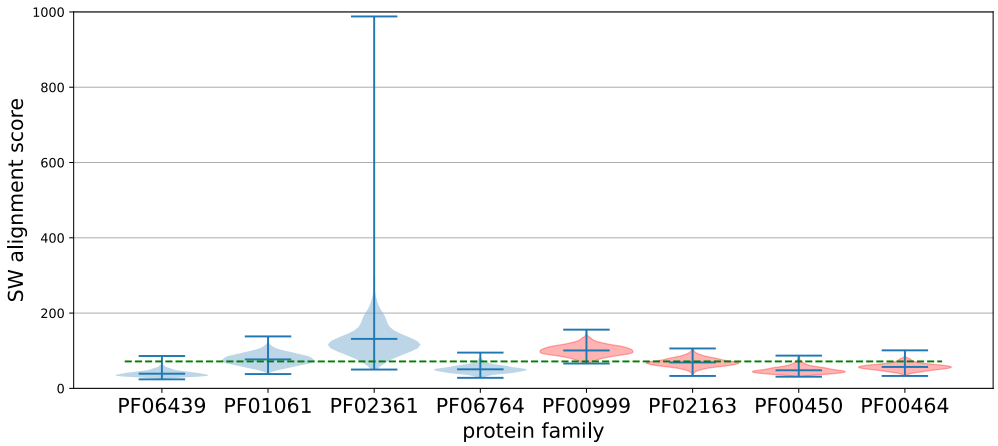


Figure 4.6: Violin plot of SW alignment scores over protein families in *Pfam8_Lhybrid*. The reference sequence, again, belongs to PF02361 and is called A0A095XZ51. The 4 violins depicting the new protein families are colored in red. The dashed green line marks the global mean over all protein families. The violins lie very similarly to the ones of the classes in *Pfam8_L200*.

because the reference sequence in question does not belong to any of them. Only PF00999 is noticeably above the mean unlike any other, besides PF02361 which contains the reference sequence A0A095XZ51. This might explain the slight increase of accuracy on the *Pfam8_Lhybrid* dataset.

That the protein families with the sequences of length of ≈ 400 are easier to discriminate when both available alignment scores are used becomes apparent in figure 4.7.

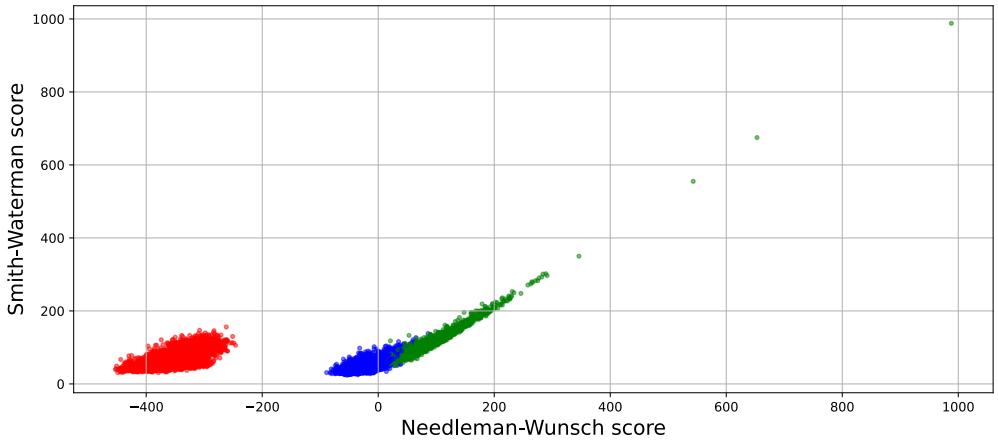


Figure 4.7: Scatter plot of NW and SW scores for the reference A0A095XZ51 from PF02361 (here in green). All alignments with the *longer* sequences are colored in red and the rest is colored in blue.

The red dots can be separated perfectly from the rest in the NW-dimension. This was evidently not possible in the SW-dimension. The original protein families from *Pfam8_L200* are also mostly separable from the references own family. Only very few green dots are in the far right corner. They represent the sequences that are most similar to the reference. Interestingly, but not

surprisingly, a clear linear relationship between the two dimensions is only perceptible in the green dots, *i.e.* the reference-lending class. This makes sense seeing that the alignments of the reference to most other sequences in both the data sets is likely nonsensical from a biological standpoint. High alignment scores are expected only within the own protein family. With NW, alignment scores to a random other sequence should be somewhere around zero, or negative if the sequences are of unequal length and many gaps are introduced. With SW, length discrepancy should not have such an effect. The observation matched the notion that SWs score is not correlated with the lengths of the sequences but only the lengths of the aligned domain.

And so it stands to reason that sequence length is the dominant discriminating factor for the score of the Needleman-Wunsch alignment algorithm, not merely because accuracy increased when differently sized sequences got introduced. In order to retain the NW score for further investigations, only *Pfam8_L200* will be regarded in the following sections. Because there is such a pronounced discriminative power in the alignment scores within the references class, the next point to be investigated is the choice of the reference sequence. The latest remarks focused on one single reference sequence as a showcase for this limited case. In the following sections, more than one reference sequence will be chosen in order to produce input features for the classifier.

4.3 Choice of references

The choice of substitution matrices could be brought forward as the next avenue to be investigated. Substitution matrices hold, however, biological knowledge and also bias by design. It might be that the performance that is induced through the use of a certain substitution matrix is tied too much to a specific case, *i.e.* the sequences of a specific protein family. The investigation of a multidimensional score profile will therefore be conducted first. Each data point in a dataset will be equipped with not only one alignment score, such as NW or SW, but will be aligned to multiple reference sequences in order to represent it better within the feature space. This way, such a bias of a substitution matrix towards certain sequences or whole protein families might be circumvented.

In an effort to facilitate reproducibility and reduce bias due to the choice of the references, a predefined set of references was generated from which all reference sequences are drawn. In it, all protein families are represented by an equally large number of sequences. The set is available on https://github.com/sicim/SRPMastersThesis.

As discussed, the following experiments are carried out on the *Pfam8_L200* dataset. To start off, the number of references was doubled. Two protein families were chosen randomly from which two sequences were picked as the references from the just-mentioned subset. Then the pairwise alignment scores were calculated with NW and SW. The scores were both used independently and together as the input to the GLVQ classifier. The dimensionality was, thus, $N = 2$ and $N = 4$ respectively. The process was repeated 256 times for distinctive reference subsets. The results of the experiments with only a single

reference from the previous section are displayed alongside the new results in figure 4.8.

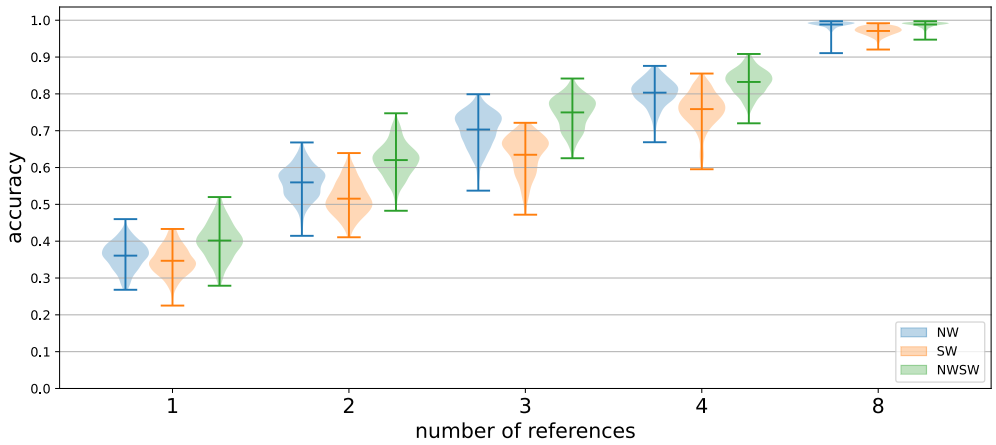


Figure 4.8: Accuracy depending on the number of references n. The violin plot shows the accuracy distributions for 256 differently sized sets of reference sequences each for $n = \{1, 2, 3, 4, 8\}$. The different inputs are colored differently.

