CHALMERS
UNIVERSITY OF TECHNOLOGY

UNIVERSITY OF GOTHENBURG

Machine learning to predict enzymes' optimal catalytic temperature

Master's thesis in Computer science and engineering

JOSEFIN ULFENBORG

Department of Computer Science and Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
UNIVERSITY OF GOTHENBURG
Gothenburg, Sweden 2020

Master's thesis 2020

Machine learning to predict enzymes' optimal catalytic temperature

JOSEFIN ULFENBORG

UNIVERSITY OF
GOTHENBURG

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2020

Machine learning to predict enzymes' optimal catalytic temperature
JOSEFIN ULFENBORG

Supervisor: Martin Engqvist, Department of Biology and Biological Engineering
Supervisor: Graham Kemp, Department of Computer Science and Engineering
Examiner: Peter Damaschke, Department of Computer Science and Engineering

Master's Thesis 2020
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2020

Machine learning to predict enzymes' optimal catalytic temperature
JOSEFIN ULFENBORG
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Enzymes are proteins which operate as biological catalysts in chemical processes, for instance in biofuel production. The efficiency and sustainability of these processes may be greatly improved by knowing the optimal catalytic temperature (T_{opt}) of the enzymes. However, determining these temperatures experimentally is time-consuming and instead a machine learning approach for predicting T_{opt} is suggested. In a previous approach, sequential features were used to predict T_{opt}. In this thesis, new structural features which account for various structural properties in the enzymes were used alongside the sequential features. Test scores from the models show that structural features combined with sequential features improve previous R^2 scores from 0.4 to 0.48. Furthermore, in the case where there is a pair of similar enzymes, but one has a colder and one a hotter temperature, the models correctly predicts the temperature order of the enzymes 83% of the time. By gathering more data and fine-tuning the structural features, it is anticipated that scores will improve even further.

Keywords: Structural bioinformatics, enzymes, machine learning, feature engineering.

Acknowledgements

I would like to express my sincerest gratitude to both my supervisors Graham Kemp and Martin Engqvist. Their guidance and unwavering support throughout the entire duration of this thesis have been invaluable and highly appreciated. I am grateful for all programs and papers Graham has offered to aid in my research, and to Martin for assisting in data processing providing the dataset for my disposal.

Sincerely,

Josefin Ulfenborg, Gothenburg, June 2020

Contents

List of Figures

List of Tables

1

Introduction

This project covers the study of feature engineering and applying machine learning methods to predict enzymes' optimal catalytic temperature. Enzymes are present in organisms, for instance in bacteria or humans, and all of these organisms have a growth temperature, also called the *organism's growth temperature* (OGT). Moreover, enzymes are used as catalysts in biochemical reactions and when they are, a specific process temperature is used to increase the reaction rate, however, this is not necessarily the temperature when the enzymes are the most effective. Instead, it is necessary to develop tools to be able to predict their *optimal catalytic temperature* (T_{opt}), to increase the process temperature, and thereby the reaction rate even further.

1.1 Background

Every day we rely more and more on effective and renewable biomass production. Biomass is used in both heat and electricity production, but it can also be converted into liquid fuel which is used to power vehicles. However, it is apparent that we are not using the biomass energy to its full potential [33]. In biomass production there are several chemical reactions involved. These reactions need catalysts to power them, and one biological catalyst which can be used is enzymes. Enzymes are usually proteins but they also operate as biological catalysts and exist in all living organisms. As biological catalysts they can be used, for instance, in chemical reactions such as biofuel production [47, 49].

In order to increase the efficiency, sustainability and environmental friendliness of these reactions, and in turn be able to consume more biomass, it is necessary to know at which temperature these enzymes are the most effective [18, 2]. Furthermore, different enzymes will be particularly useful for different applications [3]. Thus, one important factor to be considered when selecting or designing an enzyme is the temperature range at which the enzyme is most effective, but determining this experimentally is time-consuming and expensive.

Machine learning can be used on relevant features extracted from enzymes to build models to predict their optimal catalytic temperature. Finding these optimal values means increasing the effectiveness of the chemical reactions. However, enzymes are complex molecules where their properties are a result of both their amino acid

sequences and their three dimensional structure. The main challenge in this project will thus be to interpret and extract meaningful features from enzymes, which can be used to build more accurate machine learning models.

1.2 Previous work

Previous work to estimate enzymes' T_{opt}, by Engqvist *et al.*, used the OGT and amino acid residue frequencies of the enzyme as features in machine learning models [26]. However, the OGT is not always available in a dataset, so without this feature the prediction scores of the model were lower. Moreover, even in the cases where OGT is available, it has been found that the T_{opt} of an enzyme is not always true to the OGT. That is, the optimal catalytic temperature might be lower or higher [13].

The models they found to be the most successful were support vector regression (SVR) and random forest. Furthermore, they used an R^2 score to determine the performance of the model. The R^2 score is the percentage of variation explained by the model, that is, how well a model fits the data (1 being a perfect fit). With both the OGT and amino acid residue frequencies as features, they got a score of 0.5 and this translates to that the relationship between the two features and T_{opt} accounts for 50% of the variation. In turn, this means that something else must explain the remaining 50%. Additionally, when only the amino acid residue frequencies were used as a feature, they received a score of 0.4.

They also used the sequence amino acid residues as a basis to generate hundreds of secondary features to train the models with. Additionally a convolutional neural network was used to train on the data. However, neither of these attempts resulted in a higher score than previously achieved [26].

1.3 Purpose

The aim of this project is to hand-craft and interpret structural features of enzymes and to use these to develop machine learning models that can predict T_{opt}. Following this, the features will be further analyzed to examine which ones are the most relevant and provide additional information about T_{opt}, and which features that do not contribute.

The previous score, obtained by Engqvist *et al.*, was 0.4 and this will be used as a baseline to examine if the new, structural features explain T_{opt} [26]. Even though they received a higher score with OGT, it will not be included as OGT and T_{opt} do not always correspond and since OGT is not always available. Instead, one of the objectives with this project is to study how well T_{opt} can be predicted without OGT.

If the structural features do not improve the score above the baseline, this will not be considered a failure. Even with lower scores, there will be highly valuable information encoded in the features, and to locate those that are the most relevant biologically is also considered criterion for success.

1.4 Limitations

Identifying and capturing the relevant features from enzyme structures are two of the main goals in this project. If machine learning models on the new structural features produce better results than previously obtained, it could be argued that using neural networks improve the results further. However, neural networks are beyond the scope of this thesis. Furthermore, the final models should not be considered production-ready. The results concern biological aspects and locating the features that give the most information about T_{opt}.

1.5 Ethical considerations

One common energy resource today is fossil fuel. By replacing it and instead using biofuel, it would have as an effect that CO_2-emissions would vastly decrease, as the source of biofuel, for instance plants, recycle the CO_2 from the atmosphere [3]. Nevertheless, it is not yet possible to fully replace fossil fuels with biofuels, due to the time, energy and costs required [17].

Chemical reactions are dependent on the temperature, and with accurate predictions of T_{opt}, the time and energy needed for, for instance biofuel production, would decrease, making it more sustainable and effective. However, another issue that needs to be addressed is where the source for biofuels should be extracted from. Thus, the ethical considerations that need to be taking into account when producing biofuels in the future are how to sustainably produce the source, to not lead to deforestation or compete for space with other food productions [39].

1.6 Outline of thesis

The outline of this thesis is as follows. Chapter 2 covers the background information of enzymes and machine learning needed to understand the methods used in this thesis. Chapter 3 lists which datasets were used and which pre-processing steps were applied to the data. Chapter 4 covers the feature extraction and feature calculations and Chapter 5 presents the results from training machine learning models using the features along with a discussion of the results. Chapter 6 includes a section about future work and Chapter 7 concludes the report.

2

Background

This chapter explains some of the theory and background information concerning enzymes and machine learning. The first section explains what enzymes are, how they are constructed and how different properties arise from their shape and structure. The last section explains the main concepts within machine learning used for this project.

2.1 Enzymes

Enzymes are proteins, which are macromolecules and what makes them special is that they act as biological catalysts. Catalysts are used in chemical reactions to make them go faster. For instance, enzymes in a human body help to break down food, whereas enzymes used in biofuel production help to increase efficiency of production [47, 49].

The building blocks which make up enzymes are called amino acid residues, residues for short. There are 20 commonly occurring residues and various combinations of them exist as a long chain in enzymes, linked together by chemical bonds [5, Chapter 3]. The biological function of enzymes will vary drastically depending on the order of these residues in the chain. If an enzyme consists of 300 residues, there are 20^{300} different ways of ordering these in a sequence and each combination could yield a different biological function of the enzyme, but most would not be functional at all. Nevertheless, not all of these combinations would be found in nature [12, Chapter 5]. Further details of amino acid residues are explained in Section 2.1.1.

Moreover, in order for enzymes to operate as catalysts they must fold from this long chain of residues into a three-dimensional structure. This is also called their native state which is the natural shape they fold into. Commonly, enzyme structure is referred to as four levels of structures; namely the primary, secondary, tertiary and quaternary structure [12, Chapter 5]. They are visualized in Figure 2.2. Even though enzymes must exist in their tertiary structure to function properly, important information will also be found in the other structures (primary and secondary) and thus it is important to study these as well. Additional information is found in Section 2.1.2.

2.1.1 Amino acid residues

Enzymes are constructed from different combinations of 20 amino acid residues in a long chain, linked together by chemical bonds [5, Chapter 3]. The name amino acid residue comes from the term *amino acid*, which consists of a carbon atom surrounded by a carbon atom, an amino group (NH_2), a carboxyl group (COOH) and a side chain [12, Chapter 5]. The side chain is different for each amino acid, which is what makes each of the 20 amino acid (residues) unique. When two amino acids are linked together, there is a chemical reaction, which results in a water molecule being released and what is left of the two amino acids are now called amino acid residues. In this process the center carbon, also called alpha carbon or Cα, and the side chain remain intact. A residue is explicitly written as NH - CαR - CO (occasionally without the hydrogen atom), where R is the side chain.

The residues, excluding the side chain, are part of the enzyme's *main chain*, or backbone. Figure 2.1 provides a simple example of a main chain with side chains attached. Each residue contains an Cα and for 19 out of the 20 residues, there is also a side chain attached to the Cα [38]. As mentioned, the side chain is different for each of these residues, both in size and which atoms are present, however, each side chain will contain one or more carbon atoms (along with other atoms). Following the alpha carbon at the center of the residue, the first carbon in the side chain is called beta carbon, or Cβ. The second is gamma carbon, or Cγ, and the numbering of the carbons continues according to the Greek alphabet [20]. In particular, when residues are linked together there will be a long chain of alpha carbons, possibly with further carbons attached (if there are side chains present). Therefore, each enzyme will have as many alpha carbons as residues, but not necessarily the same amount of beta carbons. From the 20 amino acid residues, only one, Glycine, does not have a side chain and thus only contains an alpha carbon.

2.1.2 Enzyme structure

There are four levels of an enzyme's structure: primary, secondary, tertiary and quaternary structure as can be seen in Figure 2.2 [12, Chapter 5]. In the *primary* structure the residues, which make up the enzymes, are depicted as a long chain, or sequence. This sequence symbolizes the order in which the residues appear and is a simple, straight-forward way to study from which residues an enzyme is constructed. The *secondary* structures are locally folded structures and are formed when a set of amino acid residues interact in a certain way. Two of the most common secondary structures are called α-helix and β-sheet [12, Chapter 5]. Proteins and enzymes can contain both, oftentimes several, of these structures.

The *tertiary* structure is formed when the long chain of residues, and the previously folded secondary structures, fold into a compact, three-dimensional form. In this stage, enzymes become fully operational and it becomes possible to observe where the residues in each enzyme appear. The tertiary structure of an enzyme is also called its native state as this is the natural shape and structure it folds into. Many enzymes are constructed this way: a single, long chain that folds into a tertiary

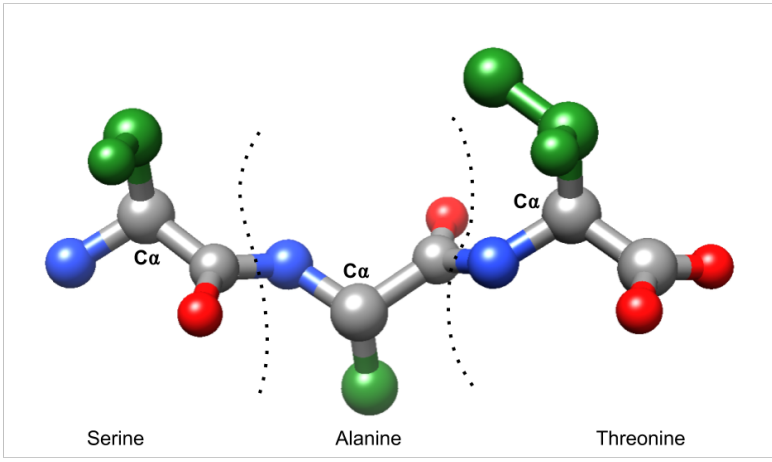


Figure 2.1: Three amino acid residues (Serine, Alanine and Threonine) linked together by chemical bonds. For each residue there is an Cα (in gray) which is a part of the main chain, and for each Cα there is a side chain attached (in green). The oxygen atoms are marked in red, the nitrogen atoms in blue and in each residue there is an additional carbon atom in gray. Molecular graphics and analyses performed with UCSF Chimera, developed by the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco, with support from NIH P41-GM103311 [35].

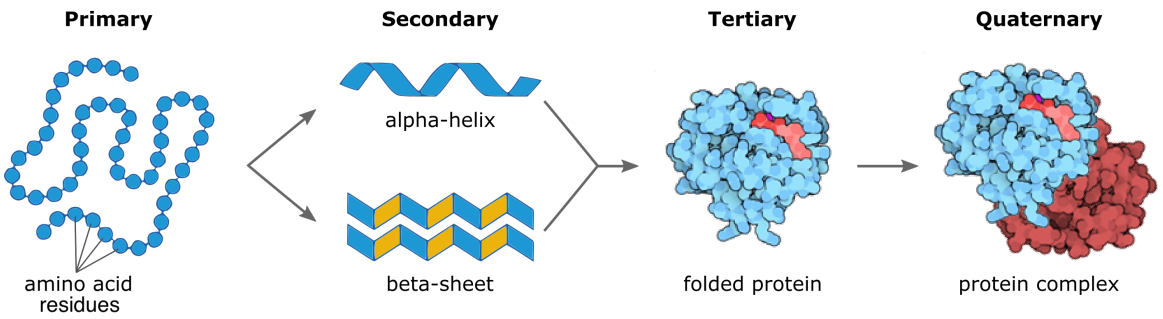


Figure 2.2: The four levels of enzyme structures. Starting at the primary structure, which is a long chain of amino acid residues, it folds into secondary structure elements and then into their tertiary structure. In some cases, when enzymes consist of multiple chains, a quaternary structure is formed. Image provided by Martin Engqvist

structure. However, some enzyme complex, as depicted in the last part of Figure 2.2, consist of more than one chain. When they do, the structure is referred to as the enzyme's *quaternary* structure [12, Chapter 5]. Finally, all levels of enzyme structure are important to fully capture the different features, or properties, that exist.

2.1.3 Enzyme commission number

Enzyme commission (EC) numbers are recommendations by the IUBMB (International Union of Biochemistry and Molecular Biology) on how to classify enzymes by enzyme-catalyzed reactions [11]. The EC number does not specify enzyme structure, only what reactions enzymes catalyze. Moreover, different enzymes which catalyze the same reactions are classified by the same EC number. Similarly, if one enzyme catalyzes different reactions, it is given an EC number for each reaction.

Each EC number is represented by four numbers, separated by a dot, e.g. 1.1.1.1. The first digit represents the top class, the second the first sub-class, and so on. There are 7 top classes (numbered 1-7) and each class has a corresponding name. For instance, enzymes in top class 1 are called oxidoreductases and catalyze oxidation/reduction reactions [11]. In this case, the second digit indicates what atomic group undergoes oxidation. However, for each of the seven top classes, the sub-classes represent different things. Furthermore, if there is uncertainty in the enzyme-catalyzed reaction, each affected sub-class is replaced by a hyphen [45]. Two examples are 1.1.1.-, or 1.1.-.-, where only the fourth sub-class is unknown in the former case, but both the third and fourth sub-class are unknown in the latter case.

2.2 Machine learning

Machine learning is a field within computer science where a computer is not explicitly programmed to perform a certain task, instead it is provided with the ability to

teach itself and learn from past mistakes [30]. A classic example of an application is using machine learning for spam filtering [15]. In this example, a system is given a dataset where each data point is an e-mail, labeled as either spam or not spam. Given enough time to train and learn from the data, it will try to determine if an e-mail should be labeled (also called classified) as spam or not spam, based on what properties the e-mail has. Additionally, when a new data point arrives which does not yet have a label, it will be labeled either as spam or not spam, based on what the system has learned. However, it is not certain this classification will be correct, which is the case when a spam e-mail ends up in the normal inbox, or when a non-spam e-mail ends up in the spam inbox.

