FHDW
Fachhochschule der Wirtschaft

Bachelorthesis

A Deep Learning Approach for Predicting Pesticide Degradation Based on Enzyme Classes

Prüfer(in):

Prof. Dr. Thomas Ströder

Fethi Temiz

Verfasser(in):

Tobias Polley

100853

Gutenbergstr. 5

51469 Bergisch Gladbach

BFWC321B

Cyber Security

Eingereicht am:

Sunday 23rd June, 2024

Sperrvermerk

Diese Arbeit enthält vertrauliche Informationen über die Firma Bayer AG. Die Weitergabe des Inhalts dieser Arbeit (auch in Auszügen) ist untersagt. Es dürfen keinerlei Kopien oder Abschriften - auch nicht in digitaler Form - angefertigt werden. Auch darf diese Arbeit nicht veröffentlicht werden und ist ausschließlich den Prüfern, Mitarbeitern der Verwaltung und Mitgliedern des Prüfungsausschusses sowie auf Nachfrage einer Evaluierungskommission zugänglich zu machen. Personen, die Einsicht in diese Arbeit erhalten, verpflichten sich, über die Inhalte dieser Arbeit und all ihren Anhängen keine Informationen, die die Firma Bayer AG betreffen, gegenüber Dritten preiszugeben. Ausnahmen bedürfen der schriftlichen Genehmigung der Firma Bayer AG und des Verfassers.

Die Arbeit oder Teile davon dürfen von der FHDW einer Plagiatsprüfung durch einen Plagiatsoftware-Anbieter unterzogen werden. Der Sperrvermerk ist somit im Fall einer Plagiatsprüfung nicht wirksam.

Contents

List of Figures

List of Tables

Code Listings

1 Introduction

1.1 Motivation

In recent years, the prediction of pesticide degradation has gained significant attention due to the environmental and health impacts of pesticide residues. Traditional experimental methods for determining the degradation pathways and rates of pesticides are labor-intensive and time-consuming. Consequently, there is a growing need for computational methods that can efficiently and accurately predict the degradation behavior of pesticides. One promising approach involves leveraging the capabilities of deep learning to predict enzyme classes responsible for pesticide degradation based on their interaction with specific enzyme binding sites. By combining the prediction of the active binding sites of enzymes with their corresponding protein sequences, it is possible to develop a model that can accurately predict the enzyme class.

This need is particularly pronounced at Bayer Crop Science, where the efficient and accurate prediction of pesticide degradation is crucial for developing environmentally friendly and safe agricultural products. The implementation of advanced computational methods, such as deep learning, can significantly enhance Bayer Crop Science's ability to predict and manage the environmental impact of their pesticide products, ensuring compliance with regulatory standards and promoting sustainable agricultural practices.

1.2 Problem Statement

Despite advancements in bioinformatics and computational biology, predicting enzyme classes is still fraught with uncertainties. Traditional methods rely heavily on experimental data, which can be resource-intensive and time-consuming. Moreover, the vast diversity of enzyme functions and their complex interactions with various substrates add layers of difficulty to accurate predictions. Thus, there is a pressing need for computational tools that leverage modern machine learning techniques to

enhance the prediction accuracy of enzyme-related pesticide degradation. Moreover, there are several models available for predicting enzyme classes based on sequence data, but there is still room for improvement in terms of performance. Therefore, this study aims to address this gap by developing a deep learning model that can predict enzyme classes responsible for pesticide degradation based on their interaction with specific enzyme binding sites.

1.3 Purpose and Research Question

This thesis aims to develop a deep learning model to predict the degradation of pesticides based on enzyme classes. The core research question guiding this study is: "How can deep learning be applied to predict pesticide degradation pathways based on enzyme class data?" By addressing this question, the study seeks to contribute to the fields of computational biology and environmental science, providing a tool that can aid in the rapid assessment of pesticide biodegradation potential. In addition, this research aims to outperform existing models in predicting enzyme classes responsible for pesticide degradation, thereby enhancing the accuracy of enzyme classification predictions.

1.4 Structure of the Thesis

This thesis is structured into five chapters, each addressing different aspects of the research and providing a comprehensive overview of the study. The first chapter sets the stage for the entire thesis. It begins by outlining the motivation behind the research, highlighting the environmental concerns related to pesticide use and the need for effective degradation prediction methods. The problem statement section identifies the challenges associated with predicting enzyme-mediated pesticide degradation. The purpose and research question section defines the main objective of the study, which is to develop a deep learning model to predict pesticide degradation based on enzyme classes. Finally, this chapter provides an overview of the structure of the thesis.

The literature review chapter delves into existing research and foundational theories relevant to the study. It covers enzymatic mechanisms involved in pesticide breakdown, offering insights into how enzymes facilitate the degradation process. Additionally, it explores the application of deep learning techniques in environmental science, emphasizing their potential to enhance predictive accuracy. The chapter also includes sections on the principles of enzymology, detailing enzyme classification, function, and their role in biodegradation, as well as the fundamentals of deep learning, including introductions to ligand-binding prediction and model evaluation techniques.

The methodology chapter provides a detailed description of the research design and procedures followed in this study. It begins with data collection, specifying the sources and preprocessing steps to prepare the dataset for analysis. The feature engineering section discusses how relevant features were extracted from the data to improve model performance. The chapter then explains the model development process, including the architecture of the deep learning model, the training process, and the techniques used for model evaluation to ensure its reliability and accuracy.