The second set of violins, which shows the accuracy distributions over 256 sets of two references each, looks very similar to the first set, but shifted upwards. The shift amounts to +19.9%, +16.8% and +21.8% from 36.1%, 34.7% and 40.2% respectively, regarding the mean accuracy achieved with each input score. It may be remarked that virtually the entire violins experienced a shift upwards as not only the means increased, but also the minima (respective to each alignment method). It can be said therefore, that the second alignment score measurement from an extra reference sequence, that is from another protein family, consistently enhanced the performance as measured by the accuracy of the GLVQ classifier. The results are consistent with the concept of the SRP inasmuch as the extra measurement provided additional information about the sequence space and therefore helped the classifier to better classify the sequences. More importantly, the performance can be increased even more by using an increasing number of references. Using the same subset of predefined reference candidates for the random selection of references from the different protein families, the violins underwent an upward shift. The average accuracy means and accuracies with the best-performing and worst-performing references, as well as their standard deviation, may be observed in table 4.1. The last set of violins in figure 4.8 shows the accuracy distributions of the runs with 8 reference sequences, 1 from each of the protein families. The accuracy of the GLVQ classifier was nearly perfect at 98.8% on average for the NW score input, to which the SW score evidently did not add anything in their combined usage. The best-performing reference set with the SW score lead to an accuracy of 99.2% while those with the inputs including NW scores even accomplished an accuracy of 99.7%. This means that with only the alignment scores of 1 reference sequence from each family to all others, a nearly perfect representation of the sequence space was achieved. More importantly, the protein families are well-separable within the sensor response feature space.

# references	characteristic	NW	SW	NW+SW
1	maximum	46.0 ± 1.0	43.3 ± 0.9	52.0 ± 1.4
	mean	36.1 ± 3.7	34.7 ± 3.8	40.2 ± 4.8
	minimum	26.8 ± 1.2	22.5 ± 1.2	27.9 ± 1.6
2	maximum	66.8 ± 1.0	63.9 ± 1.0	74.7 ± 0.8
	mean	56.0 ± 4.4	51.5 ± 4.5	62.0 ± 4.6
	minimum	41.5 ± 1.5	41.1 ± 2.3	40.2 ± 4.8
3	maximum	79.9 ± 0.8	72.1 ± 0.7	84.2 ± 0.8
	mean	70.3 ± 4.7	63.5 ± 5.2	75.0 ± 4.3
	minimum	53.7 ± 1.2	47.2 ± 0.7	40.2 ± 4.8
4	maximum	87.6 ± 1.2	85.5 ± 0.9	90.8 ± 0.8
	mean	80.3 ± 3.6	75.9 ± 4.6	83.2 ± 3.3
	minimum	66.9 ± 0.9	59.5 ± 0.7	72.0 ± 1.5
8	maximum	99.7 ± 0.2	99.2 ± 0.2	99.7 ± 0.1
	mean	98.8 ± 0.8	97.1 ± 1.2	98.8 ± 0.7
	minimum	91.1 ± 0.9	92.0 ± 0.3	94.7 ± 0.4

Table 4.1: Table of accuracies using different number of references. Mean accuracies, minimal and maximal average accuracies, as well as their standard deviation, are listed in percentages for the use of n references with $n = \{1, 2, 3, 4, 8\}$.

It should be regarded that the SW score seems to provide a feature space that is not quite as suitable for separating the protein families as the NW is. Accuracies with the SW score were consistently worse for this dataset and this set of alignment parameters. The reasons for this can not be determined easily, however, a conjecture may be hazarded. Firstly, sequences in a Pfam protein family are all regions of proteins that contain the same domain. That makes them quite similar, but if the domain part is much shorter than the entire sequence ($\approx$ 200 amino acids for *Pfam8_L200*), which is likely, then it is not representative of the whole sequence. Secondly, both alignment algorithms are likely to get relatively high scores when aligning the sequences in one protein family, but a potentially substantial part of the sequences is probably quite different and weakens the meaningfulness of the scores. The scores can be observed for the example reference A0A095XZ51 in figures 4.3 and 4.4. While NW is forced to align the entire sequence, SW tries to only align the most similar and mostly contiguous regions, the domains. NW might therefore reward certain other regions that are present in many intra-class sequences that coincide with the domain of the class. Thirdly, the gap cost in combination with BLOSUM62 is set to -4. This is also the lowest value in BLOSUM62.

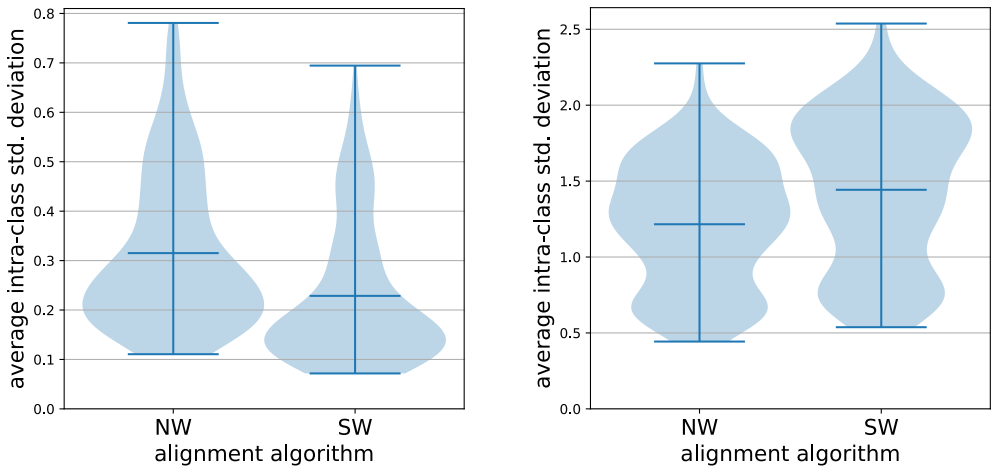


Figure 4.9: Standard deviation of NW and SW scores in *Pfam8_L200* dataset. Standard deviation of both alignment scores over all references in the predefined subset are shown. On the right is the standard deviation of reference-providing class and on the left is the standard deviation of the respective other 7 classes, *i.e.* the scores of the reference's class are excluded on the left. The scores were z-scaled beforehand.

Furthermore, the mean intra-family alignment scores for the A0A095XZ51 reference are only 110.9 ± 57.6 for NW and 131.5 ± 53.5 for SW. Considering the average length in this class of 199.9 ± 6.0, those are surprisingly low scores. Other random samples have shown, the same trend towards somewhat low scores. Finally, the fact that the protein families in Pfam are built with a much more advanced algorithm [Mistry et al., 2020] compared to NW and SW could mean that they are not sophisticated enough to pick up on the same signals for protein similarity that Pfams algorithm is. It might therefore be possible that the relatively high, but at the same time unexpectedly low intra-class alignment scores are not distinctive enough. NW might only produce more meaningful alignment scores than SW because it is biased less by the gap penalty, as it allows for the score to be negative without ending the alignment. In fact, intra-class standard deviation of the SW scores is systematically smaller than that of the NW scores for the classes that do not provide the reference. This may be observed in figure 4.9. Standard deviation of the SW scores is greater on average, however, within the class that provided the reference. The mean of the standard deviations of NW scores for the protein sequences that are not in the same class as the reference is 37.8% higher than that of SW scores. For the reference's class NW scores' standard deviation is only 15.7% less than the SW equivalent. Although this does not explain the reason for the difference in expressiveness between the two alignment scores, it does shed a light on possible quality measures for the scores that could be investigated further in the future.

Moving on, it is noticeable by looking at the violins in figure 4.8 and backed by closer inspection of the accuracies in table 4.1 that the intensity that the performance increases levels off as the number of references rises. This phenomenon indicates that the second measurement from another reference

carries some redundant information present already in the first. A third measurement has again some redundant information from the first two and so on. It is also the case that it was already apparent in 4.3 and 4.4 that a reference measurement does not only contain discriminative information about its own class, but also, to a small extent, about the data in the other classes. Those two observations support the notion that the sequence space is like a dark room that can be illuminated by placing light sources at different locations. One light source might not be enough to be able to make out all the objects in the room, maybe because the objects can conceal each other. Furthermore, two light sources that are placed right next to each other might not have a big effect on visibility.