A system that trains on and labels new data is called a machine learning model and is an algorithm with a set of hyperparameters that can be tuned based on the application. The data on which it trains on is called the *training set* and similarly, the data with unknown labels is called the *test set*. In order for a model to learn something from the training set, the data must be given one or more *features*. For instance, one feature in the spam example might be if the header contains non-ascii symbols or not. If it does, it *could* be an indicator that the e-mail is spam. Lastly, after the training phase the model is evaluated with the test set, i.e. a metric score is calculated which tells how many data points were correctly labeled [8, 16].

This section is further divided into shorter subsections in order to capture the essential machine learning concepts necessary for this thesis. Namely, the theory behind supervised and unsupervised learning is explained directly below, followed by the bias/variance tradeoff, K-fold cross-validation, linear and nonlinear machine learning models and finally scoring methods used to evaluate the models.

2.2.1 Supervised and unsupervised learning

The example of spam-filtering is known as a *supervised classification learning problem*. In supervised learning, each data point in the training set has a label which can be either a discrete or continuous value. In the former case the approach is known as a classification problem and in the latter case it is a regression problem. Moreover, the opposite to supervised learning is unsupervised learning. Data in these problems does not have labels and as a consequence there is no straight-forward way to define success. Instead, they are solved by trying to find hidden patterns in the data, for instance by clustering [8, 16]. In this project, the focus will be on supervised regression problems, as it is the optimal catalytic temperature (T_{opt}) the model will label the enzymes with.

2.2.2 Bias-variance tradeoff

An essential aspect of machine learning is getting a model to perform as good as possible. Obviously, it would be ideal to classify all new, unseen data samples correctly. However, this is rarely the case as some data samples might be extremely difficult to classify. For example, if the model misclassifies a spam e-mail as non-spam, this e-mail probably had some property which was difficult for the model

to capture. In order to classify this sample correctly, the model would need to be modified so it learns to put this e-mail in the spam inbox. However, by doing so, a sample which was correctly labeled previously, or would be correctly labeled, might now be incorrectly labeled if this change to the model affects this sample. In machine learning terminology, this scenario is often referred to as the *bias-variance* tradeoff [16].

If a model has high *bias*, it means it does not learn anything while training on the training set. If there are patterns in the data, the model does not capture them. This is common if there is not enough data to train on, or if the model is not complex enough. Another term for this is underfitting and this corresponds to the left graph in Figure 2.3. One way to detect high bias is by studying if the training error is high. By acquiring more data or building a more complex model, underfitting can often be avoided.

The opposite to bias is *variance*, and thus if a model has high variance it means it is overfitting the data. This corresponds to the middle graph in Figure 2.3. The model in the figure has learned the data too well and fails to generalize and as consequence the model will both capture the noise in the data, as well as it will be sensitive to outliers. High variance can be detected if the training error is low (as the model has perfectly captured all the train samples), but the test error is high. As the model is too complex, it will most likely fail to classify new samples. Overfitting is common if the amount of data is limited, the data has too many features or if the model is too complex with too many parameters. Therefore, a few countermeasures consist of decreasing the complexity of the model or removing some features. However, by removing features there is also a loss of information. To mitigate this, it is possible to put *weights* on them, thus the more important features will be prioritized by the model.

The bias-variance tradeoff is a tradeoff, since a countermeasure to high bias is adding more data, however, this can easily cause high variance as a consequence. It is important to find the balance in-between, when it is "just right", as in the last graph in Figure 2.3.

2.2.3 K-fold cross-validation

Usually when training and evaluating a machine learning model the dataset is split into a single training and test set and this ratio is usually 70/30 or 80/20, depending on the size of the data. However, if the dataset is small, both the train and test splits will be small so the model will neither have much data to train on, nor to be validated on. Due to this, the prediction scores of the model will be unreliable and unstable. One way to mitigate this is to instead use a technique called k-fold cross-validation. The idea is to split the entire dataset into k parts, or folds. The first fold is put away and will represent the test set. The model will only train on the $k - 1$ remaining folds and evaluated with put-away test fold and this produces one score. This is repeated $k-1$ times (a total of k iterations) and which fold acts as the test fold is swapped between iterations. In the end, each fold will have acted as

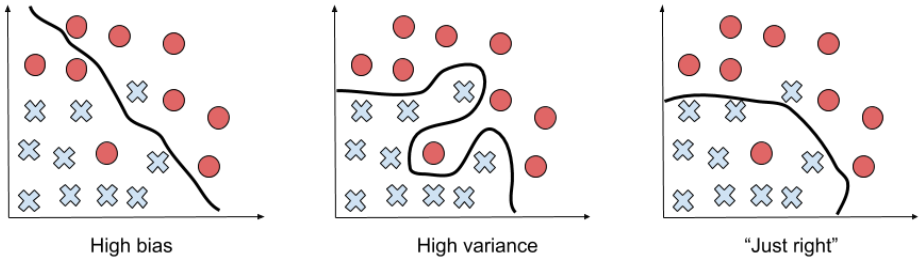


Figure 2.3: The left-most curve illustrates a machine learning model with high bias and the one in the middle a model with high variance. The curve to the right is "just right".

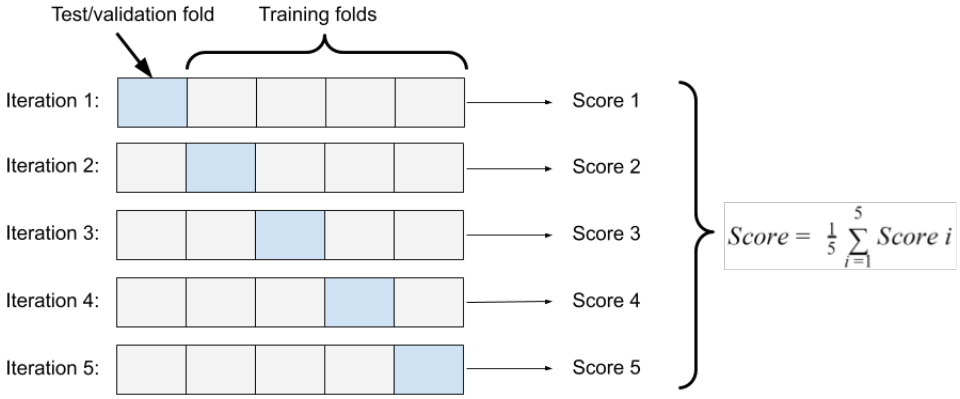


Figure 2.4: 5-fold cross-validation. In the first iteration, the first fold is chosen as the test fold and the other four act as training folds. The model trains on the training folds and is evaluated using the test fold. This process is iterated five times and the final score is the average of the output from the five iterations.

the test fold and there will be k scores and the final score of the model is the average value. Figure 2.4 visualizes this with $k = 5$, in other words, a 5-fold cross-validation [8].

K-fold cross-validation works well when the dataset is small and more variation is needed between the test and training sets. Rather than having simply one test set, as in the traditional 70/30 split, there will have been k test sets. This produces a more reliable score and more information of how the model would perform with new, unseen data. On the other hand, one downside is that this takes more time to run, as the machine learning model must be run k times.

2.2.4 Linear and nonlinear models

Different machine learning models exist for different applications. The two main groups which they are usually divided into, are linear versus nonlinear models. If

the input data (the features) is linearly correlated to the output data (T_{opt}), a linear model is often preferred. On the other hand, if they are not linear, a nonlinear model is required for good performance. A downside to nonlinear models is that they are usually more difficult to interpret. For both linear and nonlinear models, there is a cost function, or error function, that must be minimized, which symbolizes the amount of error a model makes.

2.2.4.1 Linear regression

Linear regression is a linear machine learning model which tries to find a line to best fit the data. Suppose the dataset now consists of enzymes and this is a regression problem to predict the enzymes' T_{opt}. For the sake of easy visualization, assume the training data only has a single feature, namely enzyme size. This feature is used to predict the temperature, see Figure 2.5. The x-axis represents the feature and the y-axis the true value, the T_{opt} of the enzyme. Using the method of least squares, linear regression attempts to fit a line which minimizes the residual sum of squares between the true values and the predicted values.

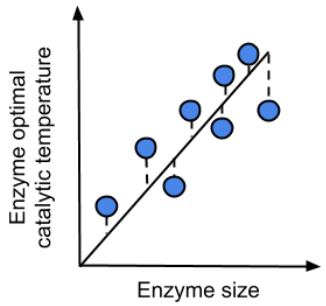


Figure 2.5: Linear regression example with only one feature. This is a visualization of how enzyme size could help predict enzyme T_{opt}. For every feature added, there will be an additional dimension to this graph.

2.2.4.2 Trees

Trees, for instance decision trees, are models which are useful when the data is non-linear. They consist of a root at the top of the tree, followed by other nodes. Each parent node has at most two children, where each child either has children of its own, or is a leaf node. For all nodes that are not leaves, the node represents a question where the answer is either yes or no, and leads to either the left or right child respectively. If the node is a leaf, the node represents the prediction, i.e. a numerical value for T_{opt}. An example of a small decision tree can be found in Figure 2.6.

One of the issues when constructing a decision tree is what question to start off with. This is an iterative approach which becomes more complex the more features that are included in the dataset. Nevertheless, the solution is to consider one feature at a

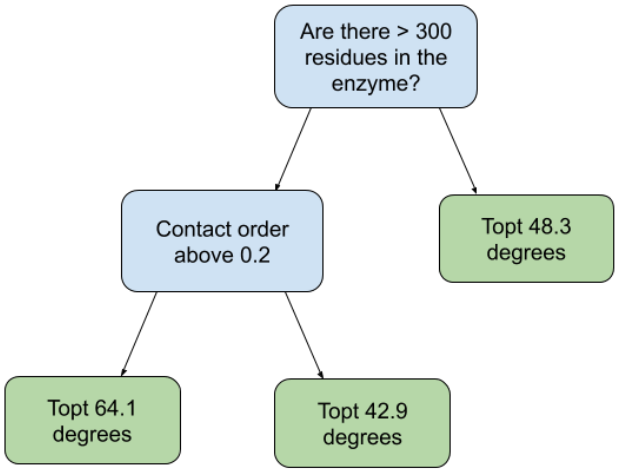


Figure 2.6: Simplified example of a decision tree.

time using different thresholds and for each, study how well it predicts the outcome. Measuring the sum of squared residuals, the error for each combination, choose the feature with the smallest error as the root. This approach continues iteratively for the rest of the tree, and the same question may appear more than once but with different thresholds. Each time there is a question node, the dataset is split into smaller groups depending on if they belong to the yes/no branch. In this manner, if the algorithm for splitting the tree would continue until each leaf only contains a few data points, it will most likely mean this tree is now overfit (it is too precise). A solution for this is to only split a node into two new nodes when there is some minimum number of data points in this category. Lastly, in order to choose the numerical value for the leaves, take the average value (T_{opt}) for each of the subset of enzymes that fall on the respective yes/no branch and place this value in the leaf [16].

2.2.4.3 Ensembles

An example of an ensemble model is random forest, which is created from multiple decision trees. The first step is creating a so-called *bootstrapped* dataset. From the original dataset, of size n, draw n random samples and add to the bootstrapped dataset. Both the bootstrapped and the original dataset is now of the same size, however, there will be data samples which do not appear in the bootstrapped dataset or, similarly, some samples which appear more than once. Moreover, from the bootstrapped dataset, a decision tree is created but only a random subset of features is chosen at each step (at each node which is not a leaf). This process continues iteratively for a chosen number of steps, in each step a new bootstrapped dataset is created and thus a new decision tree. The size of the trees will vary, and all trees

together are what make up a random forest [16].

When classifying a new sample, run it through all of the trees and get a prediction from each. The final prediction from the random forest is the average value of all individual predictions. Finally, in order to estimate how well a random forest performs, consider one tree at a time. Run the samples from the original dataset, which did not appear in the bootstrapped dataset, through the respective decision tree. Based on how well it performs, it is possible to study if it can perform even better by choosing a different subset of features, with different sizes of those subsets. It is possible to choose all features in the dataset, however, this method is prone to overfitting [16].

2.2.4.4 Support vector regression

Support vector machines for regression (SVR) are defined by the following terminology: a hyperplane, a margin, support vectors and a kernel. The support vectors are data samples, a subset from the training dataset which are used to create a hyperplane that satisfies a linear regression function $f(\boldsymbol{x})$, defined as below:

$$f(\boldsymbol{x}) = \boldsymbol{x} \cdot \boldsymbol{w} + b \text{ , where } \boldsymbol{w} \in X, b \in \mathbb{R} \text{ , and } X \text{ is the input space [43]} \tag{2.1}$$

The regression function is used to fit the data, however, to allow some misclassifications and in turn make the model more flexible, there is also a margin around the function. The margin is also called a symmetric, ϵ-insensitive tube, and stretches at most $\pm\epsilon$ from $f(x)$, as depicted in Figure 2.7. Predictions inside the tube are not penalized in the cost function, as long as they are at a distance of maximum ϵ. Furthermore, ϵ is a hyperparameter that must be tuned accordingly, as a too large value will cause the model to overlook large errors which would otherwise be penalized (the tube is too wide). Similarly, if it is too small, it will instead vastly increase the errors in the cost function and therefore will attempt to fit the regression function more precisely, which can lead to overfitting [4].

Lastly, SVR:s use something called a kernel function. If the data is not linearly separable as it is, it is possible to use a kernel function, which "transforms" the data into a higher-dimensional space in which the data is linearly separable. Although SVR:s are linear in nature, using kernels makes it possible for them to operate also on nonlinear data.

2.2.5 Scoring methods

To estimate a model's performance, scoring methods are needed. One common scoring method for regression problems, which is going to be the main approach in this thesis, is the R^2 score. This is a metric which indicates how well the features explain the variation in the true value (T_{opt}). For instance, in the above example with linear regression, the R^2 score tells how the variation in T_{opt} can be explained,

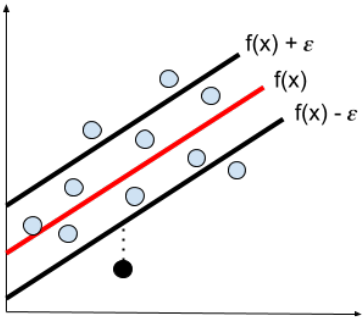


Figure 2.7: An SVR with regression function $f(x)$ surrounded by the ϵ-tube.

or captured, by enzyme size. A score of 80% means, in this scenario, that enzyme size explains 80% of the variation. Alternatively, the relationship between enzyme size and T_{opt} accounts for 80% of the data and there must be something else that accounts for the remaining 20%.

Mathematically, it is calculated by first calculating the mean of the true values ($\overline{v_t}$). Taking the residual sum of squares between the true values (v_t) and predicted values (v_p), divided by total sum between the true values and mean, and subtracting this from 1, gives the R^2 score (Equation 2.2).

$$R^2 = 1 - \frac{\sum(v_t - v_p)}{\sum(v_t - \overline{v_t})} \tag{2.2}$$

An R^2 score of 1.0 is the best possible score, in which case the chosen features accounts for 100% of the variation. On the other hand, an R^2 score of 0.0 means no variation could be explained (regression is equal to the mean value). It is possible to for R^2 to be negative because the model can be arbitrarily bad.

3

Datasets

The datasets used in this project were gathered from multiple different sources, combined and pre-processed prior to feature calculations. Section 3.1 describes how the data extraction was performed, Section 3.1.1 describes the format of the data files, Section 3.1.2 describes how the labels (T_{opt}) were retrieved and finally Section 3.1.3 describes the how the data was filtered.

3.1 Extracting data

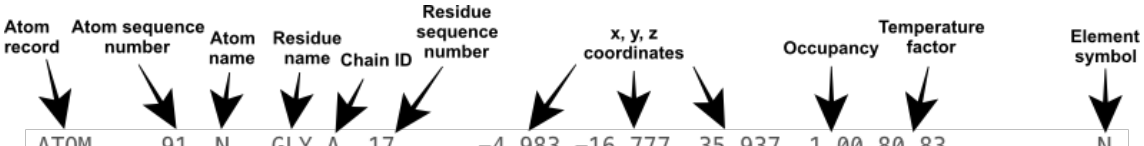
Four databases were used to extract the data and combine structural information with T_{opt}. The first, main, database that was used was The Protein Data Bank (PDB)[1], which contains roughly 150,000 proteins along with their experimentally determined structures [6]. Besides this, SWISS-MODEL and ModBase were used as these contain predicted structures of enzymes [48, 14, 36]. In those cases when an experimental structure was not available, but a predicted structure was, these two were used instead. All three databases use the same data file format, also called the PDB file format (details about the file format is in Section 3.1.1). From this, it was possible to calculate the structural features, further details in Chapter 4.