The results chapter presents the outcomes of the research. It begins with an evaluation of the model's performance, highlighting key metrics and the effectiveness of the model in predicting pesticide degradation. A comparative analysis with existing models is included to demonstrate the improvements and advantages of the developed model. The chapter also interprets the model predictions, offering insights into the practical implications of the findings and how they can be applied in real-world scenarios.

The discussion chapter summarizes the key findings of the research, reflecting on the significance and impact of the results. It discusses the strengths and limitations of the study, acknowledging areas where the model performed well and identifying potential areas for improvement. The chapter concludes with an overview of the contributions to the field, highlighting the novelty and practical applications of the research. Additionally, it provides recommendations for future work, suggesting directions for further research to build on the findings of this study.

2 Literature Review

2.1 Enzymatic Mechanisms Involved in Pesticide Breakdown

The breakdown of pesticides in the environment is a complex process involving various mechanisms, primarily driven by microbial enzymes. These enzymes catalyze reactions that convert toxic pesticide compounds into less harmful substances, facilitating their removal from the environment. This section explores the key enzymatic mechanisms involved in pesticide degradation, focusing on hydrolytic, oxidative, and reductive enzymes.

Microbial enzymes play a pivotal role in the biodegradation of soil contaminants, including pesticides. They can be categorized based on the reactions they catalyze:

Hydrolytic Enzymes: Hydrolytic enzymes, such as esterases and amidases, catalyze the cleavage of ester and amide bonds in pesticide molecules. This hydrolysis results in the formation of smaller, more water-soluble compounds that are easier to further degrade and eliminate. For example, microbial esterases can hydrolyze organophosphate insecticides, significantly accelerating their breakdown.

Oxidative Enzymes: Oxidative enzymes, such as cytochrome P450 monooxygenases, introduce oxygen atoms into the pesticide molecules, increasing their solubility and reactivity. This oxidation process often converts the pesticides into less harmful substances or intermediates that can be further degraded by other enzymes. The cytochrome P450 enzymes are particularly versatile, capable of metabolizing a wide range of xenobiotics, including pesticides.

Reductive Enzymes: Reductive enzymes, including reductases, catalyze the reduction of pesticides, often by adding electrons and hydrogen atoms to the molecules. This reduction can break down complex structures and facilitate the conversion of pesticides into simpler, less toxic forms. Reductive dehalogenases, for instance, play a significant role in the degradation of halogenated organic compounds.

Incorporating microbial enzymes into bioremediation strategies can significantly enhance the degradation of pesticides in contaminated soils. This approach leverages the natural capabilities of microbes to detoxify pollutants through enzymatic reactions. According to a review on the function of microbial enzymes in breaking down soil contaminated with pesticides, these enzymes are highly effective in transforming and mineralizing pesticides, thus reducing their environmental impact. [**singhMicrobialDegradationOrganophosphorus2006**]

Another study highlights the advancements and applications of microbial enzymes in biodegradation processes. This review emphasizes the critical role of enzymes in the degradation pathways of various pesticides and discusses the potential for engineered enzymes to improve bioremediation efficiency. [**chiaFunctionMicrobialEnzymes2024**]

2.2 Deep Learning Techniques in Environmental Science

Deep learning has emerged as a powerful tool in environmental science, offering advanced methods for predicting and understanding complex biochemical processes. In the context of pesticide degradation, deep learning models can analyze vast amounts of biochemical data to predict enzyme interactions and degradation pathways. There are several deep learning architectures that have been successfully applied to enzyme classification and prediction tasks, providing valuable insights into the mechanisms of pesticide breakdown.

For instance, the DEEPre model uses deep learning to predict enzyme commission (EC) numbers from raw sequence data. This model has shown significant improvements in prediction accuracy over traditional methods by utilizing convolutional and sequential feature extraction techniques. Such models can be crucial for predicting the biodegradation pathways of pesticides, enabling more accurate and efficient environmental risk assessments. [**liDEEPreSequencebasedEnzyme2017**]

Another example is the DeEPn model, which uses a deep neural network to classify enzymes into their functional classes, including all seven EC classes. This model has

demonstrated high precision and accuracy, making it a valuable tool for environmental scientists looking to understand and predict enzyme-mediated degradation processes. By accurately classifying enzymes, DeEPn facilitates the identification of potential candidates for bioremediation and other environmental applications. [**DeEPnDeepNeural**]

Despite the advances made by these models, there is still a need for new approaches to further improve the accuracy and applicability of pesticide degradation predictions. Traditional models often rely on pre-defined features and limited datasets, which can restrict their performance and generalizability. By contrast, my proposed approach leverages the deep learning tool p2rank to analyze the interactive parts of enzymes, focusing on the ligand-binding sites and the specific amino acids involved. [**krivakP2RankMachineLearning2018**] This method can potentially provide a more detailed and accurate prediction of enzyme classes responsible for pesticide degradation, enhancing our understanding of the biodegradation pathways and mechanisms involved.