The following experiment demonstrates that the location of the references in sequence space is important. Instead of randomly choosing 8 references from 8 different protein families they are all chosen from a single family or from only 2 families. The resulting performance of the GLVQ classifier on *Pfam8_L200* is visualized in figure 4.10. The exact values for mean, minimum and maximum accuracy are listed in table 4.2.

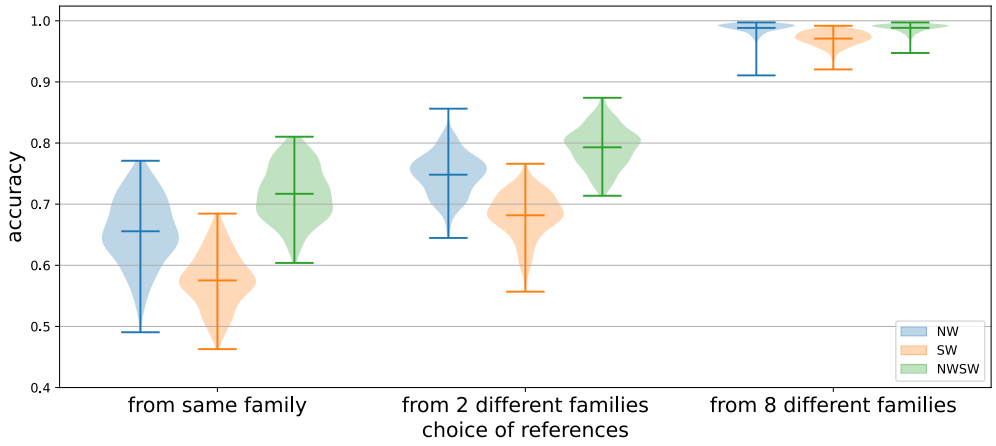


Figure 4.10: Violin plot of accuracies with references from 1, 2 or 8 different classes. Note that the accuracy axis begins at 0.4.

Accuracy is significantly worse when the references are all from the same protein family. Interestingly, though only similarity scores to sequences from one class were input for the classifier, the worst accuracies are just under 50% (49.0% $\pm$ 1.1% for NW and 46.3% $\pm$ 0.5% for SW). This is surprising, considering that the worst accuracy with just a single reference, which of course also represents only one class, was 26.8% $\pm$ 1.2% and 22.5% $\pm$ 1.2% respectively for the alignment algorithms. It can be concluded from this, that the sequences in the clusters in feature space are divers enough to increase the amount of information that is transferred from sequence space to feature space when more of them are used as references. This seems to be consistent over all classes because all were chosen the same amount of times to provide the references. Note that the best-performing set of references with both alignment algorithms was able to provide the GLVQ classifier with enough information to classify on average 81.0% $\pm$ 0.9% of sequences correctly. That is remarkable with such a limited selection of reference sequences.

# origins	characteristic	NW	SW	NW+SW
1	maximum	77.1 ± 1.3	68.5 ± 0.6	81.0 ± 0.9
	mean	65.6 ± 5.5	57.5 ± 4.5	71.7 ± 4.4
	minimum	49.0 ± 1.1	46.3 ± 0.5	60.4 ± 1.3
2	maximum	85.6 ± 0.4	76.6 ± 0.8	87.4 ± 0.8
	mean	74.8 ± 3.4	68.2 ± 3.7	79.3 ± 3.3
	minimum	64.5 ± 1.1	55.7 ± 1.0	71.4 ± 0.3
8	maximum	99.7 ± 0.2	99.2 ± 0.2	99.7 ± 0.1
	mean	98.8 ± 0.8	97.1 ± 1.2	98.8 ± 0.7
	minimum	91.1 ± 0.9	92.0 ± 0.3	94.7 ± 0.4

Table 4.2: Table of accuracies using sets of 8 references from 1, 2 or 8 different origins/protein families. Mean accuracies, minimal and maximal average accuracies, as well as their standard deviation, are listed in percentages.

Accuracy improved consistently with the refined choice of references from 2 of the 8 protein families. As before, the SW score seemed to either hold a little less information than the NW score, or the local alignments generate a worse feature space for the specific classification problem.

4.4 Choice of substitution matrices

As already mentioned at the beginning of the previous section, the influence of different substitution matrices is to be examined. The 5 default BLOSUMs in BLAST are selected, namely BLOSUM45, BLOSUM50, BLOSUM62, BLOSUM80 and BLOSUM90. They are attached in the appendix. Because of the already very high accuracy that was achieved with 8 references from all 8 protein families in *Pfam8_L200*, the decision was made to perform the following tests with only 4 references. The results are summarized in a violin plot in figure 4.11.

The averaged accuracies with the BLOSUM that has been used so far, BLOSUM62, are depicted in the third set of violins. The accuracies of the GLVQ classifier with alignment scores that used the alternative substitution matrices are mostly equivalent to that of the previous default substitution matrix. Two violins form visible exceptions, both using the SW algorithm, in combination with BLOSUM80 and BLOSUM90. Like previously, this is reflected in the standard deviations of the alignment scores with protein sequences that are not in the same class as the reference sequence. This is portrayed in figure 4.12. The described effect is more pronounced with both BLOSUM80 and BLOSUM90 compared to BLOSUM62. The mean of standard deviations of NW scores is 152.0% higher compared to the mean of the standard deviations of SW scores for BLOSUM80. For BLOSUM90 it even amounts to a 183.9% increase of NW over SW score standard deviation. This matches the systematically worse accuracies with those two specific substitution matrices.

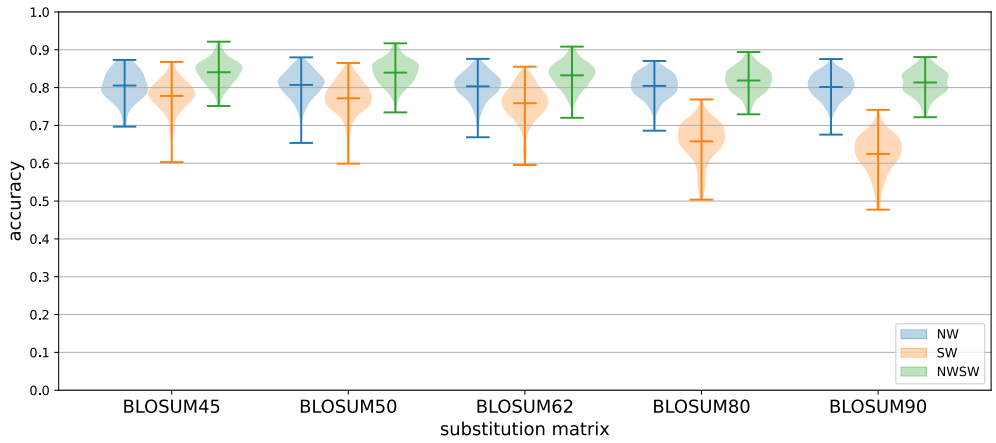


Figure 4.11: Violin plot of averaged accuracies with 5 different substitution matrices. As before, NW score and SW score were used alone and in combination.

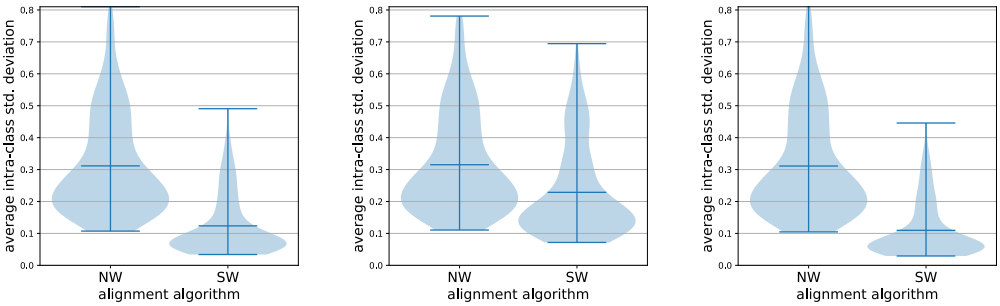


Figure 4.12: Standard deviation of NW and SW scores in *Pfam8_L200* dataset. Standard deviation of both alignment scores and with 3 different substitution matrices are shown over all references in the predefined subset. The alignment scores of all protein sequences that are member of the reference's class are excluded from the calculations. The right plot corresponds to BLOSUM80, the middle plot to BLOSUM62 and the left plot to BLOSUM90. The scores were z-scaled beforehand.