Although PDB, SWISS-MODEL and ModBase contain a large set of enzyme structures, only a smaller subset of these are in fact labeled with their T_{opt}. This information was necessary to perform the machine learning calculations, as the models need the labels to train. Moreover, the optimal temperature labels were not present in either one of the three databases, but came from the fourth, and last, database: namely BRENDA [22]. How all four databases were combined to create a dataset for this thesis is presented in Section 3.1.2.

3.1.1 Protein Data Bank data files

The PDB file format is a standardized format that comes from the Protein Data Bank. At the top of each file, there is information of how the protein structure was either experimentally determined, or predicted. The predicted structures usually come from homology modeling; predicting enzyme structure from its sequence and comparing it to other known structures, which have a similar sequence [48].

[1] `www.rcsb.org`



ATOM	91	N	GLY	A	17	−4.983	−16.777	35.937	1.00	80.83	N
ATOM	92	CA	GLY	A	17	−5.606	−18.051	36.286	1.00	81.71	C
ATOM	93	C	GLY	A	17	−7.077	−18.058	35.895	1.00	82.41	C
ATOM	94	O	GLY	A	17	−7.580	−17.051	35.376	1.00	82.56	O
ATOM	95	N	PRO	A	18	−7.781	−19.188	36.135	1.00	82.76	N
ATOM	96	CA	PRO	A	18	−9.174	−19.296	35.701	1.00	82.97	C
ATOM	97	C	PRO	A	18	−10.123	−18.550	36.633	1.00	83.16	C
ATOM	98	O	PRO	A	18	−9.937	−18.560	37.850	1.00	83.13	O
ATOM	99	CB	PRO	A	18	−9.438	−20.805	35.729	1.00	82.95	C
ATOM	100	CG	PRO	A	18	−8.535	−21.325	36.782	1.00	83.14	C
ATOM	101	CD	PRO	A	18	−7.339	−20.388	36.870	1.00	82.88	C

Figure 3.1: An extract of a PDB file. Here, 'CA' signifies the alpha carbon and 'CB' the beta carbon of the amino acid residue. For each atom there are (x, y, z) coordinates.

Figure 3.1 shows an extract of how the structural information is presented in a PDB file. Every row with the ATOM record was parsed, as this symbolizes all the atoms in an enzyme. For each atom the (x, y, z) coordinates are present, as well as which residue the atom belongs to. The chain ID column shows the number of chains in the enzyme. As discussed in Section 2.1.2 about enzyme structure, if the enzyme only has a tertiary structure, the PDB file will only contain one chain. If it has a quaternary structure as well, there will be several chains in the file. This information was relevant for calculating the features later, as it will show that in some cases, it was necessary to keep the chains separate. Moreover, the data structures which were saved from each of these files were: the chains that were present, which residues belonged to which chain, and furthermore which atoms belonged to which residue. In the last scenario, the atoms were saved as "only alpha carbons", "only beta carbons" as well as "all atoms".

The two last columns are occupancy and temperature factor. In some cases, side chains have different conformations (spatial arrangements) due to local flexibility. In particular, some atoms may have been identified at different (x, y, z) coordinates. If an atom is only ever identified in one place, occupancy will have a value of 1.0. If it is found in two places with equal probability (or other distributions), there will be two rows with the same atom, with different coordinates and also marked with an "alternative location" tag, and occupancy will in this scenario have a value of 0.5 for each. Further details of how occupancy was used in this project is found in Section 3.1.3.

Temperature factor is, unfortunately, not correlated with the optimal catalytic temperature. All atoms inside enzymes move around with varying flexibility. The temperature factor is an indication of how much an atom moves around its average position. This column was not used in this project.

	Number of enzymes
Unique sequences queried	1902
Structures retrieved from PDB	305
Structures retrieved from SWISS-MODEL	796
Structures retrieved from ModBase	454
Sequences with no found structure	*348*
Same sequence with different structures	349
Total unique structures	**1554**
Total structures	**1903**

Table 3.1: Summary of the distribution of the data files.

3.1.2 Retrieving labeled data

As mentioned, the labels were not present in either PDB, SWISS-MODEL or ModBase, and instead were retrieved from the database BRENDA [22]. BRENDA is a dataset which contains functional information about enzymes, for instance their T_{opt}. The data from BRENDA had previously been extracted by Engqvist *et al.*, who provided the necessary data for this project [27]. From BRENDA, they extracted 5343 enzymes. In addition, they filtered the data in order to reduce the noise in T_{opt}, which reduced the size of the provided dataset to 1902 enzymes.

Each of these enzymes came with a unique identifier, which came from the UniProtID, or Universal Protein Identifier [10]. In order to retrieve the structures of the enzymes, the identifier was used to query PDB, SWISS-MODEL and ModBase. As experimentally determined structures are more reliable, PDB was queried first. Out of the 1902 enzymes, 305 were found in PDB. In the next step, when SWISS-MODEL was to be queried, the 305 files which had already been found in PDB were excluded from the search. From SWISS-MODEL, 796 files were found and finally from ModBase 454 files. In the end, there were 348 files which were not found to have a structure in any of the three databases and were thus excluded from the dataset. In total, 1554 unique structures were found. Additionally, there were 349 structures which came from one or more of the same sequence from SWISS-MODEL. Specifically, from the same sequence slightly different structures were predicted. Thus, the size of the dataset was $1554 + 349 = 1903$ enzymes. Table 3.1 summarizes this information.

3.1.3 Pre-processing data

Before feature calculations could start, pre-processing steps were applied to the 1903 established data files.

Hydrogen atoms

First, most of the structures in PDB are determined through X-ray crystallography, which has a hard time resolving hydrogen atoms [24]. Furthermore, there was still a small fraction of PDB files that contained hydrogen atoms but in order to treat all data files the same, all hydrogen atoms were excluded.

Alternative locations of residues

As mentioned previously, if an atom is identified in more than one place in a residue, there will be duplicate rows of this atom, with an alternative location tag, as well as an occupancy value which is not 1.0. In order to not overrepresent these atoms, the first alternative location was parsed to the data structure, and the other one ignored. Furthermore, in a few cases there were entire residues with the alternative location tag, which happens if the residue has been identified at another location. Similarly for this scenario, only the first alternative location was parsed.

Multiple models

Sometimes a PDB file contains several *models* of the same enzyme. As described on the PDB website[2], each enzyme model should have the same atoms, however, their locations will vary. Generally, if an enzyme has several models, it is due to that model being identified through the method known as NMR (nuclear magnetic resonance spectroscopy). If an enzyme has several models, the PDB file will have the "MODEL" tag, and a sequence number for each such model. When parsing the enzyme, only "MODEL 1" was used.

Disconnected residues

While parsing the data files there were instances of disconnected residues. In the PDB files, the residue sequence number usually starts at 0 or 1, however, the identified outlier residues started at a negative number. Particularly, in one example, with PDB code 2vbf (Figure 3.2), these residues start at -14 and end at -10, followed by a "jump" in the PDB file, with the next residue starting at 1. By studying both the coordinates, and the structural image, it became clear these residues were disconnected from the structure. As some metrics, for instance radius of gyration, is sensitive to "outliers" , residues with a negative sequence number were skipped.

Unparseable by external program

Two external programs were used as an aid when calculating two of the features (see Section 4.1.4 on Surface Atoms and Section 4.1.5 on Torsion Angles). There were nine PDB files that were unparseable by the program that calculated the surface atoms. These were excluded as to not spend a disproportional amount of time attempting to fix the error.

[2] `http://www.wwpdb.org/documentation/file-format-content/format33/sect9.html#MODEL`

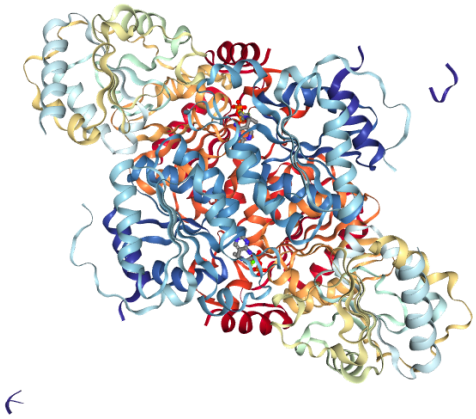


Figure 3.2: Enzyme with PDB code 2vbf, with two outliers on either side. Image from the RCSB PDB (`www.rcsb.org`) of PDB ID 2vbf, in [7]. Created with NGL viewer [41].

Duplicates

Enzymes with the same EC number, sequence, structure and the same features were considered to be duplicates. Nevertheless, it is possible these enzymes have different T_{opt} values, based on what has been previously reported. If the difference in T_{opt} was small (5-10 degrees) one enzyme was kept with the averaged T_{opt}. However, if the difference was too large, all duplicates were removed as it became uncertain which T_{opt} was correct. In total, 12 duplicate enzymes were removed.

Miscellaneous

Besides the 21 above excluded PDB files, four additional PDB files were excluded, each with its own reason. Out of these four, one PDB file was empty, one was a DNA, one was an RNA and the last file had two unparseable residues. Specifically, there was a glycine residue with sequence number 355 and and an arginine residue with sequence number 355A. In Section 3.1.3 it was mentioned that residues may have an alternative location, however, that pre-processing step referred to when the same residue had different locations. In this case, it was two different residues with similar locations. Rather than choosing which residue to use, the entire PDB file was excluded. In total, 25 PDB files were omitted from the remaining of the project.

4

Feature Extraction

This chapter describes how the feature extraction of the enzymes was performed. A considerable part of this thesis was to analyze what features were thought to be relevant for T_{opt} prediction and to hand-craft them. There was no trivial answer as to what features would be the best to use, instead the effort needed to calculate each feature was weighed against an estimation of relevance.

4.1 Feature calculations

There exist several unique features which explain an enzyme's characteristics, both in terms of shape and structure, but also functionality. To understand these features are important steps to understand the enzyme itself. Moreover, it makes it possible to draw comparisons between enzymes and other proteins [19].

The enzyme features exist in multiple spatial dimensions. On the one hand, when we only consider the primary structure of an enzyme, an example of a one-dimensional feature is the amino acid residue frequency, used in [26]. This is also called a sequential feature. On the other hand, the secondary, tertiary and quaternary structure of an enzyme give rise to more complicated, three-dimensional features. Five examples of such features are explained below, namely pairwise interaction between residues, contact order, radius of gyration, atomic groups on the surface and residue torsion angles.

Choosing which features that should be calculated based on relevance was not a trivial task and thus an estimation of the correlation between effort needed to calculate the feature, and a guess of relevance, was made. The first feature, pairwise interactions between residues, was included as calculations were straight-forward and fast. The second, contact order, was included as there is a correlation to an enzyme's folding rate (how fast the enzyme folds from its primary to its tertiary structure), and thus it was interesting to study if there was also a correlation to catalytic temperature [37]. Thirdly, radius of gyration is directly related to enzymes' compactness, and similarly to contact order, it might be correlated to catalytic temperature [28]. The fourth feature, atomic groups on the surface, describes properties of the surface of an enzyme, unlike the other features which describe the internal properties and it was for this reason surface atoms was included. The fifth and last feature, residue torsion angles, tells how rotated an enzyme is, and likewise, the hope was this would

be correlated to T_{opt} as well.

4.1.1 Pairwise residue-residue interactions

Pairwise interactions between amino acid residues in an enzyme is a metric to determine how often any pair of residues interact with each other. Particularly, how often any two residues are within a set distance threshold of one another. It is common to think about the residues in a simplified manner and only consider the alpha carbons of the main chain, as well as only the beta carbons of the side chain (if present), thus ignoring other atoms. When calculating the pairwise interaction between two residues, the distance is measured from their respective *beta* carbons [25]. Concerning Glycine, which does not have a beta carbon, it is common to use its alpha carbon instead. Moreover, two residues are considered to be interacting if the distance is within a distance threshold (usually less than $8\text{Å} = 8 \cdot 10^{-10}\text{m}$) [1].

The first step was to construct a 20x20 pairwise distance matrix for all residue pairs and every time two residues interacted, a "+1" was added to the cell. Thus, in total there would be up to 210 interactions, since, from Figure 4.2, residue 13 interacting with residue 30 is equal to residue 30 interacting with residue 13. From this count it was apparent how often, if ever, a pair of residues interacted within an enzyme. Such a pairwise distance matrix can be seen in Figure 4.1. Finally, this feature was saved as frequencies, by dividing each count by the total number of interactions.

When considering all residues in a sequence, it might seem unfavorable to include two adjacent residues in the calculations. Specifically, any two adjacent residues in the primary structure, will most likely be interacting in the tertiary structure as well. Therefore, a separation distance of 1 was included, so for each residue in the sequence, the residue that directly follows was skipped.

If the enzyme had multiple chains, an extra step had to be made. First of all, for each separate chain, the pairwise residue interactions were calculated. Secondly, it was calculated between chains and the results aggregated. For instance, if an enzyme had two chains A and B, the interactions were measured internally for A and internally for B, followed by all residues in A which were in close contact to residues in B.

4.1.2 Contact order

Contact order is a metric which measures the average sequence distance between two interacting residues, normalized by the total sequence length [37, 21]. Study Figure 4.2 of protein with PDB code 1CRN. If only the primary structure is considered, the residues with sequence number 13 and 30 would seem far apart, as they are separated by another 17 residues. Nevertheless, the primary sequence does not speak the full truth and in fact, in the tertiary structure, it can be seen how they are in close contact. Accordingly, the contact order finds two interacting residues in the tertiary structure, and measures, on average, how far apart they are in the primary structure. It will be low for enzymes that tend to have their contacts mainly between residues

	A	C	D	E	F	G	⋯
A	16						
C	6	0					
D	7	2	0				
E	8	0	2	0			
F	8	0	1	0	2		
G	18	2	8	2	3	5	
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

Figure 4.1: An extract for a number of residue-residue interactions for an enzyme. Only the lower half of the triangle was considered.

which remain close together in the sequence (majority of local contacts). Conversely, the contact order will be high for proteins which has the majority of their contacts between residues far apart in the sequence (non-local contacts).

In comparison to the pairwise interactions metric, where only beta carbons were used (except for Glycine), when calculating the contact order of an enzyme, all atoms were used. Thus, two residues could be interacting although their beta carbons did not fall within the set distance threshold, as another pair of atoms might be close. As in the metric definition, "total sequence length" implies the total number of atoms which construct the enzyme.

There is both the *relative* contact order and the *absolute* contact order. Both equations come from the papers by Baker *et al.*, [21, 37]. The absolute contact order (Abs_CO) was calculated as in Equation 4.1, and the relative contact order (CO) as in Equation 4.2.

$$Abs\_CO = \frac{1}{N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Delta S_{ij} \tag{4.1}$$

$$CO = \frac{1}{L \cdot N} \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} \Delta S_{ij} = \frac{Abs\_CO}{L} \tag{4.2}$$

In both equations, N is the total number of interactions in the enzyme, L is the total number residues in the enzyme and ΔS_{ij} is the sequence separation between two residues i and j.

To calculate contact order, the program iterated through all residues r_i, and all

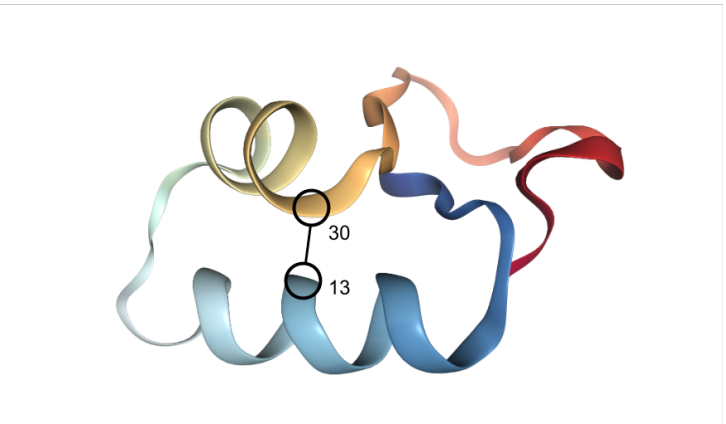


Figure 4.2: Visualization of the protein with PDB code 1CRN. It shows how residue 13 and residue 30 are close together in the tertiary structure, although in the primary structure, they are 17 sequences apart. Image from the RCSB PDB (`www.rcsb.org`) of PDB ID 1CRN, in [44]. Created with NGL viewer [41].

residues $r_j : j > i$. Further, it iterated through all atoms a_k in r_i, and all atoms a_l in r_j. If the distance between a_k and a_l in the tertiary structure fell below a threshold then L was incremented by 1 and ΔS_{ij} was calculated for the two residues. In order to avoid unnecessary calculations early stopping was applied. Meaning, if two atoms were found to be in close contact in two residues, the calculations stopped and continued with the next residue in the sequence. Finally, if the enzyme had multiple chains, the contact orders were calculated for each chain separately and the final result was the average value.