3 Theoretical Background

3.1 Principles of Enzymology

Enzymology is the study of enzymes, which are biological catalysts that accelerate biochemical reactions in living organisms. These macromolecules are essential for various cellular processes, including metabolism, DNA replication, and signal transduction. The understanding of enzyme structure, function, and kinetics is crucial for developing applications in biotechnology, medicine, and environmental science. [**robinsonEnzymesPrinciplesBiotechnological2015**]

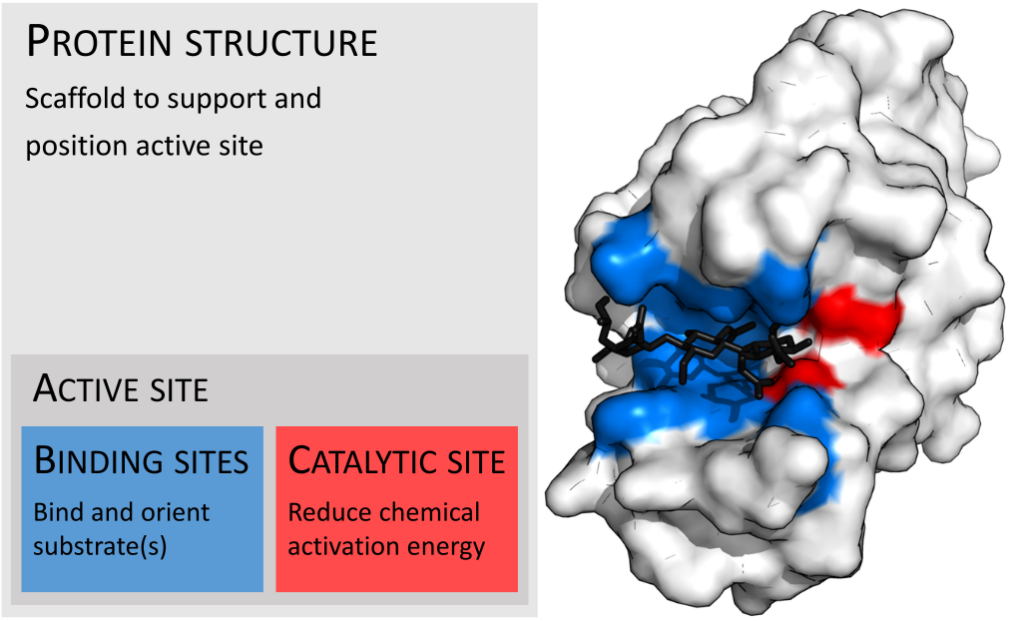
3.1.1 Enzyme Classification and Function

Enzymes are classified based on the reactions they catalyze, following a system established by the Enzyme Commission (EC). This classification system groups enzymes into six main classes, each with specific types of reactions they facilitate:

1. **Oxidoreductases:** These enzymes catalyze oxidation-reduction reactions, where the transfer of electrons occurs between molecules. Examples include dehydrogenases and oxidases.
2. **Transferases:** These enzymes transfer functional groups from one molecule to another. Examples include kinases, which transfer phosphate groups.
3. **Hydrolases:** These enzymes catalyze the hydrolysis of various bonds, including ester, glycosidic, peptide, and others. Examples include proteases and lipases.
4. **Lyases:** These enzymes add or remove groups to form double bonds, without hydrolysis or oxidation. Examples include decarboxylases and dehydratases.
5. **Isomerases:** These enzymes catalyze the rearrangement of atoms within a molecule, leading to isomerization. Examples include racemases and epimerases.
6. **Ligases:** These enzymes catalyze the joining of two molecules with the simultaneous hydrolysis of a diphosphate bond in ATP or a similar triphosphate. Examples include synthetases and carboxylases.

– Überleitung finden – The three-dimensional (3D) structure of enzymes is fundamental to their function. Enzymes are composed of one or more polypeptide chains that fold into specific shapes to form the active site. The active site is where substrate molecules bind and undergo a chemical reaction. The enzyme structure serves as a scaffold to support and correctly position the active site for optimal catalytic activity.

Figure 1: Organisation of enzyme structure and lysozyme example.



Source: Thomas Shafee, CC BY 4.0 via Wikimedia Commons

- **Protein Structure:** The overall structure of the enzyme provides the framework that supports and positions the active site. This structure is critical for the enzyme's stability and functionality. The enzyme's polypeptide chains fold into a unique 3D shape, creating a specific environment for the active site.
- **Active Site:** The active site includes two critical regions: binding sites and the catalytic site. The binding sites (highlighted in blue) are regions where substrates bind to the enzyme. These sites ensure that the substrates are properly oriented for the reaction. The catalytic site (highlighted in red) is the region where the chemical reaction occurs. The catalytic site often contains amino acids with specific functional groups that participate directly in the reaction, reducing the activation energy required for the reaction to proceed.

The precise arrangement of amino acids in the active site allows enzymes to be highly specific for their substrates, facilitating efficient catalysis. This specificity is a key feature that enables enzymes to perform their roles in various biochemical pathways with high precision.

A study by Veselovsky et al. (2001) emphasizes the importance of visualizing active site structures, even for enzymes with unknown 3D structures. By analyzing enzyme interactions with reversible competitive inhibitors and molding the substrate-binding region, researchers can predict the shape and dimensions of the active site. This approach has been validated by comparing it with known enzyme-inhibitor complexes, demonstrating its utility in understanding enzyme function and aiding in the search for new ligands.[**veselovskyApproachVisualizationActive2001**]

3.1.2 Role of Enzymes in Biodegradation

Enzymes play a crucial role in the biodegradation of pollutants, including pesticides. The process involves the breakdown of complex organic molecules into simpler, less toxic forms. This degradation is essential for reducing environmental pollution and mitigating the adverse effects of hazardous chemicals.