A biological interpretation is yet to be made, but the effect is likely connected to the *Pfam8_L200* dataset and its length condition.

Apart from the two exceptions, the choice of substitution matrix seems not to make a noticeable difference to the accuracy with the SRP approach.

4.5 Test with heterogeneous dataset

In this section, the *Pfam8* dataset is revisited in order to test the SRP on a dataset that is unrestricted regarding sequence lengths (see section 3.1). Thus, it will become clear whether to increase the number of references also has the same effect on an entirely different set of data. First, the experiment with 4 reference sequences from 4 different protein families was repeated for *Pfam8*. The same procedure was reiterated as before, *i.e.* NW and SW alignment scores to the references were used both independently and combined as the input to

the GLVQ classifier. Like in previous experiments, accuracies from all k folds were averaged for each set of references and the distribution of those averages was plotted as violins in figure 4.13. The mean, minimal and maximal accuracy values and their standard deviations are listed in table 4.3.

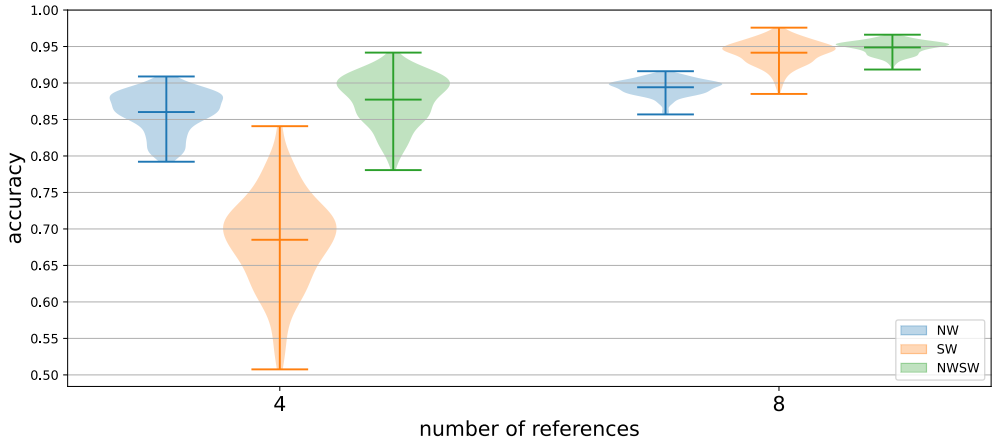


Figure 4.13: Violin plot of averaged accuracies on *Pfam8* with 4 and 8 references from different protein families. Like in previous experiments, NW score and SW score were used alone and in combination.

# references	characteristic	NW	SW	NW+SW
4	maximum	90.9 ± 0.7	84.1 ± 0.6	94.2 ± 0.3
	mean	86.0 ± 2.8	68.5 ± 6.2	87.7 ± 3.4
	minimum	79.2 ± 0.4	50.8 ± 0.6	78.1 ± 1.0
8	maximum	91.6 ± 0.2	97.6 ± 0.3	96.6 ± 0.3
	mean	89.4 ± 1.1	94.2 ± 1.7	94.9 ± 0.9
	minimum	85.7 ± 0.8	88.5 ± 0.6	91.8 ± 0.4

Table 4.3: Table of accuracies on *Pfam8* using different number of references. Mean accuracies, minimal and maximal average accuracies, as well as their standard deviation, are listed in percentages for the use of 4 and 8 references. Note that the accuracy axis begins at 0.5.

The result of the experiment with 4 references is that the accuracy with the NW score at a mean of $86.0\% \pm 2.8\%$ was comparable to the *Pfam8_L200* dataset, but that is not the case with the SW score, which performed worse with a mean accuracy of only $68.5\% \pm 6.2\%$. That may be comparable to the performance with SW on *Pfam8_L200*, but it is significantly worse than NW. It seems like local alignment did not produce meaningful alignment scores for some of the reference sets. The classifier's worst accuracy of $50.8\% \pm 0.6\%$ could be explained by alignments that were adequate only for the sequences in the same protein family as one of the references, but lacking expressiveness for the other half. This might be the case when the alignments are too short

because no sensible alignments are possible. The SW does not seem to add any information to the NW score in most cases, because the mean accuracies with NW and with NW+SW are almost the same and within each other standard deviation.

The runs with 8 references, 1 from each class, paint a different picture. The SW score outperformed NW significantly with a difference of 5.5%. The NW score, however, yielded a dependable accuracy of 89.4% $\pm$ 1.1%. And so did the combination of the two with a mean accuracy of 94.9% $\pm$ 0.9%, which marks the highest. It is however surprising to see that the SW score performed better for some reference sets than the combination of NW and SW, because the classifier should have performed at least equally as well if not better with both alignment scores as inputs than with only one. The maximal accuracies differ by 1.0%, which seems to be significant, judging by the standard deviations. The reason for that might be the classifier that was employed, GLVQ. The input with 2 scores for each of the 8 references was $2 \cdot 8 \rightarrow$ 16D. All of the 16 features span a 16D feature space in which the prototypes are updated, a number big enough to trigger what is called the *Curse of Dimensionality* [Keogh and Mueen, 2017]. For GLVQ as it is described in section 2.3, the input dimensions are weighed equally, as remarked also in [Bohnsack et al., 2022]. Eventually, there might be too much noise in the input features and GLVQ is not equipped to ignore it. GMLVQ on the other hand is able to learn the importance that each feature has for classification and may therefore circumvent the problem.

The same runs with the same sets of references were therefore repeated with the GMLVQ classifier and the results confirm the hypothesis. The averaged accuracies that were attained with GLVQ and GMLVQ with the same reference sets are displayed in the violin plot in figure 4.14.

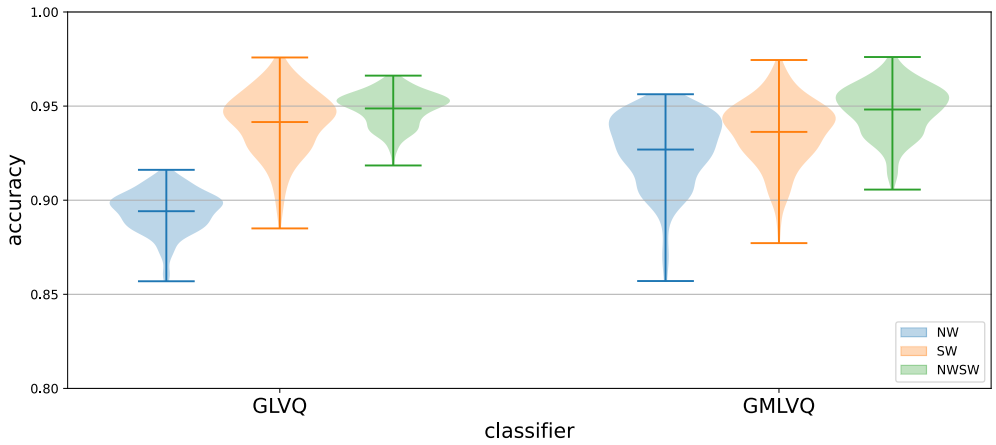


Figure 4.14: Violin plot of averaged accuracies with 8 references utilizing different classifiers. Resulting accuracy distributions are displayed of GLVQ on the left (same as in figure 4.13 on the right) and GMLVQ on the right. Like in previous experiments, NW score and SW score were used alone and in combination. The accuracy axis begins at 0.8.