4.1.3 Radius of gyration

The radius of gyration of an enzyme measures its compactness and distribution of atoms around its center of mass. It will be low for compact enzymes and high if the enzyme is loose and less compact [28]. Mathematically, it is the root mean square distance of the atoms in the enzyme from its center. To calculate the center of mass, either the atoms' individual masses are included, or they are assumed to have a uniform mass. For this thesis, the latter was applied, see Equation 4.3 [28].

$$R_c = \sum_{i=1}^{N} a_i / N \tag{4.3}$$

From this equation, a_i are the (x, y, z) coordinates of the *i:th* atom and N is the total number of atoms in the enzyme. The center of mass, R_c is also represented as (x, y, z) coordinates. Next, the radius of gyration was calculated as in Equation

4.4 [28]. Furthermore, this metric disregards if the enzymes are constructed from multiple chains, and instead treats them as one large chain.

$$R_g = \sqrt{\sum_{i=1}^{N}(a_i - R_c)^2/N} \tag{4.4}$$

4.1.4 Atomic groups on the surface

The shape of the surface of a protein is an important aspect to understand and predict protein-protein or protein-ligand interactions (a ligand can be an ion or molecule), where the latter is used in, for instance, drug design [29]. Tsai *et al.* defined 13 *atomic groups* which can be found in proteins [46]. Each atomic group is labeled as *XnHm*, where X is the chemical symbol of a non-hydrogen atom (e.g. C, N, O, S representing carbon, oxygen, nitrogen and sulfur respectively), n is its valence (connectivity) and Hm is the number (m) of attached hydrogen atoms to the non-hydrogen atom.

An external program was used which identifies which atomic groups are solvent accessible and therefore considered to be on the surface of a protein. The program implements a technique for studying the protein surface which involved searching for triplets of atomic groups that could be touched simultaneously by rolling a small probe over the surface. The probe, shaped as a sphere, represented a small molecule. Whenever the probe touched three atomic groups simultaneously, the triplet was recorded and depending on the properties of the triplet, the authors of the program were able to deduce preferences the triplets had for different ligands [29]. The feature used in this thesis was a frequency for how many atomic groups that were identified on the surface of each protein. This produced an additional 13 features, one feature for each atomic group.

4.1.5 Residue torsion angles

Each residue, can be written as NH - Cα - CO (omitting the side chain from the alpha carbon). There is a rotational restriction between CO-NH due to the chemical bond that connects them, and as a consequence there is a restriction to how the other atoms are positioned in the protein backbone. In particular, each Cα - CO - NH - Cα constructs a segment which lies in a plane, connected at the Cα. Therefore, each residue reaches across two planes, and, additionally, each plane contains parts of two residues [12].

There is, however, rotational freedom between N - Cα and between Cα - C. The angle between these two pairs are called torsion angles and are angles between the two planes the residue is part of. The angle between N - Cα is called the *phi* angle (ϕ) and the angle between Cα - C is called the *psi* angle (ψ), see Figure 4.3. Note that, in the Figure, the hydrogen atoms are not present as these are hard to resolve in X-ray crystallography [24].

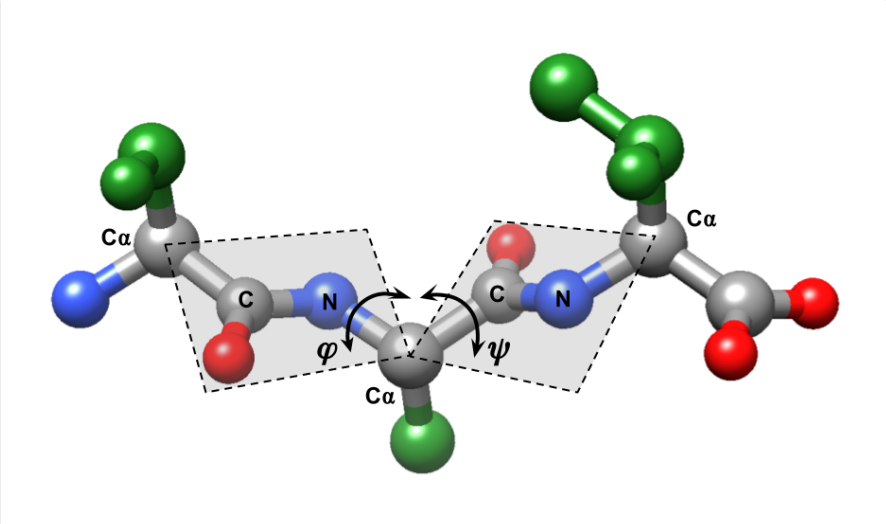


Figure 4.3: Similar figure as Figure 2.1. Each Cα - CO - N - Cα forms a plane, connected at the Cαs. The phi angle is the torsion angle between N - Cα, and psi is the torsion angle between Cα - C of a residue. Molecular graphics and analyses performed with UCSF Chimera, developed by the Resource for Biocomputing, Visualization, and Informatics at the University of California, San Francisco, with support from NIH P41-GM103311 [35].

Basin	A	B	D	G	L	P	R	T	U	V	Y
Phi	-62	-120	-134	-93	51	-64	-68	55	82	-93	77
Psi	-42	135	70	95	42	139	-18	-129	-3	2	-171

Table 4.1: 11 pre-defined basins with labels. For each basin there is a phi and psi angle pair that acts as the centroid of that basin, and where each basin region stretches $\pm 10^{\circ}$ in each direction from the centroid.

To calculate the phi and psi angle for residue i meant calculating the angles between the planes. In particular, to calculate phi for residue i, N - Cα - C from i, as well as C from $i-1$ were necessary variables. Similarly, the variables for psi were N - Cα - C from i and N from $i+1$.

As both residues $i-1$ and $i+1$ were needed to calculate the torsion angles for residue i, it was not possible to calculate the phi angle for the last residue or the psi angle for the first residue in the chain. As a consequence, these two residues were not considered in feature calculations. Furthermore, in some cases, there were residues missing in the PDB files and to account for this, the distance between the alpha carbons between two residues were calculated. Due to stereochemical constraints on bond length and bond angles, the distance between two consecutive alpha carbons is close to 3.8Å [31]. To allow for a margin of error a maximum distance of 4Å between alpha carbons was allowed. If they were further apart than this, they were not considered.

Most of the residues now had a corresponding phi/psi pair. In [9], 11 basins were defined by a phi and psi angle, see Table 4.1. The idea in the paper was to transform a protein structure into a basin sequence by mapping each residue of the protein to its nearest basin. This basin sequence was then compared to another protein's basin sequence, for a faster comparison of protein structures. In this thesis, these 11 basins were used as 11 additional features. For each residue's phi/psi pair, the Euclidean distance was measured to each basin and the residue was placed in the nearest one. Each feature was saved as a frequency describing how often residues were mapped to each basin.

5

Experiments

Two different experiments were performed in order to test performance of the models and the new, structural features. *Experiment 1* used the entire dataset and cross-validation to estimate the performance of multiple models by searching for the best hyperparameters. Different feature combinations were used in the models to study which ones were the most relevant to predict T_{opt}. Following this, the best model, together with the best hyperparameters and best feature combination, is used for *Experiment 2*. A training and test set were manually constructed in Experiment 2, where the test set only contained *homologous* enzymes, and the training set *non-homologous* enzymes (two enzymes are homologous if their sequences are at least 25% similar, more details in Section 5.2). The motivation behind Experiment 2 was to study if a model trained on only non-homologous enzymes can correctly predict T_{opt} for homologous enzymes.

The rest of this chapter presents both experiments; it describes the background of each experiment, each of the setups, individual results and a discussion on the outcome.

5.1 Experiment 1

This section describes the process and results from running machine learning models on different feature combinations. A selection of the results is displayed in this section, the remaining are found in Appendix A.

5.1.1 Background

Previous score obtained by Engqvist *et al.* was an R^2 score of 0.4, with only a sequential feature set present [26]. Experiment 1 was constructed to study if there are structural features which can improve the scores further. This was the original hypothesis, that the structural features would carry additional information about the thermostability of enzymes, which the sequential feature do not. Thus, a set of structural features were hand-crafted and different feature combinations were run through the machine learning models to study which feature set was the most significant for predicting T_{opt} of enzymes.

5.1.2 Setup

Five structural features were hand-crafted for the purpose of Experiment 1. Namely, a pairwise distance matrix for residue-residue interactions (Section 4.1.1), contact order (which includes both the relative and absolute contact order, Section 4.1.2), radius of gyration (Section 4.1.3), atomic groups on the surface (Section 4.1.4) and residue torsion angles (Section 4.1.5). These features will for the remaining of the chapter be abbreviated with PDM, CO, RoG, surface atoms and Phi/Psi, respectively. All combinations of these structural features were used in the models, a total of 31 different combinations. Further combination that were run were sequential features alone and sequential features combined with all structural features. Moreover, after running the first 31 combinations, it was concluded that PDM and surface atoms out-performed the other structural features. Therefore, the sequential features were also combined with them individually, and once with both. In total, 36 different combinations of features were used in the models.

The machine learning models that were used for Experiment 1 were linear regression, bayesian ridge, elastic net, decision tree, random forest and SVR. Both bayesian ridge and elastic net are two linear models, similar to linear regression. All models were implemented with Python, using the Scikit machine learning library [34]. For the first four models, a standard 5-fold cross-validation (CV) approach was used and no particular hyperparameter tuning was done. The reason was that the linear models and decision tree were not expected to perform as well as random forest and SVR. For random forest and SVR a nested CV approach was implemented. The outer CV was a standard 5-fold CV, and the inner CV implemented a 3-fold GridSearchCV, a tool to exhaustively search for hyperparameters. Thus, the outer CV split up the dataset into five folds, were four folds represented the training set. For these four folds, another 3-fold CV was run where the model optimized the hyperparameters, and used the best set of parameters on the fifth test fold. This process was iterated five times in total.

5.1.3 Results

Out of all possible feature combinations, it became apparent that combinations of PDM, surface atoms and sequential features worked best. CO, RoG and the Phi/Psi features did not perform as well on their own, as can be seen in Figure A.1 in Appendix A. All different combinations of structural features are found in Figures A.1-A.7. Note, for the test score plots, linear regression was not included as it performed significantly worse than the other models and produced negative R^2 scores which decreased the readability in the plots.

Throughout Experiment 1, random forest and SVR were the models that performed the best with the feature sets. Below are five plots that display the R^2 test scores from all models except for linear regression, for five different feature combinations. Figure 5.1 displays the results from using sequential features, with similar results as in [26]. Figure 5.2a includes the feature PDM, Figure 5.2b includes surface atoms, and Figures 5.3a-5.3b displays the results from combining the sequential features

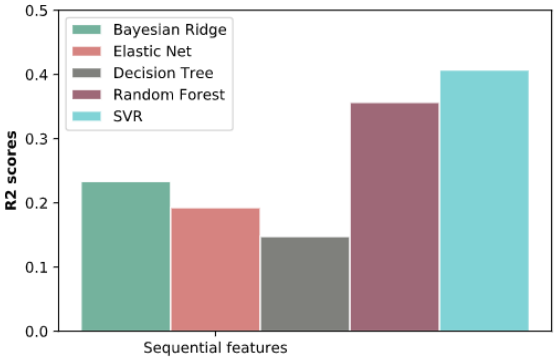
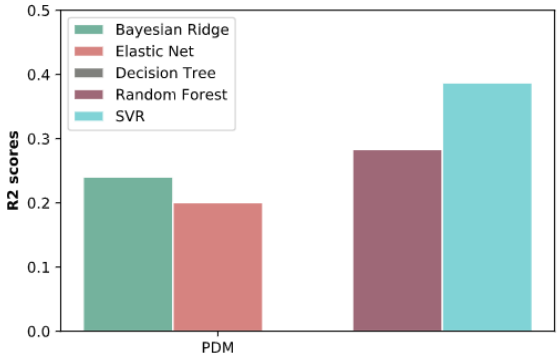
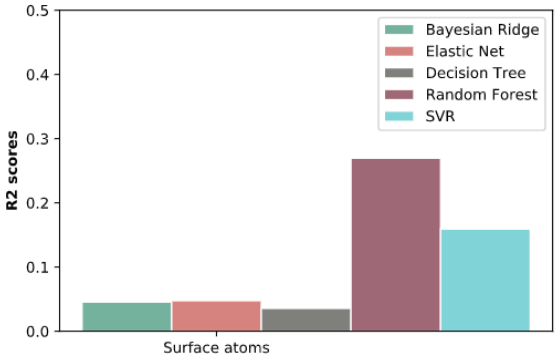


Figure 5.1: R^2 test scores from running machine learning models with sequential features (amino acid residue frequencies).



(a) Test scores from running the PDM feature.

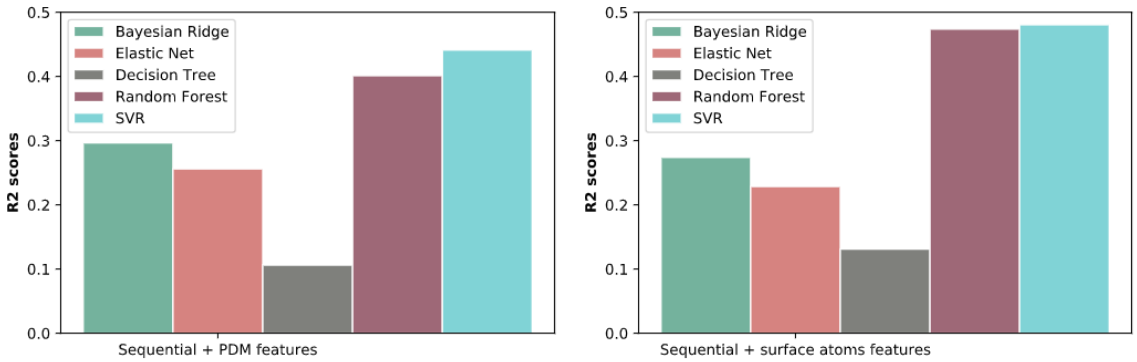


(b) Test scores from running the surface atoms feature.

Figure 5.2: Test scores scores from structural features PDM (a) and surface atoms (b)

with PDM and with surface atoms, respectively. On its own, PDM produced better results than surface atoms. However, when combined with sequential features, it was the combination of sequential features and surface atoms which perfomed better than sequential features and PDM. Furthermore, the combination of all three did not produce better scores than when two features were used at a time, as seen in Figure A.8 in Appendix A. The sequential features were also combined with all structural features, but similarly it did not produce better results than other feature combinations, see Figure A.9.

Table 5.1 presents for each of the five different feature combinations, the R^2 training and test scores for random forest and SVR. Although most of them produced a high training score, the test score is lower which signifies the models were overfitting the training data. However, although overfit, both combinations of sequential and structural features resulted in improved test scores than previously obtained. The sequential and PDM features yielded scores of 0.401 and 0.474 for random forest and SVR respectively and the sequential and surface atoms features yielded scores of 0.474 and 0.480. This means that the relationship between these features and T_{opt}



(a) Test scores from sequential + PDM features.

(b) Test scores from sequential + surface atoms features.

Figure 5.3: Test scores scores from running sequential features together with PDM (a) and surface atoms (b).

account for 48% of the data. This in comparison with the previous scores, where the relationship between sequential features and T_{opt} accounted for 40% of the data.

5.1.4 Discussion

The original hypothesis in this thesis was that new, structural features should provide additional information in predicting T_{opt} for enzymes. The sequential features previously produced R^2 test scores of 0.4, and together with the structural features predictions improved to 0.48, thus confirming the hypothesis. The remaining part of this section will be a discussion on the outcome of Experiment 1, as well as limitations in both the approach and in the data.

Different feature combinations

Different structural features turned out to produce substantially different results. On the one hand, both PDM and surface atoms produced promising results, and on the other hand, a feature as Phi/Psi did not. However, enzymes which fold to similar native states will have similar torsion angles, even if their residues differ. Furthermore, with only 11 basins, even angles which differ may end up in the same basin if it is the closest one, and in which case it should not be surprising that this feature did not perform as well.

All structural features which were hand-crafted were done so with certain limitations and restrictions. When calculating the PDM feature, a different distance threshold than 8Å could have been set. Regarding surface atoms, there are other, possibly more accurate ways of defining the surface of an enzyme. Nevertheless, even with these restrictions on the features, they produce better results than previously obtained. It remains to be seen how the scores could be improved further with additional refining to the features.

Feature set	Model	Training score	Test score
Sequential	Random forest	0.824	0.356
	SVR	0.640	0.407
PDM	Random forest	0.835	0.283
	SVR	0.866	0.387
Surface atoms	Random forest	0.749	0.270
	SVR	0.244	0.159
Sequential + PDM	Random forest	0.873	0.401
	SVR	0.883	0.441
Sequential + surface atoms	Random forest	0.875	0.474
	SVR	0.671	0.480

Table 5.1: The training and test scores from running random forest and SVR on the four best-performing feature sets. As can be seen, some overfitting occurs in most of the combinations.

Limitations in the approach

Due to a constrained time limit, certain limitations had to be done to the approach when calculating the features and running the models. As mentioned above, for PDM a distance threshold was set and if two residues were further apart than this distance, they were not considered to be interacting. It is possible there are better thresholds than the one that was set. Similarly for the rest of the features, there might be better approaches to how the features should be represented in a feature set. However, as scores improved, it is still an indication of what features seem more promising than others.