Hydrolytic Enzymes: Hydrolytic enzymes, such as esterases and amidases, catalyze the cleavage of ester and amide bonds in pesticide molecules. This hydrolysis results in the formation of smaller, more water-soluble compounds that are easier to further degrade and eliminate. For example, microbial esterases can hydrolyze organophosphate insecticides, significantly accelerating their breakdown. [**munneckeEnzymaticHydrolysisOrg**

Oxidative Enzymes: Oxidative enzymes, such as cytochrome P450 monooxygenases, introduce oxygen atoms into the pesticide molecules, increasing their solubility and reactivity. This oxidation process often converts the pesticides into less harmful substances or intermediates that can be further degraded by other enzymes. The cytochrome P450 enzymes are particularly versatile, capable of metabolizing a wide range of xenobiotics, including pesticides. [**belloTheoreticalApproachMechanism2000**]

Reductive Enzymes: Reductive enzymes, including reductases, catalyze the reduction of pesticides, often by adding electrons and hydrogen atoms to the molecules. This reduction can break down complex structures and facilitate the conversion of pesticides into simpler, less toxic forms. Reductive dehalogenases, for instance, play a significant role in the degradation of halogenated organic compounds.

The integration of enzymatic biodegradation with deep learning models can enhance the prediction and analysis of these processes. By using deep learning to analyze enzyme-substrate interactions and their corresponding (EC) classification, we can develop more accurate and efficient bioremediation strategies.

3.2 Fundamentals of Ligand Binding Site Prediction

As mentioned earlier, enzymes interact with substrates at specific binding sites, where the catalytic reactions occur. Predicting these ligand-binding sites is crucial for understanding the enzyme function. Several computational methods have been developed to predict ligand-binding sites from protein structures, including geometric, physicochemical, and machine learning-based approaches.

One approach is P2Rank, a machine learning-based tool designed for the rapid and accurate prediction of ligand binding sites from protein structures. It employs a combination of geometric and physicochemical descriptors to analyze protein structures and predict the locations of potential binding sites. P2Rank uses a random forest algorithm, an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes (classification) or mean prediction (regression) of the individual trees.

The tool focuses on the interactive parts of enzymes, particularly the ligand-binding sites and the specific amino acids involved. This detailed analysis allows for accurate predictions of enzyme classes and their associated degradation pathways. P2Rank's ability to quickly and accurately predict binding sites makes it a valuable tool for drug discovery and environmental bioremediation applications.

P2Rank leverages local chemical neighborhood features near the protein surface to infer potential binding sites for ligands. Here is an overview of the key steps involved in the P2Rank prediction process:

1. **Generation of Connolly Points**: Connolly Points are regularly spaced points generated on the protein's Connolly surface, representing the solvent-accessible surface area of the protein. These points are generated using a numerical algorithm that ensures even spacing, typically with a solvent radius of 1.6 Å.

2. **Calculation of Feature Descriptors**: Atomic Feature Vectors (AFVs) are calculated for each solvent-exposed heavy atom in the protein, describing various physico-chemical properties such as hydrophobicity, aromaticity, and more. These properties are projected onto the Connolly points using a distance-weighted approach, creating Connolly Feature Vectors (CFVs) for each point. The image shows Connolly Points (green dots) on the protein's surface, where each point is associated with a Connolly Feature Vector (CFV).

    a) Atomic Features: Features are inherited from the amino acid, including properties like hydrophobicity and polarity index. Additional features for AA atoms include H-Donor, H-Acceptor, and aromaticity.

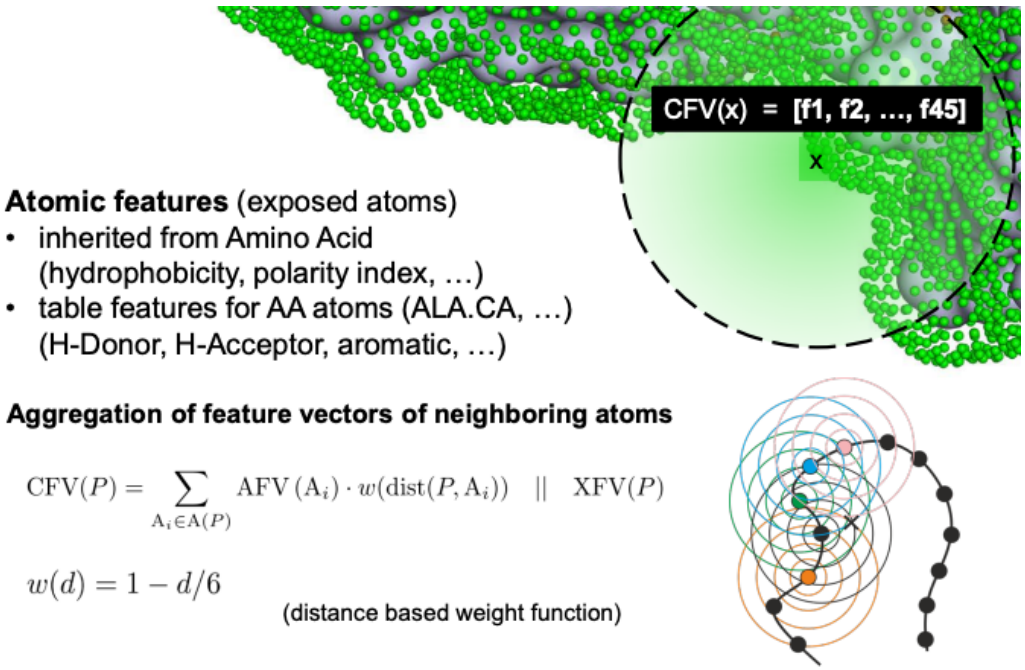
    b) Aggregation of Feature Vectors: The CFV for each Connolly point is calculated by aggregating the AFVs of neighboring atoms using a distance-based weight function $w(d) = 1 - d/6$.