Though the mean accuracy of the combined NW+SW scores remained unchanged, the maximal achieved accuracy went up to 97.6 $\pm$ 0.3, which is just slightly above the maximum of SW. That does not by itself exclude the

possibility of a run to output a better result for SW than for NW+SW, so the particular reference sets were compared against each other. It occurred a couple of times that the SW score lead to a higher accuracy than NW+SW but in those cases they were virtually the same, *i.e.* well within the respective standard deviation. It is also the case that NW scores resulted in a significantly better model accuracy with GMLVQ.

In conclusion, the SRP approach in combination with sequence similarity sensors is a successful method of feature generation for protein sequences. The approach has been tested on 3 datasets in total and on 1 dataset in particular. The specifics of the configuration of the sensors with different alignment algorithms and different substitution matrices, as well as the influence of the number of references were examined. The application of the GMLVQ classifier holds opportunities to further increase performance and seems to be the only way to allow for higher numbers of features. The possibility of interpretation and model improvement that is given by the CCM $\mathbf{\Lambda}$ appears promising in the context of the SRP and should be investigated further.

Chapter 5

Conclusions and future directions

There lies great potential in machine learning methods for bioinformatics, because the available amount of data grows significantly and needs to be analyzed and interpreted. Recent breakthroughs involving machine learning, *e.g.* AlphaFold [Jumper et al., 2021], are applying mostly deep learning methods. Those methods are not interpretable by themselves but at most explainable. Inherently interpretable approaches should be used instead [Rudin, 2019], *e.g.* variants of the prototype-based Learning Vector Quantization. Furthermore, existing methods like median or relational LVQ are unable to handle large amounts of data and are also much slower. A promising approach, the Sensor Response Principle, has therefore been adapted to protein sequences data.

Firstly, the necessary building blocks were described. Among them are two well-established but simple sequence alignment algorithms, NW and SW, that are used in combination with substitution matrices, first and foremost BLOSUMs. Together they yield a proximity measure for protein sequences, which is the input into for the classifier, GLVQ, which was introduced next. Finally, the SRP and its adaption to protein sequences was explained. Test datasets were generated from Pfam employing different conditions and the evaluation by means of accuracy metric discussed.

Having laid the foundation for the experiments, the methods were then put into practice with a naïve initial attempt with the SRP on the *Pfam8* dataset. This should be seen as a proof of concept, which revealed that it is in fact possible to classify protein sequences based on their sensor response, *i.e.* alignment score, to only a single reference sequence. The parameterless NW classifier achieved classification of k-fold validation datasets with a mean accuracy of around 55% with the NW score and over 60% with both alignment scores as input. That is respectable for a simple algorithm on an 8-class problem with low-dimensional input. The SRP approach was seemingly able to effectively translate information from protein sequences to a vectorial feature input of incredible sparsity.

The volatility of both sequence alignment algorithms regarding the diversity of sequence length was thereupon examined with GLVQ. Two different datasets were used: *Pfam8_L200* with sequences of around 200 amino acids each and *Pfam8_Lhybrid* that is a concatenation of half of *Pfam8_L200* and the other half being sequences of around 400 amino acids each. The accuracy differed significantly between the two datasets, which was at least partly accounted to the length discrepancy between the classes in *Pfam8_Lhybrid*, whose classes were seemingly easier to separate. This can also be a side effect introduced into *Pfam8_Lhybrid* by the 4 new classes that may be easier to discriminate perhaps

because of longer sequences. Closer examination of alignment scores to one example reference, however, suggested that at least NW is biased regarding length, simply because of the accumulation of gap costs. Global alignment means that there are at least as many gaps in the optimal alignment as the difference of the sequences' lengths. The NW score is, hence, very volatile when sequences have very different lengths. As a result, *Pfam8_L200* was chosen for all further experiments.

The sensor measurement was up to now restricted to a single reference at a time. Henceforth, the number of references was increased. First to 2, then to 3 and 4 and lastly to 8 references. As before, NW and SW scores were input on their own and also combined. The accuracies of the GLVQ classifier increased consistently with every additional reference. The rate, however, seemed to decrease, which suggests that with every additional reference there comes some partly redundant information. By taking one reference out of each protein family, the data seem to be represented very well in the subsequent feature space. Taking all things into consideration, it can at this point be argued that SRP is very well suited as a feature extraction method for protein sequences under the examined circumstances. With only 8 evenly spread references, accuracies of GLVQ consistently exceed 90% with a mean of $98.8\% \pm 0.8\%$ for the 8D feature vector based on the NW score. In a succeeding experiment, the condition that the references should come from different protein families was confirmed as necessary. The outstanding performance in classification is only possible if the references are chosen smartly, *e.g.* from distinct classes. Furthermore, it seemed as if the SW score was always suited slightly worse for the task than the NW score. This could only be traced back to a systematically smaller standard deviation across all tested references, at least for *Pfam8_L200*.

Alignments are undoubtedly influenced by substitution matrices, but according to experiments that were carried out, involving 5 of the most-used substitution matrices, this effect might almost be negligible for the SRP under the given circumstances. The 5 most common BLOSUMs were put to the test and did not result in varied accuracies. There were only 2 exceptions with SW in combination with BLOSUM80 and BLOSUM90, but for the rest the accuracies with identical sets of references (4 references from 4 protein families) remained very similar.

As a last step, *Pfam8* from the beginning was revisited, this time with GLVQ and more references (4 and 8). The SRP approach worked quite well, although the feature input via SW score seemed to need more references than SW, perhaps due to it being more fragile with regard to nonsensical inter-class alignments than the NW score. Intriguingly, the GLVQ classifier showed minor indications of the adverse effect of the *Curse of Dimensionality* [Keogh and Mueen, 2017]. Thus, GMLVQ was utilized and validated the hypothesis. Besides allowing a higher number of features, be it multiple references, alignment algorithms or substitution matrices, another advantage of GMLVQ over GLVQ is that it can provide information about the importance of certain features regarding the classification task. That means that it would be possible to learn a model with a bigger set of references, each with several combinations of alignment algorithms and substitution matrices. The CCM $\mathbf{\Lambda}$ could then be used to judge the importance of each of the feature dimensions and give an

insight about which combination works the best. Only those may then be necessary to compute in order to classify a new, unknown protein sequence. Further, it is conceivable that $\mathbf{\Lambda}$ could be interpreted in the context of a specific dataset/problem when certain features stand out. Finally, it might be possible to utilize $\mathbf{\Lambda}$ so as to generate new kinds of substitution matrices, which would be rooted in existing substitution matrices and informed by the classification task. A linear combination of the best-performing substitution matrices is theoretically possible (private communications with Prof. Villmann). This could possibly give rise to a family of substitution matrices that is motivated by classification tasks instead of protein sequence alignments of homologous proteins or protein similarity search, which have been the leading rationales so far.

Overall, this thesis provides a starting point for further research. It is necessary to test the adapted SRP approach on more datasets and to explore the opportunities that lie in the CCM.

Appendix

Poster

Prototype-based learning for sequences in molecular biology with the Sensor Response Principle

Julius Voigt, Marika Kaden, Thomas Villmann

Abstract

Sequences are an important data structure in molecular biology, but unfortunately it is difficult for most machine learning algorithms to handle them, as they rely on vectorial data. Recent approaches include methods that rely on proximity data, such as median and relational Learning Vector Quantization (LVQ). However, many of them are limited in the size of the data they are able to handle. A standard method to generate vectorial features for sequence data does not exist yet. Consequently, a way to make sequence data accessible to preferably interpretable machine learning algorithms needs to be found. Therefore a new approach called the Sensor Response Principle is adapted to protein sequences. Accordingly, sequence similarity is measured via pairwise sequence alignments with different sequence alignment algorithms and various substitution matrices. The measurements are then used as input for learning with the Generalized Learning Vector Quantization (GLVQ) algorithm. The impact of the number of references as well as the choice of substitution matrices is examined.