Limitations in the data

The majority of the enzymes in the dataset were predicted structures and only a minority were experimentally determined structures. This introduces noise in the calculated features. First of all, it would be a significant improvement to have more experimental structures as these would vastly improve the quality of the data, nevertheless, the number of known structures is limited and especially so if T_{opt} is involved. Second, each individual enzyme has not been scrutinized in detail which is relevant for future work in this area (Chapter 6). Both the structure should be carefully studied, as well as the corresponding T_{opt}. When the T_{opt} of an enzyme was determined, it is possible that only a limited number of temperatures were tested, and the one where the enzyme was the most effective was chosen as T_{opt}. Nonetheless, there might be a different T_{opt} which works better than the one reported, in which case each T_{opt} would need to be investigated. In short, there is uncertainty about

the T_{opt} values used for training and testing.

5.2 Experiment 2

Experiment 2 was created with a new training and test set, where enzymes were placed in the test set if they met a criteria of *similarity* in regard to their sequence and structure, yet had different T_{opt} values. The motivation behind Experiment 2 was to analyze whether the structural features were useful in discriminating between similar enzymes. It became meaningful to analyze what separates the enzymes to cause them to operate at different T_{opt} values, yet be similar. How sequence similarity was calculated is given in more detail in Section 5.2.2.

Random forest and SVR were used for this experiment as they produced the best results in Experiment 1. The feature sets that were used were sequential features alone, and sequential features combined with PDM. PDM was used rather than surface atoms, since PDM is a feature which is well-established in bioinformatics, as can be seen by its use in fold recognition [23]. Below in Section 5.2.1 is a more detailed description of the background and motivation to Experiment 2.

5.2.1 Background

When Experiment 1 was finished, focus was shifted to study how the data was distributed over EC numbers and T_{opt}. Figures B.1-B.4 in Appendix B visualize how the data is distributed over different T_{opt}, over the seven top classes of EC numbers and over the top class and first subclass of EC numbers. The final plot is a heatmap which describes how the enzymes are distributed both over the top EC class and T_{opt}. By studying these plots, it became evident that there was a high bias towards temperatures in the range of 30-60 degrees and towards the third enzyme top class.

Additional plots were created which visualized how different enzymes with equal EC numbers have different T_{opt}. As mentioned in Section 2.1.3, different enzymes have equal EC number if they catalyze the same chemical reaction. Figures B.5-B.10 in Appendix B show for each EC number, how the enzymes have significantly different T_{opt}. From this pattern, it became meaningful to analyze how similar those enzymes are and what makes them different, since they catalyze the same reaction, yet at different temperatures.

Similarity between enzymes is often calculated by measuring the percentage identity between two enzymes' sequences [5, Chapter 7], called sequence alignment. The program for performing sequence alignment implemented a dynamic programming approach called the Needleman-Wunch algorithm [32]. If two sequences have a percentage identity of 0.25 or above, the enzymes are said to be *homologous* [5]. Homologous enzymes most likely have a common ancestor and are similar in both sequence, structure and function. However, they are not necessarily the same protein, as it is probable they have evolved differently. If two enzymes have a percentage identity of lower than 0.25, it becomes difficult to say if they have a common ancestor, or are

similar by chance [5].

By training on non-homologous enzyme and predicting T_{opt} for homologous enzymes the ambition was that the models would be able to capture and identify the features that separated the homologous enzymes and could account for the difference in T_{opt}. This, in turn, may lead to biological insights.

5.2.2 Setup

All EC numbers for which there were more than one enzyme were recorded. For each EC number, only the enzymes which had a greater difference in T_{opt} (at least 20 degrees) were relevant. The difference in T_{opt} was important to make it easier to discriminate them in the models and to produce more meaningful results. With this subset of enzymes, the ones with the highest and lowest temperature had their percentage identity calculated, in order to represent them as a "hot" and a "cold" pair. If they were homologous they were saved to a new test set. Moreover, if, for instance, two enzymes shared the lowest temperature, only the pair of hot and cold enzymes with the highest percentage identity was saved. Thus, each pair of such enzymes was saved to a new test set, and the other non-homologous enzymes were saved to the training set. In total, this gave a test set of 88 pairs of enzymes.

Both random forest and SVR were trained on the non-homologous enzymes and predicted the T_{opt} for each pair of homologous enzymes. The feature sets that were used were sequential + PDM features, as well as sequential feature alone. The quality of the predictions were measured in two steps. First, the R^2 score was calculated as before. Second, a new accuracy metric was constructed which checked if the models preserved the *temperature order* for each of the homologous pairs in the test set. Specifically, if the "colder" enzyme was in fact predicted to be the colder enzyme in the pair, and similarly for the "hotter" enzyme. The new metric, *order accuracy* measured the percentage for how often the temperature order was correct for each pair of enzymes.

5.2.3 Results

Table 5.2 summarizes the training score, test score and order accuracy for both feature sets and models. The R^2 test scores are lower than previously obtained, however, 83% of all pairs were predicted in the correct temperature order. Figures 5.4a-5.4b and 5.5a-5.5b display scatter plots for the predicted versus observed values, for sequential + PDM features and sequential features respectively. The pairs predicted on the main diagonal line, $y = x$, symbolizes those that were predicted having equal T_{opt}, in which case the models were not able to discriminate the "cold" and "hot" enzyme. The pairs predicted under the main diagonal, are pairs for which the models swapped the temperature order of the enzymes. Lastly, the pairs predicted above the upper line, $y = x + 20$ are pairs for which the model correctly predicted the order of the enzyme, as well as keeping a temperature threshold of at least 20 degrees, which was the criteria for calculating the percentage identity. From all plots, it is clear that the vast majority of the pairs were predicted in the correct

Feature set	Model	Training score	Test score	Order accuracy
Sequential + PDM	Random forest	0.849	0.149	0.705
	SVR	0.877	0.240	0.727
Sequential	Random forest	0.770	0.128	0.693
	SVR	0.713	0.232	0.830

Table 5.2: Training, test and order accuracy scores from running Experiment 2. Sequential features and sequential + PDM features were both used in the random forest and SVR models. The order accuracy describes how often each pair of enzymes' temperatures were predicted in the correct order.

direction. Moreover, more pairs were predicted with the correct threshold and the correct order, than pairs with the correct threshold but with the order reversed.

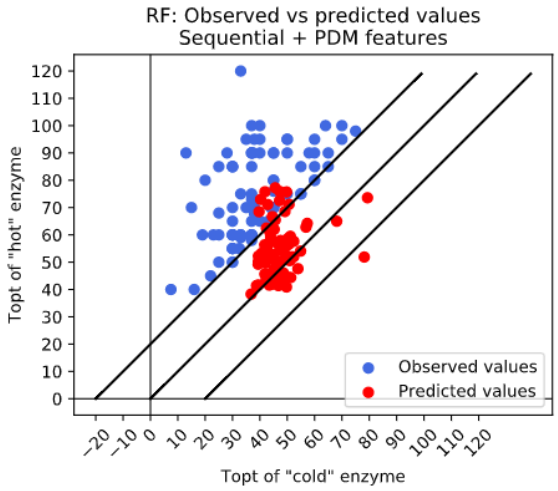
All models produced one outlier (a predicted pair below the lower line), except for SVR with sequential + PDM features which produced two. Overall, the sequential features performed better when used on its own in the models. Nevertheless, the feature importance score from sequential + PDM shows that the sequential features make up for 46% and the structural features 54% of the importance. This means there is still information the structural features contribute to the models, which the sequential features do not.

5.2.4 Discussion

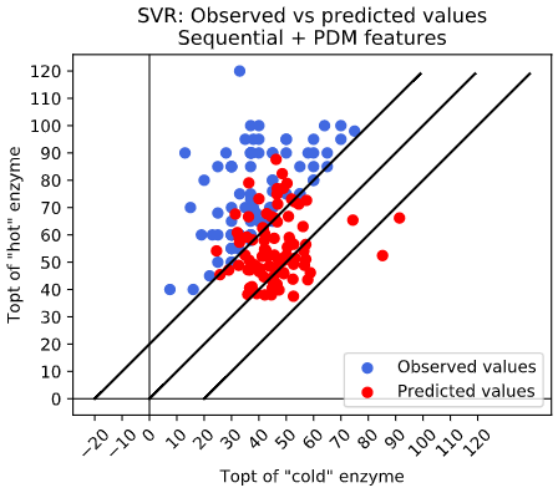
One expectation from Experiment 2 might be that, as the homologous enzymes have similar structures, the predictions will be random or equal as the models would not be able to discriminate between the enzymes and their T_{opt}. If the pairs would have been predicted having equal T_{opt}, there would have been a larger cluster along the main diagonal in Figures 5.4a-5.4b and 5.5a-5.5b. Nevertheless, as this is not the case, the feature sets and the model are in fact able to correctly separate the "hot" and "cold" enzyme, as predictions move to the upper left corner.

The R^2 test scores were lower compared to Experiment 1, but most likely this is because the homologous enzymes were not as well represented in the training set as before. The homologous enzymes are similar in sequence and will thus be similar in structure [42, 40]. As a consequence of similar structures, the structural features will be similar as well, and at times indistinguishable. However, as they have different T_{opt}, there is some property in the enzymes which accounts for this difference. Although lower R^2 test scores, the feature set is still able to distinguish a majority of the pairs in which one has a hotter temperature than the other.

As described at the beginning of Section 5.2, there is a large bias towards temperature in the range 30-60 degrees, see Figure B.4, which is most likely the reason many predictions cluster around this part of the plot. Oversampling techniques were considered but disregarded as there was not enough time to fully explore these ideas.

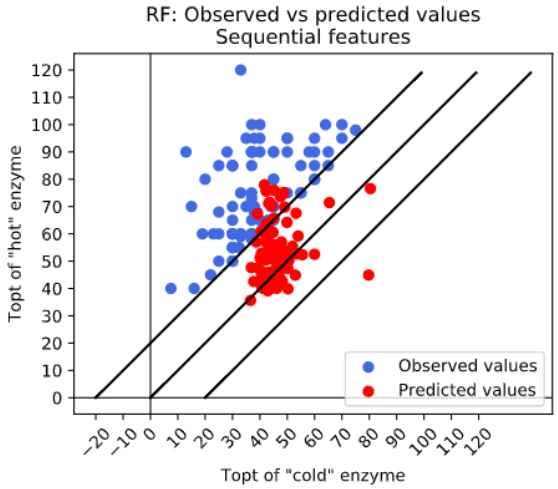


(a) Observed vs predicted values in a random forest model.

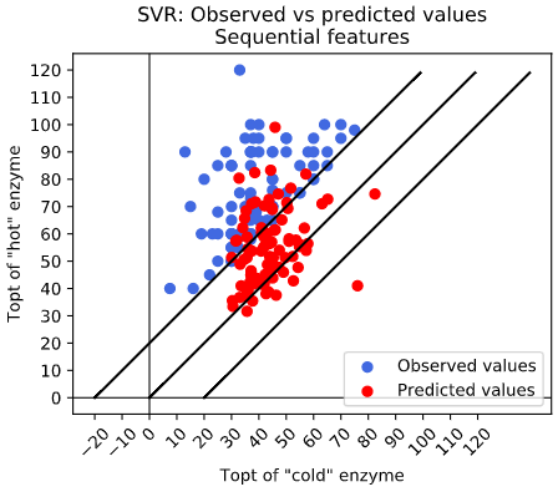


(b) Observed vs predicted values in an SVR model.

Figure 5.4: Observed vs predicted values for Experiment 2, using sequential + PDM features. The line $y = x$ marks where hot temperature equals cold temperature, and the lines above and below marks the threshold where the difference between the temperatures is at least 20 degrees.



(a) Observed vs predicted values in a random forest model.



(b) Observed vs predicted values in an SVR model.

Figure 5.5: Observed vs predicted values for Experiment 2, using only sequential features. The line $y = x$ marks where hot temperature equals cold temperature, and the lines above and below marks the threshold where the difference between the temperatures is at least 20 degrees.

Furthermore, as mentioned in Section 5.1.4, an improved quality and less bias in the data and T_{opt} is of high priority to continue this work and produce more reliable predictions.

6

Future work

The next priority for continuing this work is to study the data in more detail. To better understand each enzyme, each structure should be investigated along with corresponding literature. Additionally, each T_{opt} and the experiment which produced it should be analyzed. As mentioned in Section 5.1.4, it is possible a limited number of temperatures were tested to determine an enzyme's T_{opt}. The quality of such experiments are important to understand, in order to decide whether those enzymes ought to be included in a future dataset.

Further refining should be done to the features to increase their level of detail. For instance, for PDM it should be investigated if there are better thresholds than the one chosen for this project. A smaller threshold will allow for more interactions, while a larger threshold will allow for fewer. Regarding Phi/Psi there may be a better way to represent the torsion angles than by 11 predefined basins. Furthermore, for surface atoms there is most likely a more reliable way to represent this feature rather than a frequency count. Instead of counting the atomic groups, it might be more profitable to study the triplet itself the probe touched on the surface. However, all features rely on an increased knowledge of the structures themselves, in order to better understand how the structure is best represented.

One idea in this thesis was to work with predicted structures alone and experimental structures alone to see how much the features and model depended on the two types of structures. However, with a limited number of experimental structures this experiment was not carried through. It is possibly not feasible to be done in the near future, rather it is dependent on more structures being experimentally determined.

7

Conclusion

Through this thesis, the purpose has been to hand-craft structural features for enzymes and use these in machine learning models in order to study if T_{opt} predictions can improve from previous results. With the previous R2 test score of 0.4, the models in this thesis together with a new feature set are able to raise the scores to 0.48. Furthermore, in the scenario when there are pairs of similar enzymes but different T_{opt} values, one colder and one hotter, the models correctly predicts the temperature order of them 83% of the time. There are still properties in the enzymes which have not been represented as features, which most likely will improve the scores even further. Nevertheless, the original research question was to study if structural information hold information about T_{opt}, and if predictions can be improved. Both questions have been answered positively with this thesis.

Bibliography

[1] Badri Adhikari, Debswapna Bhattacharya, Renzhi Cao, and Jianlin Cheng. CONFOLD: Residue-residue contact-guided *ab initio* protein folding. *Proteins: Structure, Function, and Bioinformatics*, 83(8):1436–1449, 2015.

[2] Vickery L Arcus, Erica J Prentice, Joanne K Hobbs, Adrian J Mulholland, Marc W Van der Kamp, Christopher R Pudney, Emily J Parker, and Louis A Schipper. On The Temperature Dependence Of Enzyme-Catalyzed Rates. *Biochemistry*, 55(12):1681–1688, 2016.

[3] Mohammad Asadullah, Tomohisa Miyazawa, Shin-ichi Ito, Kimio Kunimori, and Keiichi Tomishige. Demonstration of real biomass gasification drastically promoted by effective catalyst. *Applied Catalysis A: General*, 246(1):103–116, 2003.

[4] Mariette Awad and Rahul Khanna. Support vector regression. In *Efficient Learning Machines*, pages 67–80. Springer, 2015.

[5] Jeremy M Berg, John L Tymoczko, and Lubert Stryer. *Biochemistry*. W. H. Freeman: New York, 2002.

[6] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, Talapady N Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The Protein Data Bank. *Nucleic Acids Research*, 28(1):235–242, 01 2000.

[7] Catrine L Berthold, Dörte Gocke, Martin D Wood, Finian J Leeper, Martina Pohl, and Gunter Schneider. Structure of the branched-chain keto acid decarboxylase (KdcA) from Lactococcus lactis provides insights into the structural basis for the chemoselective and enantioselective carboligation reaction. *Acta Crystallographica Section D: Biological Crystallography*, 63(12):1217–1224, 2007.

[8] Christopher M Bishop. *Pattern recognition and machine learning*. Springer, 2006.

[9] George D Chellapa and George D Rose. Reducing the dimensionality of the protein-folding search problem. *Protein Science*, 21(8):1231–1240, 2012.

[10] The UniProt Consortium. UniProt: a worldwide hub of protein knowledge.

Nucleic Acids Research, 47(D1):D506–D515, 11 2018.

[11] Athel Cornish-Bowden. Current IUBMB recommendations on enzyme nomenclature and kinetics. *Perspectives in Science*, 1(1-6):74–87, 2014.

[12] Srinivasan Damodaran and Kirk L Parkin. *Fennema's Food Chemistry*, volume 4. CRC Press: Boca Raton, FL, 2008.

[13] Yves Dehouck, Benjamin Folch, and Marianne Rooman. Revisiting the correlation between proteins' thermoresistance and organisms' thermophilicity. *Protein Engineering, Design & Selection*, 21(4):275–278, 2008.

[14] Nicolas Guex, Manuel C Peitsch, and Torsten Schwede. Automated comparative protein structure modeling with SWISS-MODEL and Swiss-PdbViewer: A historical perspective. *Electrophoresis*, 30(S1):S162–S173, 2009.