3. **Ligandability Prediction**: A Random Forest classifier is used to predict the ligandability score for each Connolly point, indicating the likelihood that a point is part of a ligand-binding site.

4. **Clustering**: Connolly points with high ligandability scores are clustered using a single-linkage clustering method, representing potential binding pockets on the protein surface.

5. **Ranking**: Each predicted pocket is assigned a score based on the cumulative ligandability scores of its constituent points, helping prioritize the most likely binding sites for further analysis or docking studies.

Figure 2: Calculation of feature vectors for neighboring atoms in P2Rank.



Source: Radoslav Krivák and David Hoksza

P2Rank's approach can significantly enhance the accuracy of predicting enzyme-mediated degradation of pesticides by providing detailed insights into the binding interactions at the molecular level. This integration of deep learning and enzyme analysis forms a robust framework for developing bioremediation strategies and understanding the environmental fate of various pollutants. [**krivakP2RankMachineLearning2018**]

3.3 Introduction to Recurrent Neural Networks

Recurrent Neural Networks (RNNs) are a class of artificial neural networks designed to recognize patterns in sequences of data such as text, genomes, handwriting, and spoken words. Unlike traditional feedforward neural networks, RNNs have connections that form directed cycles, allowing information to persist. This makes them particularly powerful for tasks that involve sequential data, where the order of the data points matters.

Recurrent Neural Networks (RNNs) are designed to process sequences of data by maintaining a memory of previous inputs. This memory allows RNNs to make use of information from earlier in the sequence to influence the current processing

step, which is essential for understanding context in sequential data. The fundamental difference between RNNs and traditional neural networks is the presence of loops in the network that enable the persistence of information across time steps.

Recurrent Neural Networks (RNNs) are designed to process sequences of data by maintaining a memory of previous inputs. This memory allows RNNs to make use of information from earlier in the sequence to influence the current processing step, which is essential for understanding context in sequential data. The fundamental difference between RNNs and traditional neural networks is the presence of loops in the network that enable the persistence of information across time steps.

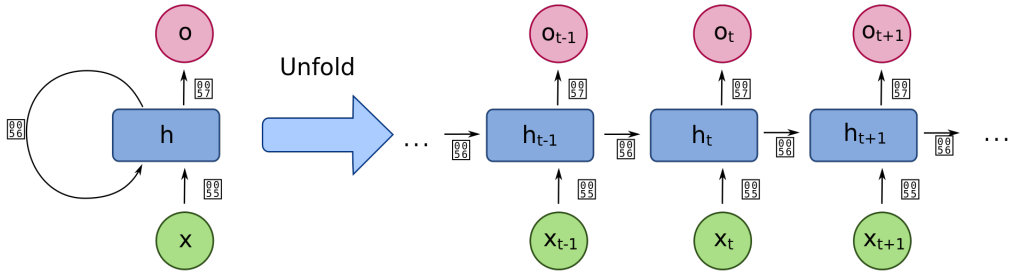
The basic structure of an RNN includes an input layer, a hidden layer with recurrent connections, and an output layer. At each time step, the hidden layer receives the input data and its own previous state, allowing it to retain and process information from previous steps in the sequence.

One of the key advancements in RNNs is the development of Long Short-Term Memory (LSTM) networks and Gated Recurrent Units (GRUs), which are designed to overcome the limitations of traditional RNNs, such as the vanishing gradient problem. These architectures use gating mechanisms to control the flow of information, making it easier to capture long-term dependencies in data.

In the context of bioinformatics, RNNs, particularly LSTMs and GRUs, are extensively used for sequence analysis tasks such as protein secondary structure prediction, gene expression analysis, and more. They are effective because they can handle the sequential nature of biological data and capture dependencies that span over long sequences. LSTM networks are a type of RNN that can learn long-term dependencies. They incorporate memory cells that can maintain their state over long periods. LSTMs have three main gates (input gate, forget gate, and output gate) that regulate the flow of information into and out of the memory cell, thus enabling the network to remember important information for longer durations. [**hochreiterLongShortTermMemory1997**]

The following image illustrates the basic structure of an RNN:

Figure 3: A diagram for a one-unit recurrent neural network (RNN).



Source: fdeloche, CC BY-SA 4.0 via Wikimedia Commons

1. Input Sequence (x): The green circles represent the input data at different time steps (x_{t-1}, x_t, x_{t+1}).

2. Hidden State (h): The blue rectangles represent the hidden state of the network. At each time step, the hidden state (h) is updated based on the current input and the previous hidden state (h_{t-1}, h_t, h_{t+1}).

3. Output Sequence (o): The pink circles represent the output of the network at each time step (o_{t-1}, o_t, o_{t+1}).

The recurrent connection (arrow looping back) in the hidden state allows information to persist across time steps, enabling the network to maintain context and capture dependencies in the sequence data.

In this study, RNNs are employed for predicting the enzyme class based on the amino acid sequences of a ligand binding site. The sequential nature of the amino acid sequences makes RNNs well-suited for this task, as they can capture the dependencies and patterns in the data that are crucial for predicting enzyme classes accurately. Especially for complex and long sequences, RNNs, particularly LSTMs, are effective in learning the underlying structure and relationships in the data.