Schematic summary

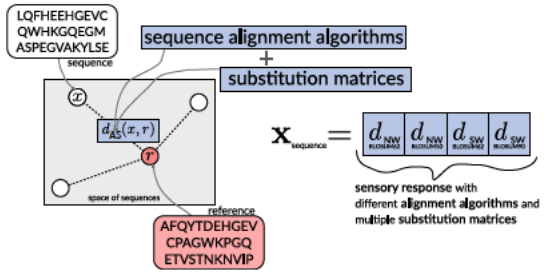
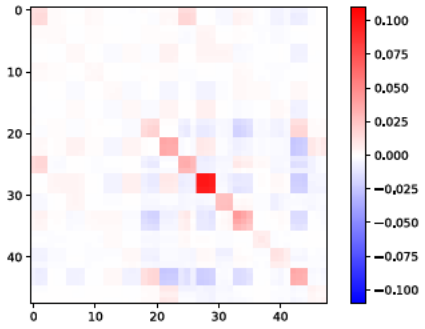


Figure : Schematic summary of the SRP for protein sequences. Space of sequences with a reference sequence r and another sequence x are shown. Choice of alignment algorithm and substitution matrix influence the sensor measurement $d_{AS}(x, r)$. In this example, NW, SW and 3 different BLOSUMs are used. Adapted from [Bohnsack et al., 2022].

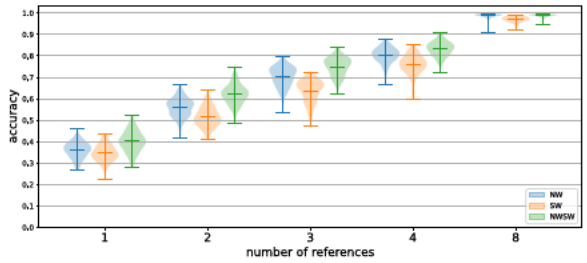
Classification Correlation Matrix

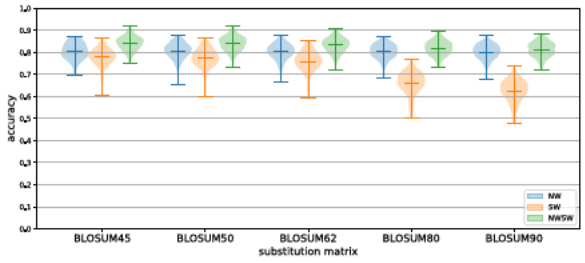


Conclusion

The Sensor Response Principle is a method to generate vectorial features from protein sequences by way of capturing their similarity to each other with pairwise sequence alignments. Only relatively few alignments need to be calculated, namely only the one to the references, which give this approach a major speed advantage. The more reference sequences are chosen the better, because the original data space is then translated better into feature space and the accuracy of any classifier will be better higher. One reference per protein family was enough to consistently achieve accuracies of over 90%. Substitution matrices do not have a big effect on performance.

The Classification Correlation Matrix is learned with the matrix variant of GLVQ, called GMLVQ. It can be interpreted in order to figure out the most important references, alignment algorithms or substitution matrices for the classification task.





Hochschule Mittweida
Technikumplatz 17, 09648 Mittweida
Julius Voigt
voigt5@hsmw.de
hs-mittweida.de

References

[Bohnsack et al., 2022] Bohnsack, K. S., Kaden, M., Voigt, J., and Villmann, T. (2022). Efficient classification learning of biochemical structured data by means of relevance weighting for sensoric response features. In ESANN 2022 proceedings. Ciaco - i6doc.com.

BLOSUMs

The BLOSUMs used in this work.

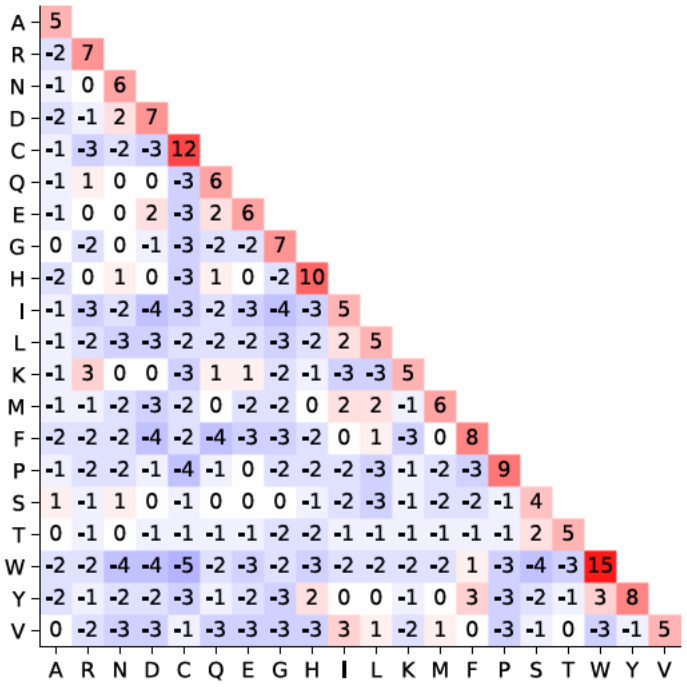


Figure 5.1: BLOSUM45 substitution matrix. Every item in the matrix stands for the score of an amino acid pair. Positive values are colored red, while negative values are colored blue. The letters stand for the amino acids as specified in [JCBN, 1984].

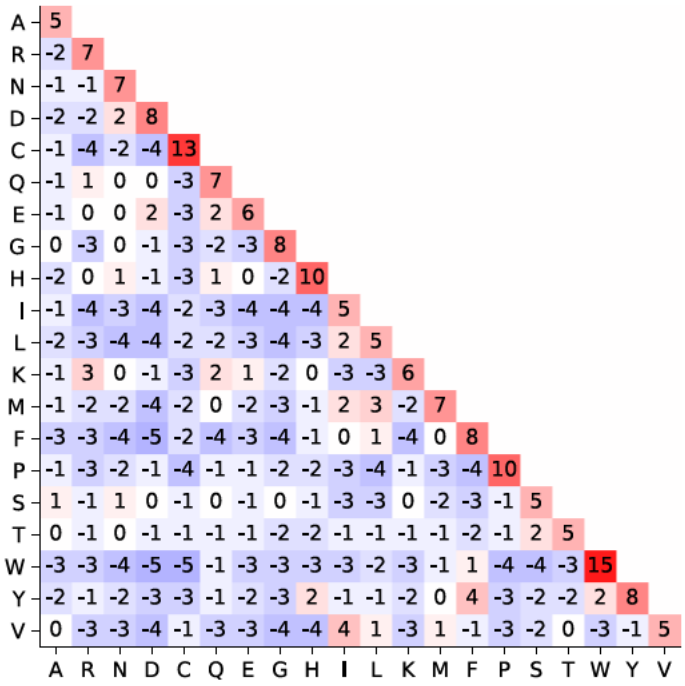


Figure 5.2: BLOSUM50 substitution matrix. Every item in the matrix stands for the score of an amino acid pair. Positive values are colored red, while negative values are colored blue. The letters stand for the amino acids as specified in [JCBN, 1984].

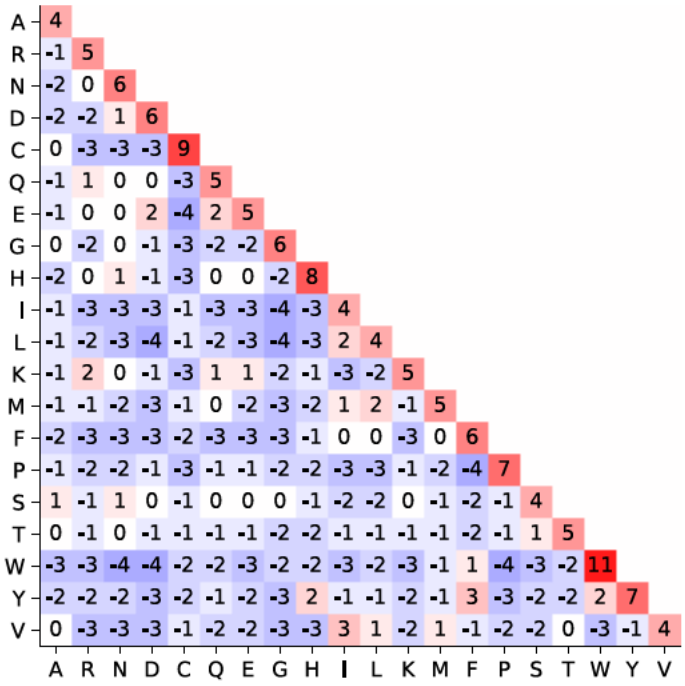


Figure 5.3: BLOSUM62 substitution matrix. Every item in the matrix stands for the score of an amino acid pair. Positive values are colored red, while negative values are colored blue. The letters stand for the amino acids as specified in [JCBN, 1984].