[15] Thiago S Guzella and Walmir M Caminhas. A review of machine learning approaches to Spam filtering. *Expert Systems with Applications*, 36(7):10206–10222, 2009.

[16] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements Of Statistical Learning: Data Mining, Inference, And Prediction*. Springer Science & Business Media, 2009.

[17] Michael E Himmel, Shi-You Ding, David K Johnson, William S Adney, Mark R Nimlos, John W Brady, and Thomas D Foust. Biomass Recalcitrance: Engineering Plants and Enzymes for Biofuels Production. *Science*, 315(5813):804–807, 2007.

[18] Joanne K Hobbs, Wanting Jiao, Ashley D Easter, Emily J Parker, Louis A Schipper, and Vickery L Arcus. Change In Heat Capacity For Enzyme Catalysis Determines Temperature Dependence Of Enzyme Catalyzed Rates. *ACS Chemical Biology*, 8(11):2388–2393, 2013.

[19] Liisa Holm and Chris Sander. Mapping The Protein Universe. *Science*, 273(5275):595–602, 1996.

[20] IUB IUPAC. IUPAC-IUB Joint Commission on Biochemical Nomenclature (JCBN). Nomenclature and symbolism for amino acids and peptides. Recommendations 1983. *Biochem J*, 219(2):345–373, 1984.

[21] Dmitry N Ivankov, Sergiy O Garbuzynskiy, Eric Alm, Kevin W Plaxco, David Baker, and Alexei V Finkelstein. Contact order revisited: Influence of protein size on the folding rate. *Protein Science*, 12(9):2057–2062, 2003.

[22] Lisa Jeske, Sandra Placzek, Ida Schomburg, Antje Chang, and Dietmar Schomburg. BRENDA in 2019: a European ELIXIR core data resource. *Nucleic Acids Research*, 47(D1):D542–D549, 2018.

[23] David T Jones, WR Taylort, and Janet M Thornton. A new approach to protein

fold recognition. *Nature*, 358(6381):86–89, 1992.

[24] Marcus Frederick Charles Ladd, Rex Alfred Palmer, and Rex Alfred Palmer. *Structure Determination By X-ray Crystallography*. Springer, 1985.

[25] Michael Levitt and Arieh Warshel. Computer simulation of protein folding. *Nature*, 253(5494):694–698, 1975.

[26] Gang Li, Kersten S Rabe, Jens Nielsen, and Martin KM Engqvist. Machine Learning Applied To Predicting Microorganism Growth Temperatures And Enzyme Catalytic Optima. *ACS synthetic biology*, 8(6):1411–1420, 2019.

[27] Gang Li, Jan Zrimec, Boyang Ji, Jun Geng, Johan Larsbrink, Aleksej Zelezniak, Jens Nielsen, and Martin KM Engqvist. Performance of regression models as a function of experiment noise. *arXiv preprint arXiv:1912.08141*, 2019.

[28] M Yu Lobanov, NS Bogatyreva, and OV Galzitskaya. Radius of Gyration as an Indicator of Protein Structure compactness. *Molecular Biology*, 42(4):623–628, 2008.

[29] Wissam Mehio, Graham JL Kemp, Paul Taylor, and Malcolm D Walkinshaw. Identification of protein binding surfaces using surface triplet propensities. *Bioinformatics*, 26(20):2549–2555, 2010.

[30] Eric Mjolsness and Dennis DeCoste. Machine Learning for Science: State of the Art and Future Prospects. *Science*, 293(5537):2051–2055, 2001.

[31] Richard J Morris, Anastassis Perrakis, and Victor S Lamzin. ARP/wARP's model-building algorithms. I. The main chain. *Acta Crystallographica Section D: Biological Crystallography*, 58(6):968–975, 2002.

[32] Saul B Needleman and Christian D Wunsch. A General Method Applicable to the Search for Similarities in the Amino Acid Sequence of Two Proteins. *Journal of molecular biology*, 48(3):443–453, 1970.

[33] Matti Parikka. Global biomass fuel resources. *Biomass and Bioenergy*, 27(6):613–620, 2004.

[34] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[35] Eric F Pettersen, Thomas D Goddard, Conrad C Huang, Gregory S Couch, Daniel M Greenblatt, Elaine C Meng, and Thomas E Ferrin. UCSF Chimera—A Visualization System for Exploratory Research and Analysis. *Journal of Computational Chemistry*, 25(13):1605–1612, 2004.

[36] Ursula Pieper, Benjamin M Webb, Guang Qiang Dong, Dina Schneidman-Duhovny, Hao Fan, Seung Joong Kim, Natalia Khuri, Yannick G Spill, Patrick Weinkam, Michal Hammel, et al. ModBase, a database of annotated comparative protein structure models and associated resources. *Nucleic Acids Research*, 42(D1):D336–D346, 2014.

[37] Kevin W Plaxco, Kim T Simons, and David Baker. Contact Order, Transition State Placement and the Refolding Rates of Single Domain Proteins. *Journal of Molecular Biology*, 277(4):985–994, 1998.

[38] Jay W Ponder and Frederic M Richards. Tertiary Templates for Proteins: Use Of Packing Criteria in the Enumeration of Allowed Sequences for Different Structural Classes. *Journal of Molecular Biology*, 193(4):775–791, 1987.

[39] G Philip Robertson, Stephen K Hamilton, Bradford L Barham, Bruce E Dale, R Cesar Izaurralde, Randall D Jackson, Douglas A Landis, Scott M Swinton, Kurt D Thelen, and James M Tiedje. Cellulosic biofuel contributions to a sustainable energy future: Choices and outcomes. *Science*, 356(6345):eaal2324, 2017.

[40] Carol A Rohl, Charlie EM Strauss, Kira MS Misura, and David Baker. Protein structure prediction using Rosetta. In *Methods in enzymology*, volume 383, pages 66–93. Elsevier, 2004.

[41] Alexander S Rose, Anthony R Bradley, Yana Valasatava, Jose M Duarte, Andreas Prlić, and Peter W Rose. NGL viewer: web-based molecular graphics for large complexes. *Bioinformatics*, 34(21):3755–3758, 2018.

[42] Chris Sander and Reinhard Schneider. Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins: Structure, Function, and Bioinformatics*, 9(1):56–68, 1991.

[43] Alex J Smola and Bernhard Schölkopf. A tutorial on support vector regression. *Statistics and computing*, 14(3):199–222, 2004.

[44] Martha M. Teeter. Water structure of a hydrophobic protein at atomic resolution: Pentagon rings of water molecules in crystals of crambin. *Proceedings of the National Academy of Sciences*, 81(19):6014–6018, 1984.

[45] Keith Tipton and Sinéad Boyce. History of the enzyme nomenclature system. *Bioinformatics*, 16(1):34–40, 2000.

[46] Jerry Tsai, Robin Taylor, Cyrus Chothia, and Mark Gerstein. The Packing Density in Proteins: Standard Radii and Volumes. *Journal of Molecular Biology*, 290(1):253–266, 1999.

[47] Pernilla Turner, Gashaw Mamo, and Eva Nordberg Karlsson. Potential and utilization of thermophiles and thermostable enzymes in biorefining. *Microbial Cell Factories*, 6(1):9, 2007.

[48] Andrew Waterhouse, Martino Bertoni, Stefan Bienert, Gabriel Studer, Gerardo Tauriello, Rafal Gumienny, Florian T Heer, Tjaart A P de Beer, Christine Rempfer, Lorenza Bordoli, et al. SWISS-MODEL: homology modelling of protein structures and complexes. *Nucleic Acids Research*, 46(W1):W296–W303, 2018.

[49] Carl J Yeoman, Yejun Han, Dylan Dodd, Charles M Schroeder, Roderick I Mackie, and Isaac KO Cann. Thermostable enzymes as biocatalysts in the biofuel industry. In *Advances in Applied Microbiology*, volume 70, pages 1–55. Elsevier, 2010.

A

Results from different feature combinations

A.1 Structural feature combinations

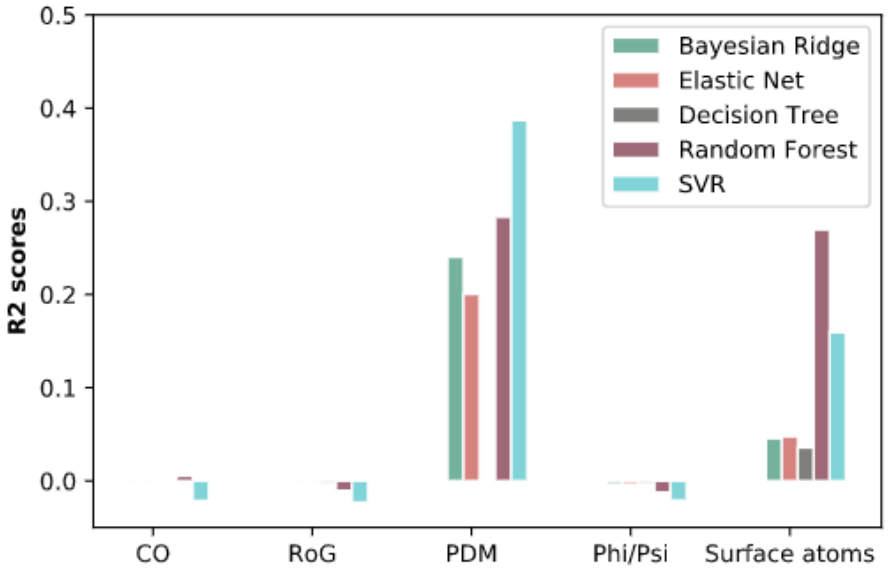


Figure A.1: Test scores from running all combinations of 1 feature. CO = contact order, RoG = radius of gyration, PDM = pairwise distance matrix (residue-residue interactions), Phi/psi = residue torsion angles, surface atoms = atomic groups on the surface.

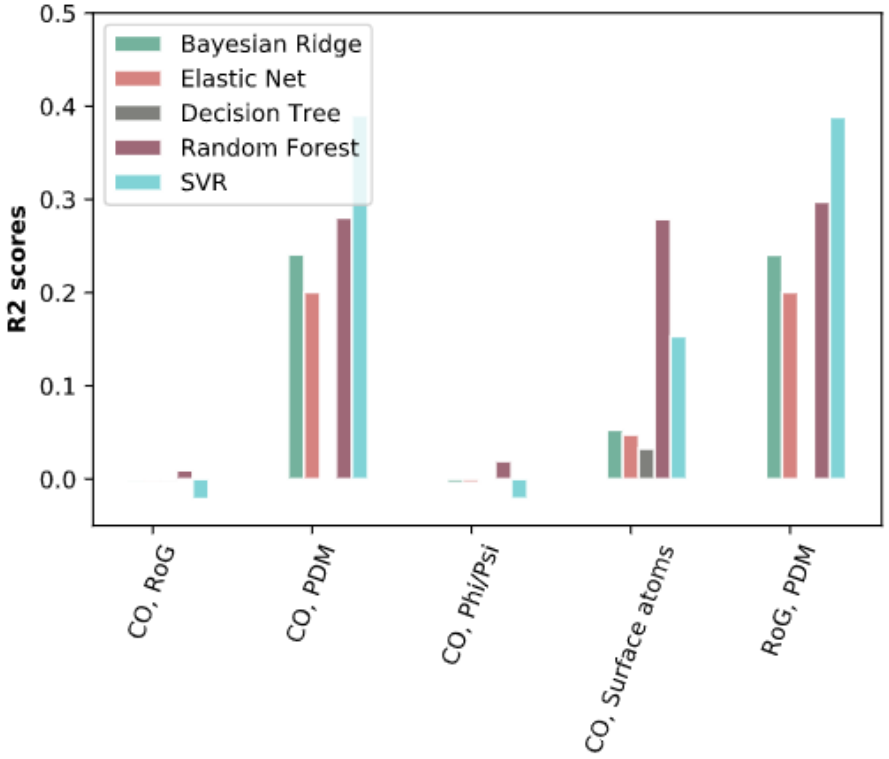


Figure A.2: Test scores from running all combinations of 2 features (1/2). CO = contact order, RoG = radius of gyration, PDM = pairwise distance matrix (residue-residue interactions), Phi/psi = residue torsion angles, surface atoms = atomic groups on the surface.

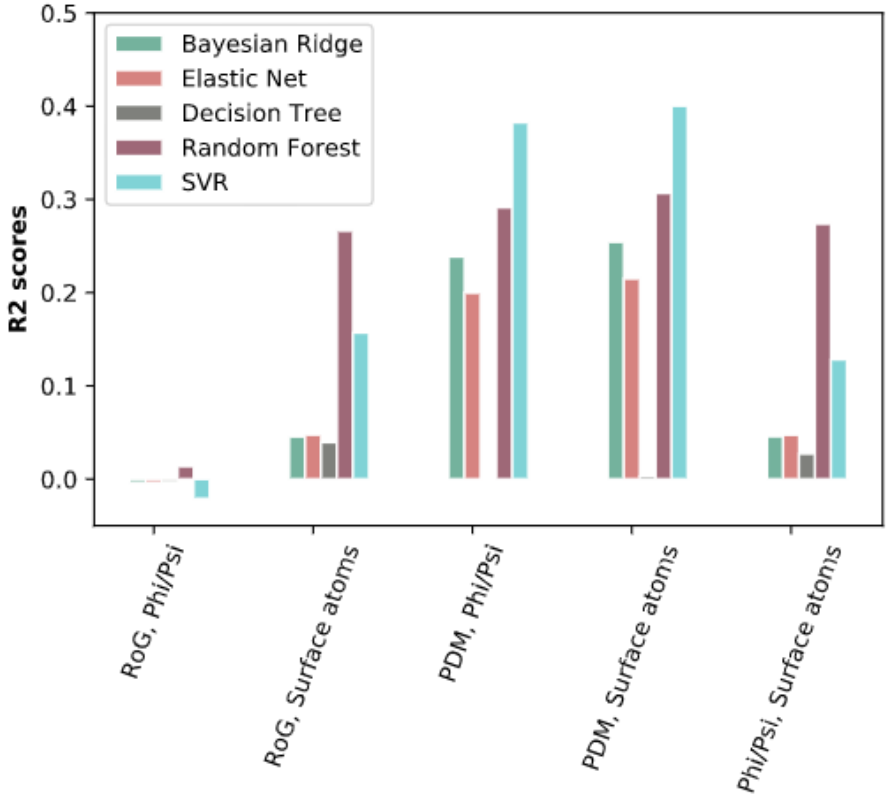


Figure A.3: Test scores from running all combinations of 2 features (2/2). CO = contact order, RoG = radius of gyration, PDM = pairwise distance matrix (residue-residue interactions), Phi/psi = residue torsion angles, surface atoms = atomic groups on the surface.

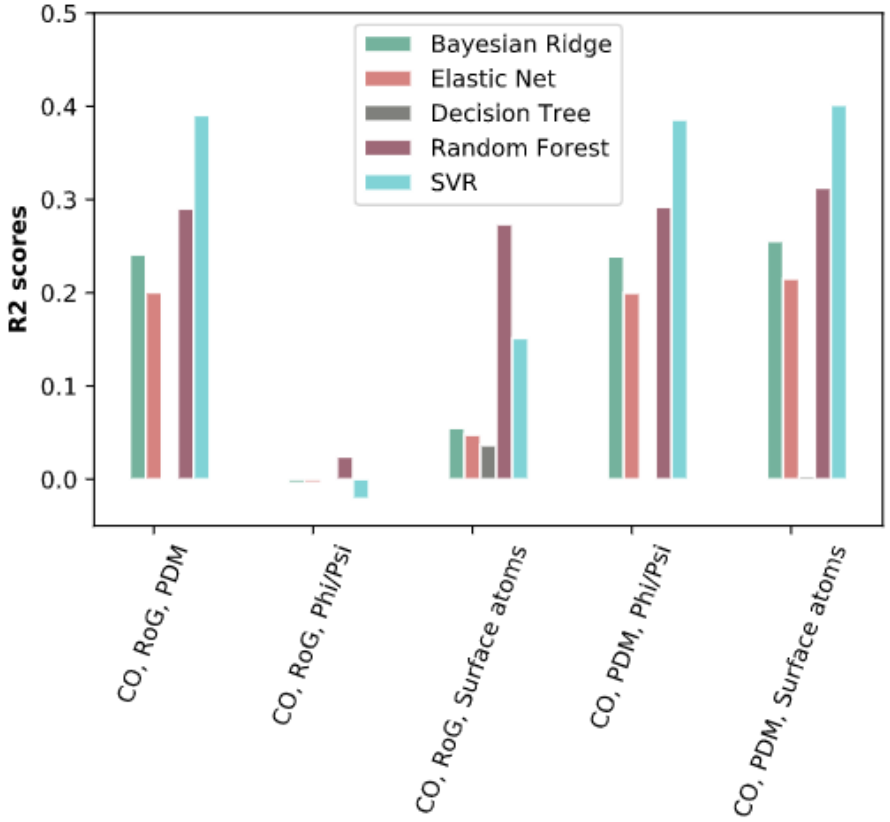


Figure A.4: Test scores from running all combinations of 3 features (1/2). CO = contact order, RoG = radius of gyration, PDM = pairwise distance matrix (residue-residue interactions), Phi/psi = residue torsion angles, surface atoms = atomic groups on the surface.