3.4 Evaluation of Deep Learning Models

Evaluating deep learning models involves several metrics and techniques to ensure their accuracy and generalizability. This is essential not only for validating the model's performance but also for comparing it against other models. Using independent datasets for benchmarking is crucial to demonstrate the model's robustness and applicability to real-world scenarios. Several key metrics are commonly used to evaluate deep learning models:

1. **Accuracy:** The ratio of correctly predicted instances to the total instances. It provides a straightforward measure of performance but can be misleading if the data is imbalanced. [**AccuracyPrecision2024**]

2. **Precision:** The ratio of true positive predictions to the total predicted positives. Precision is crucial when the cost of false positives is high.

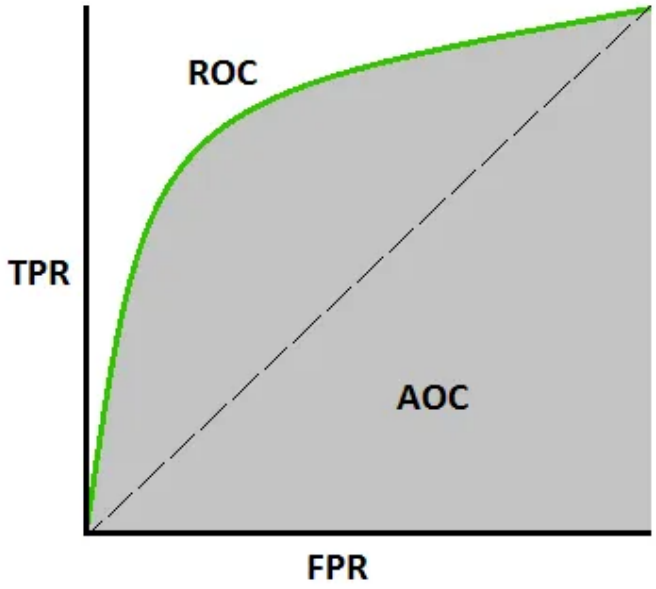
3. **Recall (Sensitivity):** The ratio of true positive predictions to the total actual positives. Recall is important when the cost of false negatives is high. [**PrecisionRecall2024**]

4. **F1-Score:** The harmonic mean of precision and recall, providing a single metric that balances both. It is useful when there is an uneven class distribution. [**Fscore2024**]

5. **ROC-AUC (Receiver Operating Characteristic - Area Under Curve):** A performance measurement for classification problems at various threshold settings. It tells how much the model is capable of distinguishing between classes. The image shows an example of an ROC-AUC curve. As the curve gets closer to the top-left corner, the model's performance improves. [**ReceiverOperatingCharacteristic2024**]

Using these metrics, the performance of deep learning models can be tested on independent datasets. This is critical for ensuring that the models are not just overfitting to the training data but can generalize well to new, unseen data. By applying the models to benchmark datasets, researches can objectively measure

Figure 4: AUC - ROC Curve



Source: Sarang Narkhede via towardsdatascience.com

and compare their performance. For instance, in the case of predicting enzyme-mediated pesticide degradation, using independent datasets ensures that the model's predictions are reliable and can be generalized across various enzyme and pesticide types.

4 Methodology

4.1 Data Collection

Data collection is a critical step in developing predictive models, as the quality and relevance of the data directly impact the model's performance. In this thesis, the data was collected from Uniprot, a comprehensive resource for protein sequence and functional information. The focus was on obtaining high-quality, reviewed entries with 3D structural data and catalytic properties to ensure the reliability and applicability of the data for predicting enzyme functions. [**uniprotconsortiumUniProtUniversalProtein202**

Uniprot, or the Universal Protein Resource, is a central repository of protein sequence and annotation data. It is widely recognized for its comprehensive, high-quality data, making it an essential resource for bioinformatics and computational biology. Uniprot integrates information from various sources, including experimental studies, computational analysis, and literature, providing a rich and reliable dataset for scientific research. Key features of Uniprot include:

1. Comprehensive Protein Data: Uniprot contains a vast collection of protein sequences, functional annotations, and cross-references to other databases, making it a valuable resource for protein research.

2. Reviewed Entries: Uniprot contains both reviewed (Swiss-Prot) and unreviewed (TrEMBL) entries. Reviewed entries are manually curated by experts, ensuring high accuracy and reliability.

3. Functional Annotations: Each protein entry includes detailed functional annotations, such as catalytic activity, biological processes, and involvement in pathways.

4. 3D Structural Data: Uniprot links to structural databases like PDB (Protein Data Bank), providing access to 3D structures of proteins, which are crucial for understanding enzyme mechanisms.

5. Cross-references: Extensive cross-references to other databases (e.g., PDB, BRENDA, Reactome) enhance the richness of the data.

For this study, Uniprot was chosen due to its high-quality data, extensive coverage of protein information, and user-friendly interface. The data collection process involved querying Uniprot for enzyme entries with 3D structural data and catalytic activity annotations, extracting relevant information, and preprocessing the data for model development. The next sections describe the data preprocessing, feature engineering, and model development steps in detail.

The data retrieval process involved using the Uniprot REST API to download protein data that matched specific criteria. The criteria included reviewed entries with both 3D structural data and catalytic properties. The Python script below was used to automate the data retrieval and preprocessing steps: [**polleyTobiasPolDeepZyme2024**]

1. API Request: The script constructs a query to the Uniprot REST API to retrieve reviewed protein entries with specified fields and criteria.
2. Data Retrieval: Data is retrieved in compressed format and decompressed using gzip.
3. Data Parsing: The decompressed data is read into a Pandas DataFrame.
4. Data Filtering: The DataFrame is filtered to retain entries with non-null EC numbers and PDB codes.
5. Data Splitting: Entries with multiple PDB codes are split into separate rows for each PDB code.
6. Data Saving: The processed data is saved as a TSV file for further analysis.