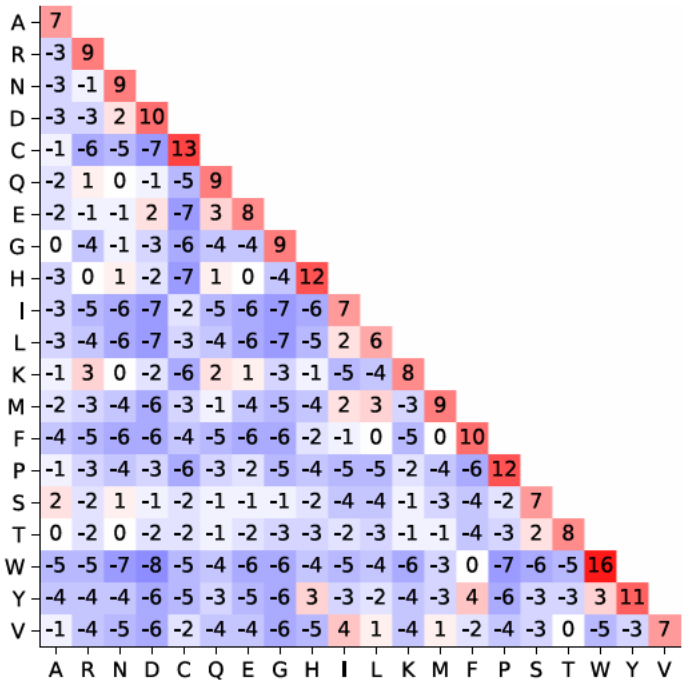


Figure 5.4: BLOSUM80 substitution matrix. Every item in the matrix stands for the score of an amino acid pair. Positive values are colored red, while negative values are colored blue. The letters stand for the amino acids as specified in [JCBN, 1984].

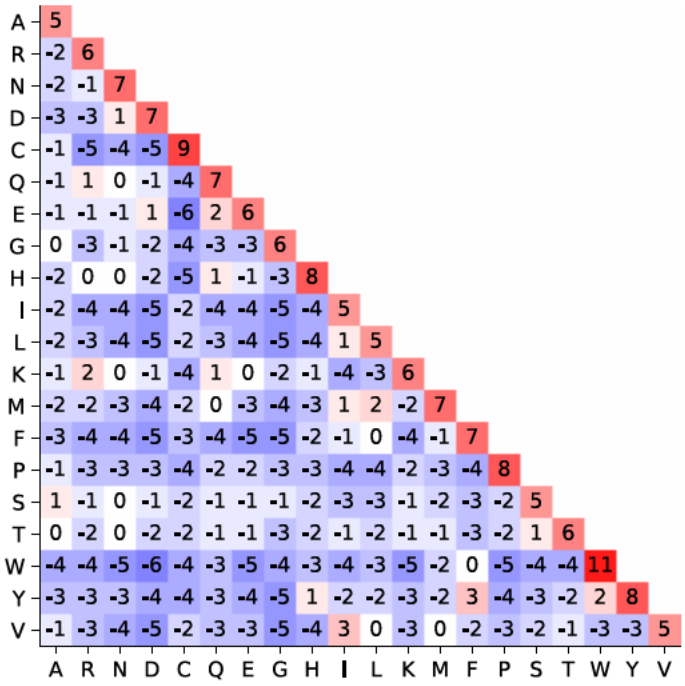


Figure 5.5: BLOSUM90 substitution matrix. Every item in the matrix stands for the score of an amino acid pair. Positive values are colored red, while negative values are colored blue. The letters stand for the amino acids as specified in [JCBN, 1984].

Bibliography

[Ahmedt-Aristizabal et al., 2021] Ahmedt-Aristizabal, D., Armin, M. A., Denman, S., Fookes, C., and Petersson, L. (2021). Graph-based deep learning for medical diagnosis and analysis: Past, present and future. *Sensors (Basel, Switzerland)*, 21.

[Altschul, 1991] Altschul, S. F. (1991). Amino acid substitution matrices from an information theoretic perspective. *Journal of Molecular Biology*, 219:555 – 565.

[Altschul et al., 1990] Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410.

[Angra and Ahuja, 2017] Angra, S. and Ahuja, S. (2017). Machine learning and its applications: A review. In *2017 International Conference on Big Data Analytics and Computational Intelligence (ICBDAC)*. IEEE.

[Berman, 2000] Berman, H. M. (2000). The protein data bank. *Nucleic Acids Res.*, 28(1):235–242.

[Biehl et al., 2016] Biehl, M., Hammer, B., and Villmann, T. (2016). Prototype-based models in machine learning. *Wiley Interdisciplinary Rev.s: Cognitive Sci.*, 7(2):92–111.

[Blaisdell, 1989] Blaisdell, B. E. (1989). Average values of a dissimilarity measure not requiring sequence alignment are twice the averages of conventional mismatch counts requiring sequence alignment for a computer-generated model system. *J. Molecular Evolution*, 29(6):538–547.

[Blum et al., 2020] Blum, M., Chang, H., Chuguransky, S., Grego, T., Kandasaamy, S., Mitchell, A., Nuka, G., Paysan-Lafosse, T., Qureshi, M., Raj, S., Richardson, L., Salazar, G. A., Williams, L., Bork, P., Bridge, A., Gough, J., Haft, D. H., Letunic, I., Marchler-Bauer, A., Mi, H., Natale, D. A., Necci, M., Orengo, C. A., Pandurangan, A. P., Rivoire, C., Sigrist, C. J. A., Sillitoe, I., Thanki, N., Thomas, P. D., Tosatto, S. C. E., Wu, C. H., Bateman, A., and Finn, R. D. (2020). The InterPro protein families and domains database: 20 years on. *Nucleic Acids Res.*, 49(D1):D344–D354.

[Bohnsack, 2020] Bohnsack, K. S. (2020). How to compare RNA/DNA sequences : a systematic approach in terms of data transformations and proximity measures = Der Vergleich von RNA/DNA-Sequenzen : ein systematischer Ansatz hinsichtlich Datentransformationen und (Un-) Ähnlichkeitsmaßen. Master's thesis, University of Applied Sciences Mittweida.

[Bohnsack et al., 2021] Bohnsack, K. S., Kaden, M., Abel, J., Saralajew, S., and Villmann, T. (2021). The resolved mutual information function as a structural fingerprint of biomolecular sequences for interpretable machine learning classifiers. *Entropy*, 23(10):1357.

[Bohnsack et al., 2022] Bohnsack, K. S., Kaden, M., Voigt, J., and Villmann, T. (2022). Efficient classification learning of biochemical structured data by means of relevance weighting for sensoric response features. In *ESANN 2022 proceedings*. Ciaco - i6doc.com.

[Bunte et al., 2012] Bunte, K., Schneider, P., Hammer, B., Schleif, F., Villmann, T., and Biehl, M. (2012). Limited rank matrix learning, discriminative dimension reduction and visualization. *Neural Networks*, 26:159–173.

[Crick, 1970] Crick, F. H. C. (1970). Central dogma of molecular biology. *Nature*, 227:561–563.

[Hammer et al., 2014] Hammer, B., Hofmann, D., Schleif, F., and Zhu, X. (2014). Learning vector quantization for (dis-)similarities. *Neurocomputing*, 131:43–51.

[Hanson and Coller, 2017] Hanson, G. and Coller, J. (2017). Codon optimality, bias and usage in translation and mRNA decay. *Nature Rev.s Molecular Cell Biology*, 19(1):20–30.