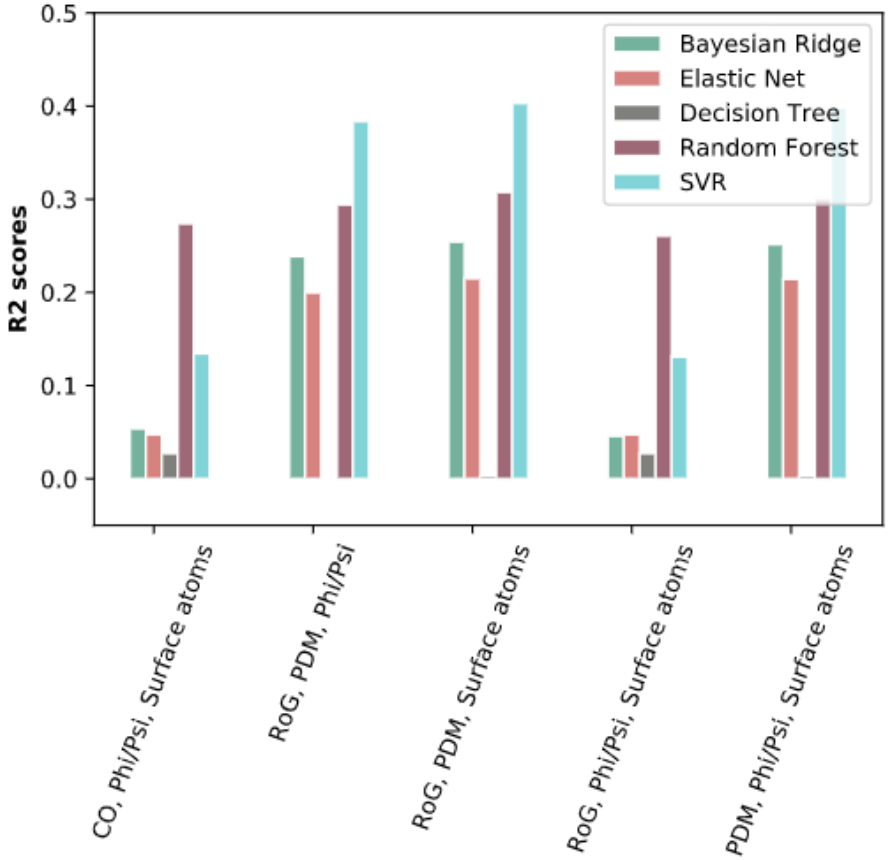


Figure A.5: Test scores from running all combinations of 3 features (2/2). CO = contact order, RoG = radius of gyration, PDM = pairwise distance matrix (residue-residue interactions), Phi/psi = residue torsion angles, surface atoms = atomic groups on the surface.

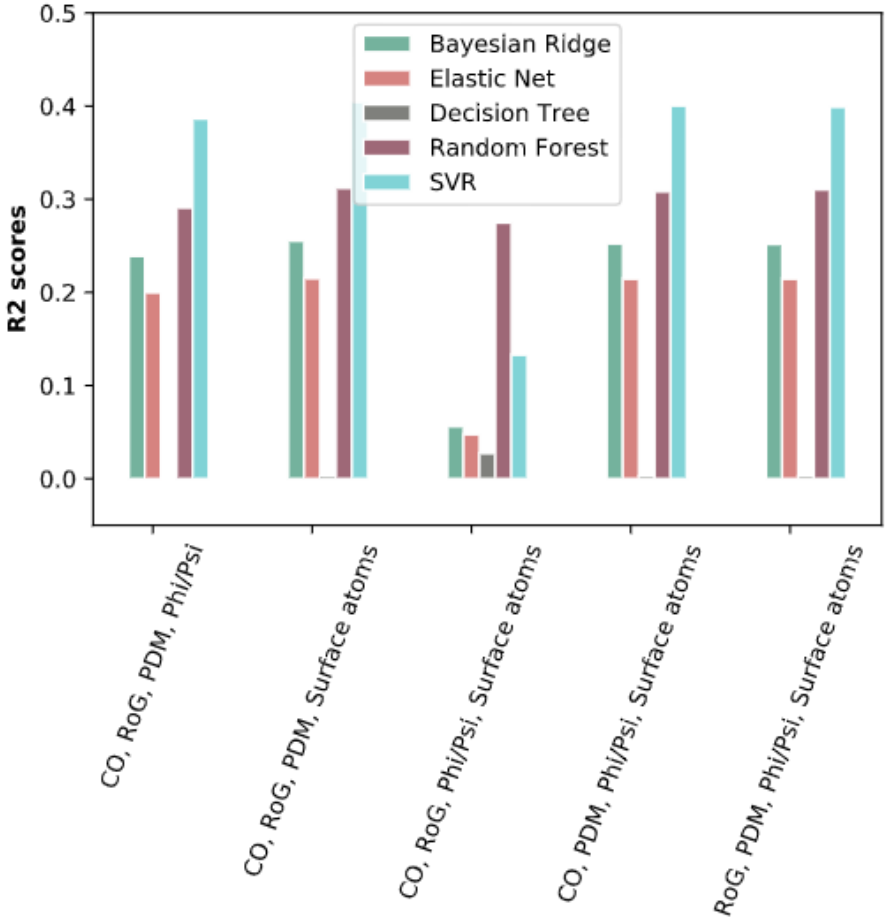


Figure A.6: Test scores from running all combinations of 4 features. CO = contact order, RoG = radius of gyration, PDM = pairwise distance matrix (residue-residue interactions), Phi/psi = residue torsion angles, surface atoms = atomic groups on the surface.

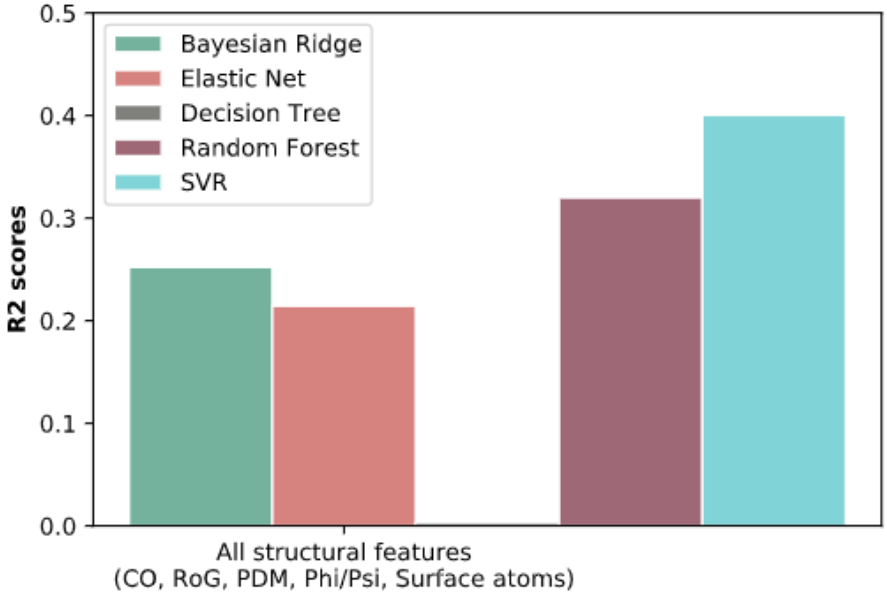


Figure A.7: Test scores from running all combinations of 5 features. CO = contact order, RoG = radius of gyration, PDM = pairwise distance matrix (residue-residue interactions), Phi/psi = residue torsion angles, surface atoms = atomic groups on the surface.

A.2 Structural and sequential feature combinations

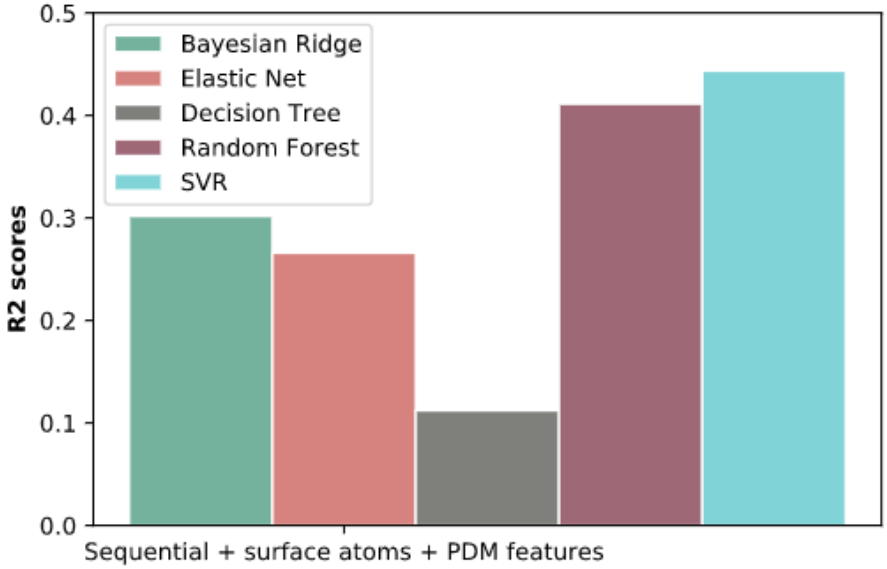


Figure A.8: Test scores from running the sequential features together with the PDM structural feature.

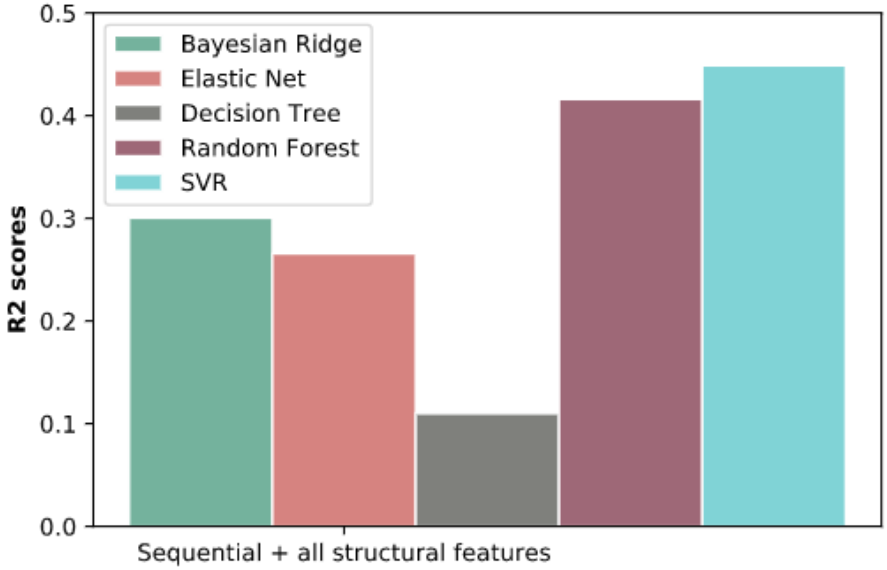


Figure A.9: Test scores from running the sequential features together with the all structural features.

B

Data visualization

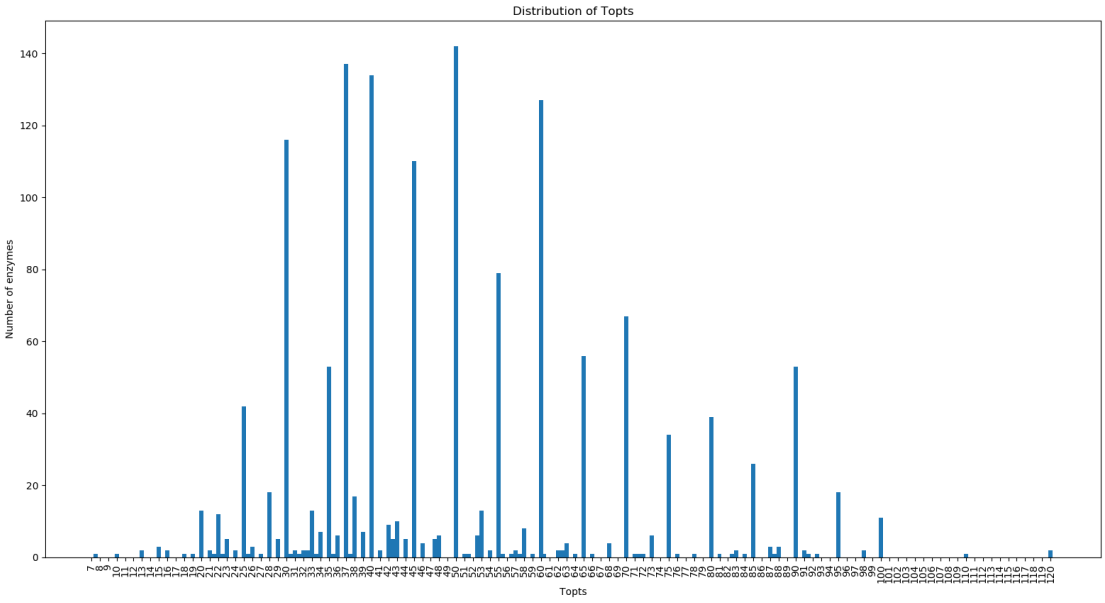


Figure B.1: Data distribution over different Topts. X axis specifies the temperature and the y axis how many enzymes are correlated with this temperature.

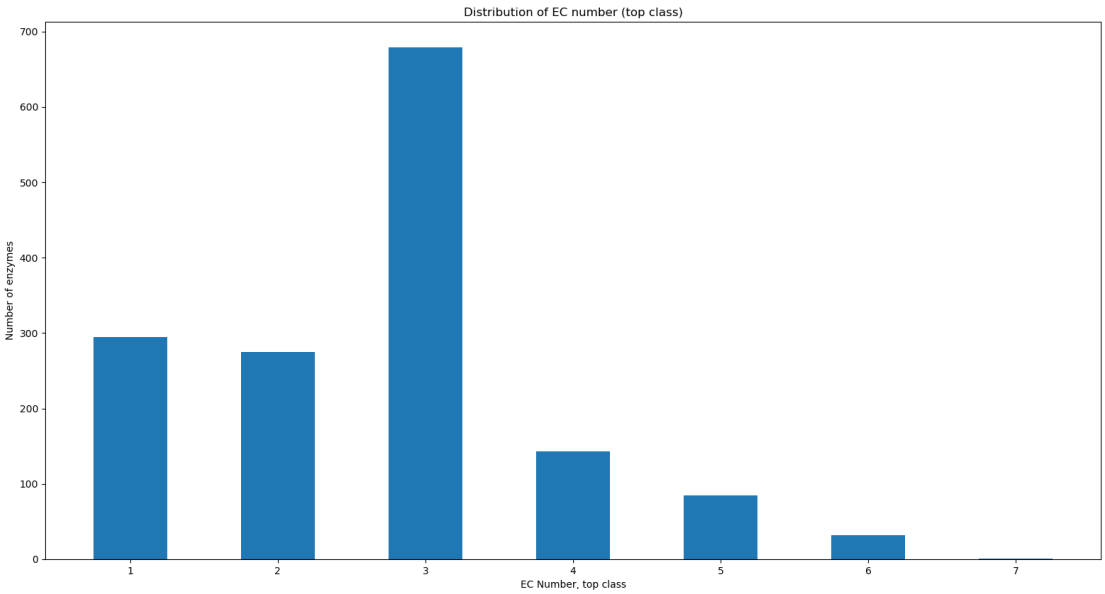


Figure B.2: Data distribution over the EC number top classes. X axis specifies the top class (1-7) and the y axis how many enzymes are correlated with this EC number.

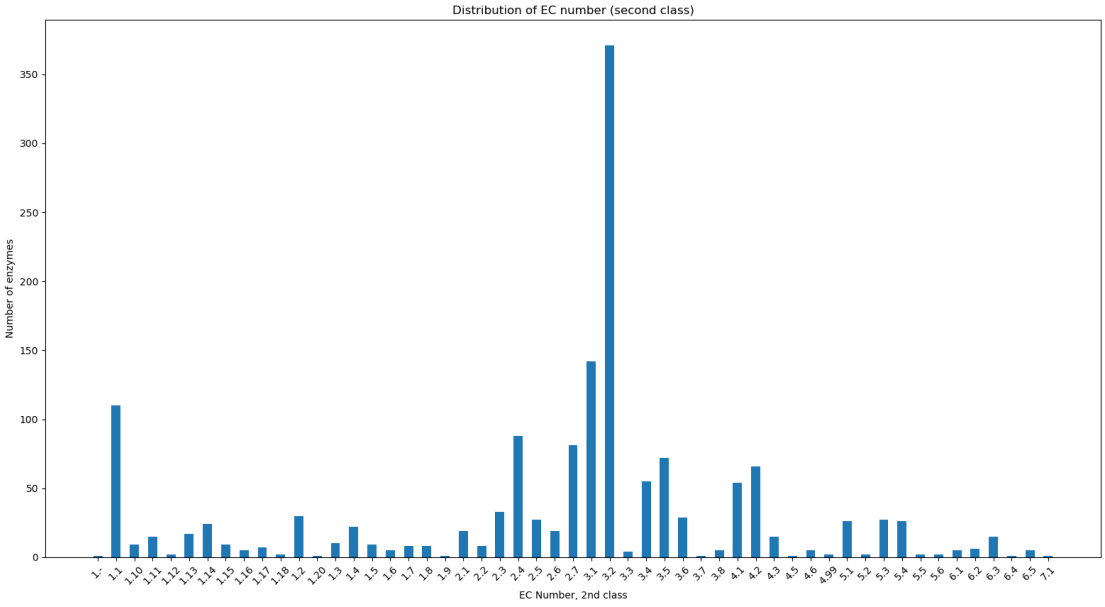


Figure B.3: Data distribution over the EC number top and second classes. X axis specifies the EC class and the y axis how many enzymes are correlated with this EC number.