4.2 Data Preprocessing

Data preprocessing is a crucial step in preparing the dataset for the prediction model. It involves the protein structure download, p2rank workflow, sequence extraction and combination the data into a single dataset. The first step in data preprocessing is to download the 3D protein structures from the Protein Data Bank (PDB) using the PDB ID obtained from Uniprot. The PDB is a repository of experimentally

determined protein structures, providing valuable insights into the 3D organization of proteins. The structures are needed for the p2rank workflow, which predicts the interactive site residues in the protein structures.

This script performs the following steps:

1. requests.get: Sends an HTTP GET request to the PDB URL to download the protein structure file.

2. Retry Logic: Attempts to download the file up to three times in case of failures.

3. File Writing: Saves the downloaded PDB file to the specified directory if the download is successful.

The ThreadPoolExecutor is used to parallelize the download process and speed up the data retrieval.

The next step in data preprocessing is to run the p2rank workflow on the downloaded protein structures to predict the interactive site residues. With the following lines of code, the p2rank workflow is executed on the directory containing the PDB files:

4.3 Feature Engineering

Feature engineering is a critical step in preparing data for prediction models. This process involves transforming raw data into meaningful features that can improve the performance of the model. In this section, the author describes the feature engineering techniques used in this study, focusing on the processing of protein sequences and the calculation of additional features to enhance the predictive power of the Deep Learning model.

To capture meaningful information from protein sequences, this study used several features derived from the sequences, including amino acid composition, molecular weight, isoelectric point, hydrophobicity, and sequence length. These features provide valuable insights into the physicochemical properties of the proteins, enabling the

model to learn patterns that correlate with enzyme functions. The ProteinAnalysis class ferom the Biopython library was used to calculate these features. The following Python code snippet demonstrates the calculation of additional features from the protein sequences:

The first step is to clean the protein sequence shown in chapter **??**. The sequence itself is used as a feature, and additional features are calculated using the ProteinAnalysis class from Biopython. To convert the cleaned sequences into a format suitable for the model, a tokenizer is used to encode the sequences into numerical data. In the context of protein sequences, each amino acid is mapped to a unique integer. For example, the sequence "ACDEFGHIKLMNPQRSTVWY" is tokenized into a list of integers.

Tokenization involves converting each amino acid into an integer based on its position in a predefined list of valid amino acids. This process can be mathematically represented as: token $(x) = i$ where $x\epsilon\{A, C, D, E, F, G, H, I, K, L, M, N, P, Q, R, S, T, V, W, Y\}$ where i is the index of the amino acid x in the list.

The tokenized sequence is then passed through an embedding layer that transforms these integers into dense vectors. This embedding process is essential for capturing the contextual meaning of each amino acid within the sequence: $embedding(i) = v$ where v_i is the embedding vector for the token i. These embeddings are fed into the RNN, which processes the sequence and updates its hidden states accordingly, allowing the model to capture complex dependencies and interactions between amino acids. The sequences are then padded to ensure they all have the same length, which is necessary for batch processing in Deep Learning models.

Recent advancements have demonstrated that sequence-based models, including language models like ESM-1b, can achieve high accuracy in predicting protein functions and properties. For instance, the study by Hu et al. (2022) highlights the potential of protein-sequence based models like ESM-1b in predicting protein function from sequences. [**huExploringEvolutionbasedFree2022**]

In addition to tokenizing the protein sequences, several biochemical features are calculated to provide a comprehensive representation of the proteins. These fea-

tures include amino acid composition, molecular weight, isoelectric point, hydrophobicity, and sequence length. The Python code for calculating these features is as follows:

1. **Amino Acid Composition**: The amino acid composition represents the relative frequency of each of the 20 standard amino acids in a protein sequence.

    a) Calculation: It is calculated as the percentage of each amino acid in the sequence.

    b) Relevance: Different proteins have characteristic amino acid compositions that can provide clues about their function and stability. For example, membrane proteins often have higher hydrophobic amino acid content.

    c) Example: A protein with a high proportion of hydrophobic amino acids might be involved in membrane-related processes.

2. **Molecular Weight**: Molecular weight is the total mass of all amino acids in the protein sequence.

    a) Calculation: It is calculated by summing the average atomic masses of the amino acids in the sequence.

    b) Relevance: The molecular weight of a protein can influence its physical and chemical properties, such as solubility and interaction with other molecules.

    c) Example: Enzymes with larger molecular weights may have multiple domains or subunits.

3. **Isoelectric Point**: The isoelectric point is the pH at which the protein carries no net electrical charge.

    a) Calculation: It is determined by calculating the pH at which the positive and negative charges on the amino acids balance out.

b) Relevance: The pI affects protein solubility and interaction with other molecules. Proteins are least soluble at their pI and more likely to precipitate.

c) Example: Proteins with a low pI are often found in acidic environments, such as lysosomal enzymes.

4. **Hydrophobicity (GRAVY Score)**: The GRAVY (Grand Average of Hydropathicity) score is a measure of the overall hydrophobic or hydrophilic nature of a protein.

   a) Calculation: It is calculated by averaging the hydropathy values of all amino acids in the sequence.

   b) Relevance: Hydrophobicity influences protein folding, stability, and interaction with membranes.

   c) Example: Transmembrane proteins typically have a high GRAVY score due to their hydrophobic transmembrane regions.