[Henikoff and Henikoff, 1991] Henikoff, S. and Henikoff, J. G. (1991). Automated assembly of protein blocks for database searching. *Nucleic Acids Res.*, 19(23):6565–6572.

[Henikoff and Henikoff, 1992] Henikoff, S. and Henikoff, J. G. (1992). Amino acid substitution matrices from protein blocks. *Proc. the Nat. Acad. of Sci.*, 89(22):10915–10919.

[Hess et al., 2016] Hess, M., Keul, F., Goesele, M., and Hamacher, K. (2016). Addressing inaccuracies in BLOSUM computation improves homology search performance. *BMC Bioinformatics*, 17(1).

[Jain and Schultz, 2018] Jain, B. J. and Schultz, D. (2018). Asymmetric learning vector quantization for efficient nearest neighbor classification in dynamic time warping spaces. *Pattern Recognition*, 76:349–366.

[JCBN, 1984] JCBN (1984). IUPAC-IUB Joint Commission on Biochemical Nomenclature (JCBN): Nomenclature and symbolism for amino acids and peptides. recommendations 1983. *European J. Biochemistry*, 138(1):9–37.

[Jumper et al., 2021] Jumper, J. M., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Zídek, A., Potapenko, A., Bridgland, A., Meyer, C., Kohl, S. A. A., Ballard, A., Cowie, A., Romera-Paredes, B., Nikolov, S., Jain, R., Adler, J., Back, T., Petersen, S., Reiman, D. A., Clancy, E., Zielinski, M., Steinegger, M., Pacholska, M., Berghammer, T., Bodenstein, S., Silver, D., Vinyals, O., Senior, A. W., Kavukcuoglu, K., Kohli, P., and Hassabis, D. (2021). Highly accurate protein structure prediction with alphafold. *Nature*, 596:583 – 589.

[Keogh and Mueen, 2017] Keogh, E. and Mueen, A. (2017). Curse of dimensionality. In *Encyclopedia of Machine Learning and Data Mining*, pages 314–315. Springer US.

[Keul et al., 2017] Keul, F., Hess, M., Goesele, M., and Hamacher, K. (2017). PFASUM: a substitution matrix from pfam structural alignments. *BMC Bioinformatics*, 18(1).

[Kohonen, 1986] Kohonen, T. (1986). *Learning Vector Quantization for Pattern Recognition*. Report TKK-F-A. Helsinki University of Technology.

[Margherita et al., 2020] Margherita, G., Enrico, B., and Giorgio, V. (2020). Metrics for multi-class classification: an overview.

[Mistry et al., 2020] Mistry, J., Chuguransky, S., Williams, L., Qureshi, M., Salazar, G., Sonnhammer, E. L. L., Tosatto, S. C. E., Paladin, L., Raj, S., Richardson, L. J., Finn, R. D., and Bateman, A. (2020). Pfam: The protein families database in 2021. *Nucleic Acids Research*, 49(D1):D412–D419.

[Nebel et al., 2015] Nebel, D., Hammer, B., Frohberg, K., and Villmann, T. (2015). Median variants of learning vector quantization for learning of dissimilarity data. *Neurocomputing*, 169:295–305. Learning for Visual Semantic Understanding in Big Data ESANN 2014 Industrial Data Processing and Analysis.

[Needleman and Wunsch, 1969] Needleman, S. B. and Wunsch, C. D. (1969). A general method applicable to the search for similarities in the amino acid sequence of two proteins. In *Molecular Biology*, pages 453–463. Elsevier.

[Pearson and Lipman, 1988] Pearson, W. R. and Lipman, D. J. (1988). Improved tools for biological sequence comparison. *Proceedings of the National Academy of Sciences*, 85(8):2444–2448.

[Pedregosa et al., 2011] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Édouard Duchesnay (2011). Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12(85):2825–2830.

[Punta et al., 2011] Punta, M., Coggill, P. C., Eberhardt, R. Y., Mistry, J., Tate, J., Boursnell, C., Pang, N., Forslund, K., Ceric, G., Clements, J., Heger, A., Holm, L., Sonnhammer, E. L. L., Eddy, S. R., Bateman, A., and Finn, R. D. (2011). The pfam protein families database. *Nucleic Acids Res.*, 40(D1):D290–D301.

[Ravichandran, 2020] Ravichandran, J. (2020). Prototorch. `https://github.com/si-cim/prototorch`.

[Rudin, 2019] Rudin, C. (2019). Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence*, 1(5):206–215.

[Saralajew et al., 2019] Saralajew, S., Holdijk, L., Rees, M., and Villmann, T. (2019). Robustness of generalized learning vector quantization models against adversarial attacks. In *Advances in Intelligent Systems and Computing*, pages 189–199. Springer International Publishing.

[Sato and Yamada, 1995] Sato, A. and Yamada, K. (1995). Generalized learning vector quantization. In *NIPS*.

[Schneider et al., 2009] Schneider, P., Biehl, M., and Hammer, B. (2009). Adaptive relevance matrices in learning vector quantization. *Neural Computation*, 21(12):3532–3561.

[scikit-learn developers, 2022] scikit-learn developers (2022). Model selection and evaluation. `https://scikit-learn.org/stable/modules/model_evaluation.html`.

[Smith and Waterman, 1981] Smith, T. and Waterman, M. (1981). Identification of common molecular subsequences. *J. Molecular Biology*, 147(1):195–197.

[Trivedi and Nagarajaram, 2020] Trivedi, R. and Nagarajaram, H. A. (2020). Substitution scoring matrices for proteins - an overview. *Protein Sci.*, 29(11):2150–2163.

[UniProt consortium, 2022] UniProt consortium (2022). Uniprot release 2022_04. `https://ftp.uniprot.org/pub/databases/uniprot/relnotes.txt`.

[Villmann et al., 2016] Villmann, T., Bohnsack, A., and Kaden, M. (2016). Can learning vector quantization be an alternative to SVM and deep learning? - recent trends and advanced variants of learning vector quantization for classification learning. *J. Artificial Intelligence Soft Computing Res.*, 7(1):65–81.

[Wang et al., 2019] Wang, Y., Tian, K., and Yau, S. S.-T. (2019). Protein sequence classification using natural vector and convex hull method. *J. Computational Biology*, 26(4):315–321.

[Yoo et al., 2014] Yoo, Y.-H., You, Y., Jang, I., Lee, K. J., Kim, H. J., and Lee, K. H. (2014). An approach for a substitution matrix based on protein blocks and physicochemical properties of amino acids through pca. *Interdisciplinary Bio Central*, 6:3.

[Zitnik et al., 2019] Zitnik, M., Nguyen, F., Wang, B., Leskovec, J., Goldenberg, A., and Hoffman, M. M. (2019). Machine learning for integrating data in biology and medicine: Principles, practice, and opportunities. *Information Fusion*, 50:71–91.

[Zoghlami et al., 2021] Zoghlami, F., Kaden, M., Villmann, T., Schneider, G., and Heinrich, H. (2021). AI-based multi sensor fusion for smart decision making: A bi-functional system for single sensor evaluation in a classification task. *Sensors*, 21(13):4405.

[Zvelebil and Baum, 2007] Zvelebil, M. J. and Baum, J. O. (2007). *Understanding Bioinformatics*. CRC Press, Boca Raton, FL.

Glossary

BLOSUM__ BLOSUM where the sequences that are clustered have a minimal sequence identity of the given number in %

Pfam Database of protein families and domains, now part of InterPro

Pfam8 dataset with $2^3 \cdot 2^{10} = 2^{13}$ protein sequences from 2^3 domain families

Pfam8_L200 like *Pfam8*, but all sequences are approximately 200 long

Pfam8_Lhybrid like *Pfam8*, but one half of the sequences are approximately 200 long while the other half is approximately 400 long

z-scaling Center to the mean and divide by standard deviation, thereby normalising the data: $z = \frac{x-\mu}{\sigma}$ with the mean μ and standard deviation σ

Declaration

I declare that this master's thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where states otherwise by reference or acknowledgment, the work presented is entirely my own.

January 3, 2023