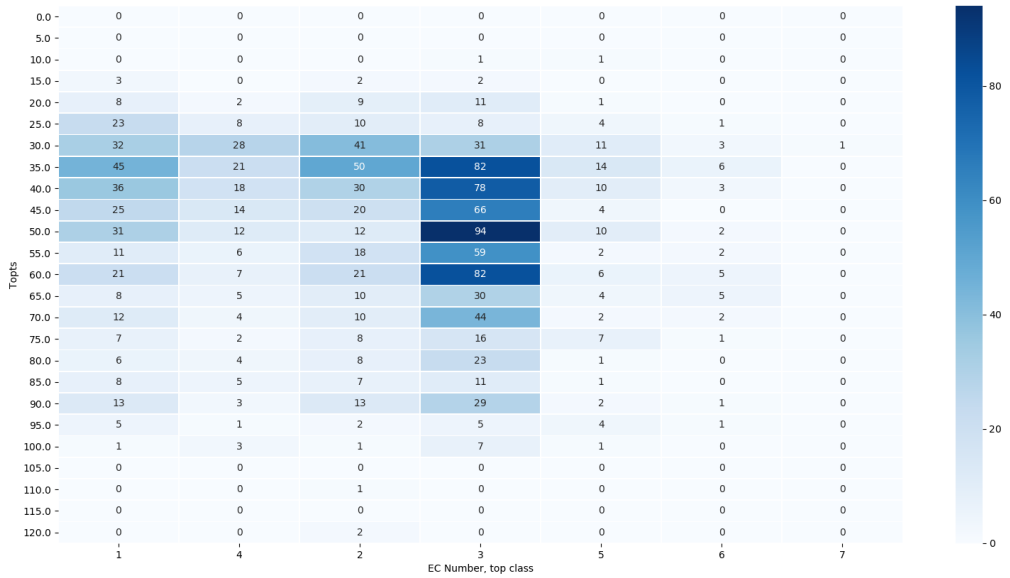


Figure B.4: Data distribution over Topt and EC EC number top classes. In each cell there is a count of how many enzymes of a certain EC top class has a certain temperature. The darker the cell, the more enzymes.

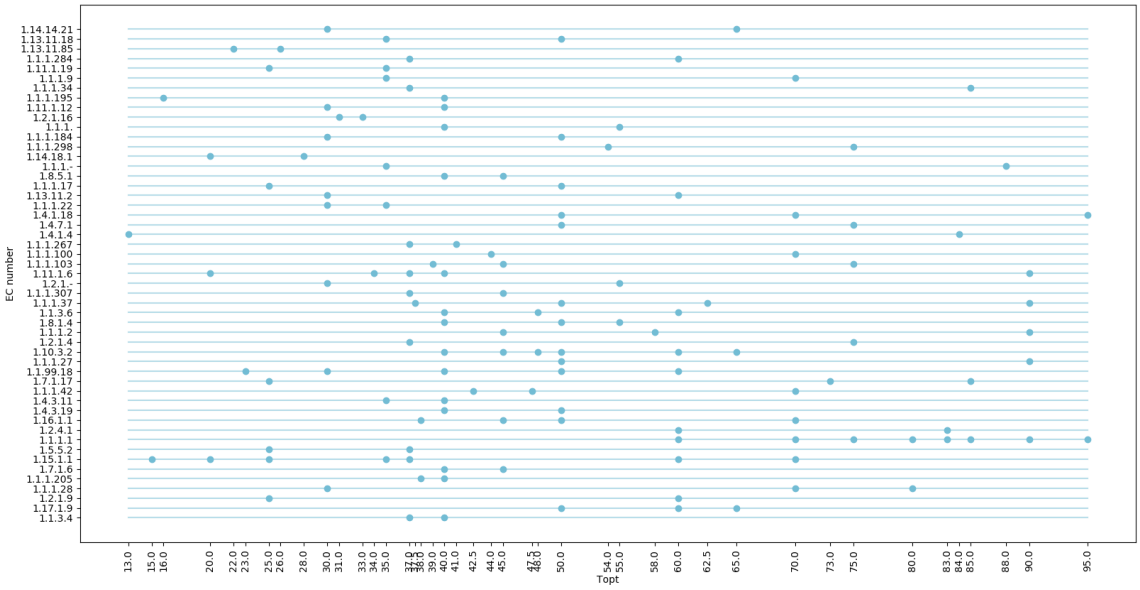


Figure B.5: Data distribution over enzymes that have the same EC number but different Topts. This visualization only shows relevant enzymes with EC number top class 1.

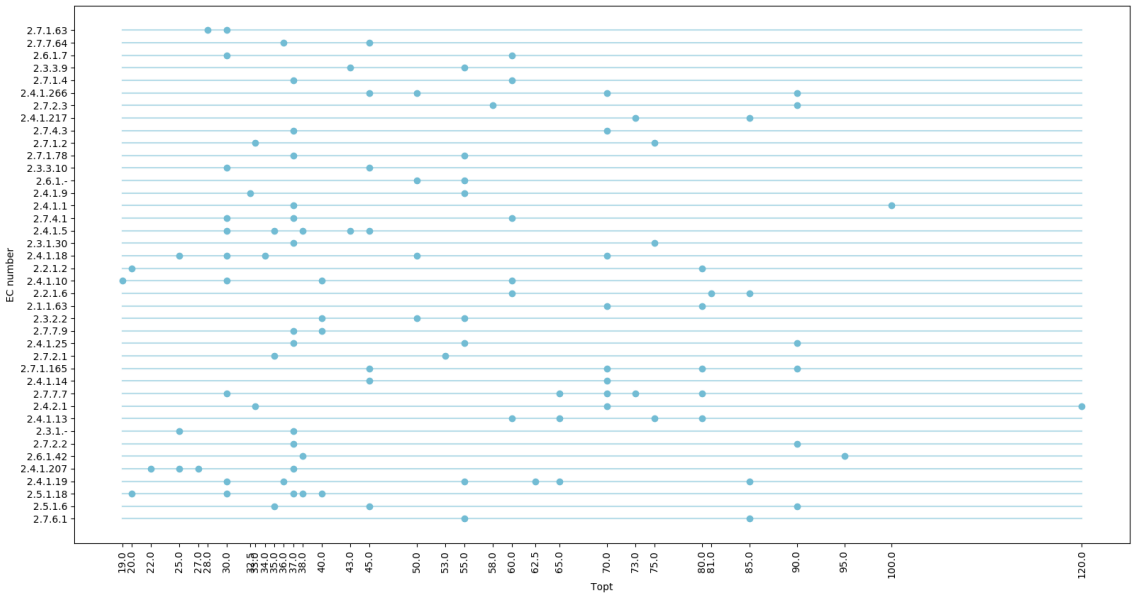


Figure B.6: Data distribution over enzymes that have the same EC number but different Topts. This visualization only shows relevant enzymes with EC number top class 2.

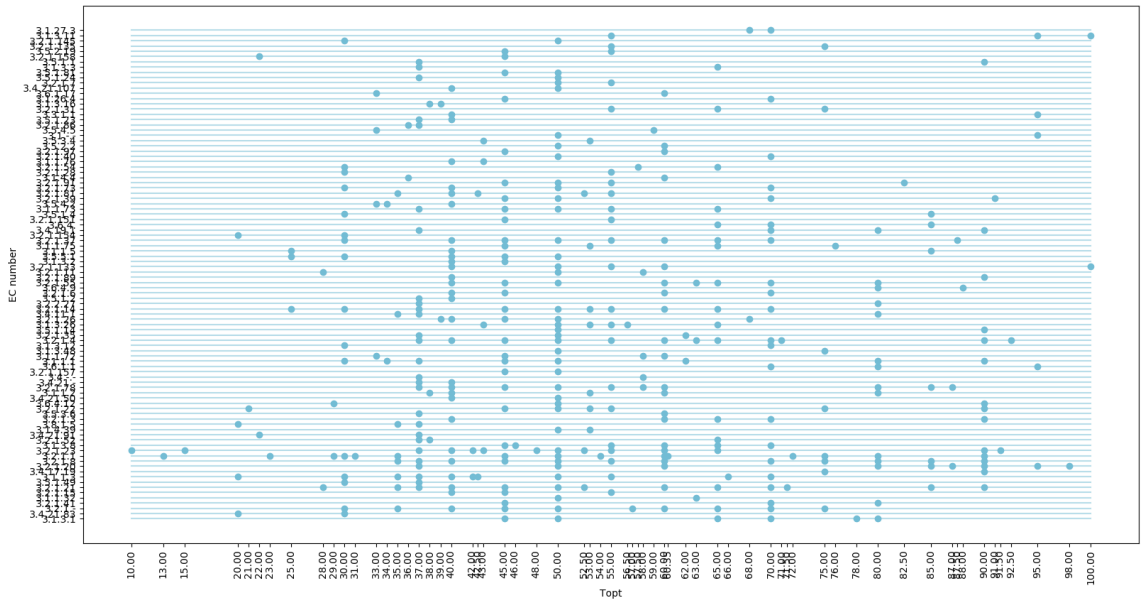


Figure B.7: Data distribution over enzymes that have the same EC number but different Topts. This visualization only shows relevant enzymes with EC number top class 3.

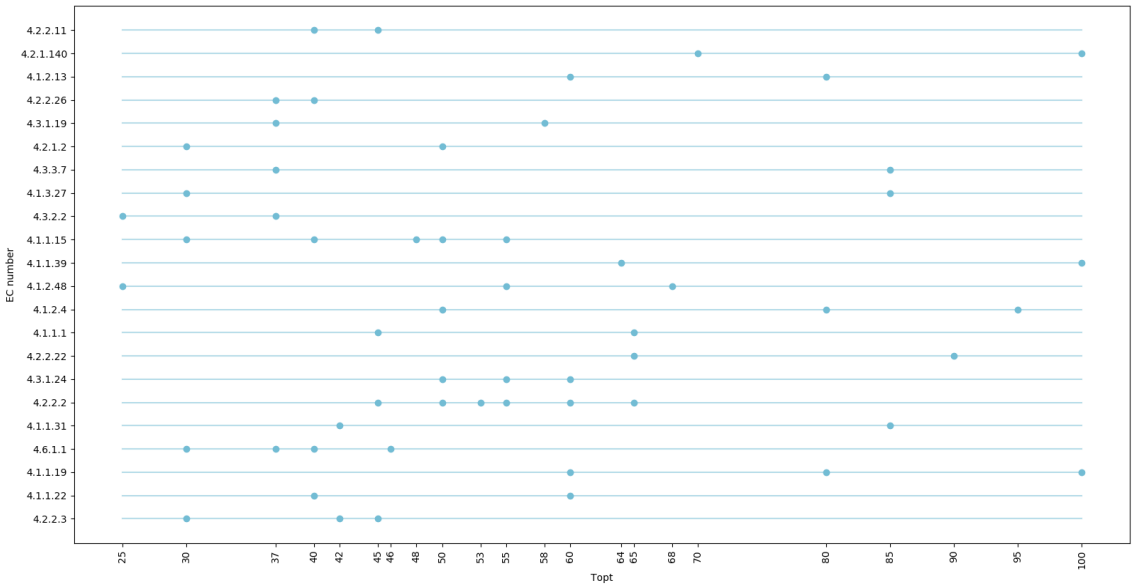


Figure B.8: Data distribution over enzymes that have the same EC number but different Topts. This visualization only shows relevant enzymes with EC number top class 4.

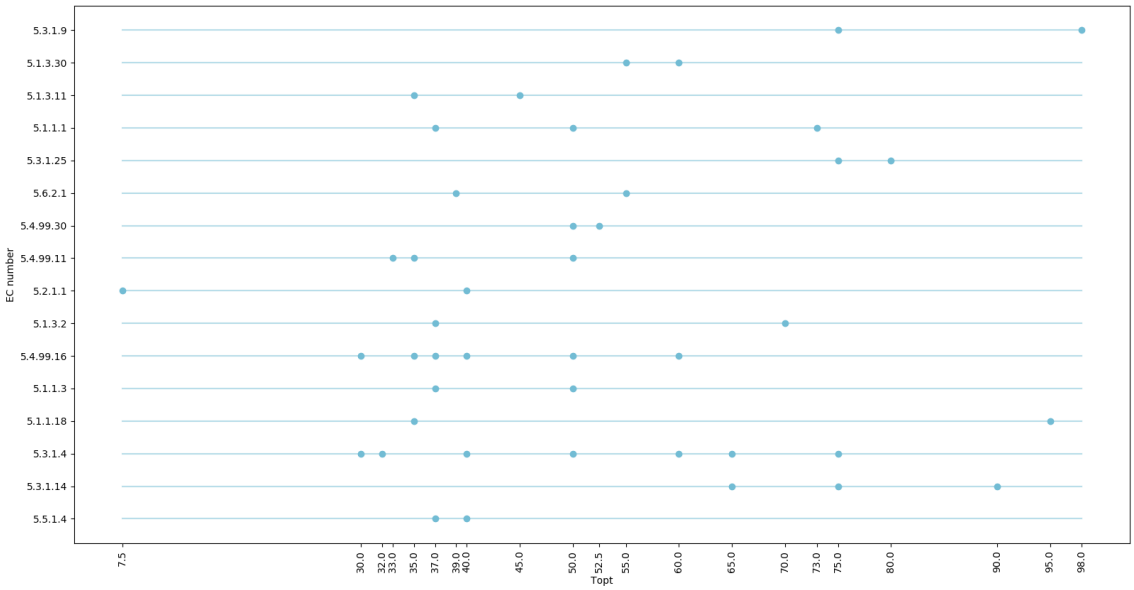


Figure B.9: Data distribution over enzymes that have the same EC number but different Topts. This visualization only shows relevant enzymes with EC number top class 5.

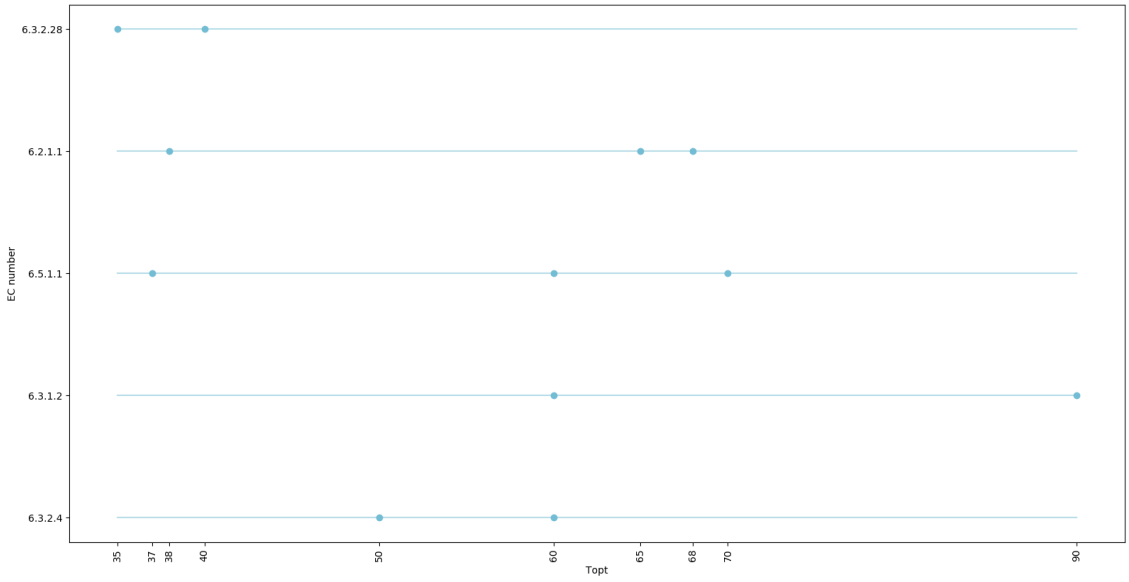


Figure B.10: Data distribution over enzymes that have the same EC number but different Topts. This visualization only shows relevant enzymes with EC number top class 6.