5. **Sequence length**: The sequence length is the total number of amino acids in the protein sequence.

   a) Calculation: It is simply the count of amino acids in the sequence.

   b) Relevance: The length of a protein can indicate its complexity and the number of functional domains.

   c) Example: Longer proteins may have multiple functional domains or be involved in complex regulatory mechanisms.

These biochemical features provide a multi-dimensional representation of protein sequences, capturing both sequence-specific information and physicochemical properties. This feature-set is essential for analyzing the enzymes and predicting their functions accurately. A study by Gainza et al. (2020) demonstrates the importance of incorporating physicochemical features in protein function prediction models, showing

that these features enhance the model's performance. [**gainzaDecipheringInteractionFingerp**

This

4.4 Model Development

4.4.1 Model Evaluation

Code Listing 1: Python script for data retrieval and preprocessing from Uniprot

```
import requests
from tqdm import tqdm
import gzip
from io import BytesIO
import pandas as pd

base_url = "https://rest.uniprot.org/uniprotkb/search?compressed=true&
fields=accession%2Creviewed%2Cid%2Cprotein_name%2Cgene_names%2Corganism_name%
2Cec%2Corganism_id%2Crhea%2Cxref_alphafolddb%2Cxref_pdb%2Cxref_brenda%
2Cxref_biocyc%2Cxref_pathwaycommons%2Cxref_sabio-rk%2Cxref_reactome%
2Cxref_plantreactome%2Cxref_signor%2Cxref_signalink%2Cxref_unipathway&
format=tsv&query=%28*%29+AND+%28reviewed%3Atrue%29+AND+%28proteins_with%
3A1%29+AND+%28proteins_with%3A13%29"

size = 500
offset = 0
all_data = []

response = requests.get(f"{base_url}&size=1")
if response.status_code == 200:
    total_results = int(response.headers.get("x-total-results", 0))
else:
    print(f"Fehler beim Abrufen der Daten: {response.status_code}")
    total_results = 0

with tqdm(total=total_results, desc="Abrufen der Daten", unit=" Eintrag") as pbar:
    while offset < total_results:
        url = f"{base_url}&size={size}&offset={offset}"
        response = requests.get(url)

        if response.status_code == 200:
            with gzip.GzipFile(fileobj=BytesIO(response.content)) as f:
                data = f.read().decode('utf-8')
            if not data.strip():
                break
            all_data.append(data)
            offset += size
            pbar.update(size)
        else:
            print(f"Fehler beim Abrufen der Daten: {response.status_code}")
            break

combined_data = "\n".join(all_data)

df = pd.read_csv(BytesIO(combined_data.encode('utf-8')), sep='\t')
```

Code Listing 2: Python script for downloading the pdb structure

```
def download_pdb(pdb_id):
    pdb_url = f"https://files.rcsb.org/download/{pdb_id}.pdb"
    retries = 3
    for attempt in range(retries):
        try:
            response = requests.get(pdb_url, timeout=10)
            if response.status_code == 200:
                with open(f'../data/data_preparation/raw_pdbs/{pdb_id}.pdb', 'w'
                    file.write(response.text)
                return f"Download of {pdb_id} successful."
            else:
                return f"Fehler beim Herunterladen der PDB-Datei {pdb_id}: {resp
        except requests.exceptions.RequestException as e:
            if attempt < retries - 1:
                continue
            else:
                return f"Error while downloading {pdb_id}: {e}"

with ThreadPoolExecutor(max_workers=10) as executor:
    results = list(tqdm(executor.map(download_pdb, pdb_ids), total=len(pdb_ids)
```

Code Listing 3: Command Line for running p2rank on a given directory

```
!./prank.sh predict /Users/tobias.polley/Repositories/DeepZyme/data/data_prepa
```

```
    def calculate_features(sequence):
        sequence = clean_sequence(sequence)
        analysis = ProteinAnalysis(sequence)
        amino_acid_composition = list(analysis.get_amino_acids_percent().val
        molecular_weight = analysis.molecular_weight()
        isolectric_point = analysis.isoelectric_point()
        hydrophobicity = analysis.gravy()
        sequence_length = len(sequence)
        return amino_acid_composition + [molecular_weight, isolectric_point

    additional_features = df["sequence"].apply(calculate_features)
    additional_features = np.array(additional_features.tolist())
```

Figure 5: Source: [**polleyTobiasPolDeepZyme2024**]

```
tokenizer = Tokenizer()
tokenizer.fit_on_texts(sequences)
encoded_sequences = tokenizer.texts_to_sequences(sequences)

max_sequence_length = max([len(seq) for seq in encoded_sequences])
padded_sequences = pad_sequences(encoded_sequences, maxlen=max_sequence_len
```

Figure 6: Source: [**polleyTobiasPolDeepZyme2024**]

5 Results

5.1 Model Performance

5.2 Comparative Analysis with Existing Models

5.3 Interpretation of Model Predictions

Discussion

5.1 Implications of Findings

5.2 Strenths and Limitations

Conclusion

5.1 Summary of Findings

5.2 Contributions to the Field

5.3 Final Remarks and Future Work

Appendix

Appendix

Appendix 1 Filler

Appendix 1.1 Filler

- **Filler**

List of References

Statement of independent work

Hiermit erkläre ich, dass ich die vorliegende Bachelorthesis selbständig angefertigt habe. Es wurden nur die in der Arbeit ausdrücklich benannten Quellen und Hilfsmittel benutzt. Wörtlich oder sinngemäß übernommenes Gedankengut habe ich als solches kenntlich gemacht. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Bergisch Gladbach, Sunday 23rd June, 2024

Tobias Polley