Dalian University of Technology
Undergraduate Capstone Project (Thesis)

Peptide Detectability Prediction Based on Interpretable Classification Model

Department:	Dalian Leicester Institute
Major:	Applied Chemistry
Name:	Dong Junjie
Student Number:	201823017
Supervisor:	Prof.He Zengyou
Review Teacher:	Prof.He Zengyou
Completion Date:	May 31, 2022

大连理工大学

Dalian University of Technology

原创性声明

本人郑重声明：本人所呈交的毕业设计（论文），是在指导老师的指导下独立进行研究所取得的成果。毕业设计（论文）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究成果做出重要贡献的个人和集体，均已在文中以明确方式标明。

本声明的法律责任由本人承担。

作者签名：　　日期：

关于使用授权的声明

本人在指导老师指导下所完成的毕业设计（论文）及相关的资料（包括图纸、试验记录、原始数据、实物照片、图片、录音带、设计手稿等），知识产权归属大连理工大学。本人完全了解大连理工大学有关保存、使用毕业设计（论文）的规定，本人授权大连理工大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用任何复制手段保存和汇编本毕业设计（论文）。如果发表相关成果，一定征得指导教师同意，且第一署名单位为大连理工大学。本人离校后使用毕业毕业设计（论文）或与该论文直接相关的学术论文或成果时，第一署名单位仍然为大连理工大学。

论文作者签名：　　日期：

指导老师签名：　　日期：

Peptide Detectability Prediction Based on Interpretable Classification Model

Abstract

In proteomics, the detectability of peptide sequence is an important problem, as the detection accuracy of peptide sequence directly affects the correctness of protein identification. However, due to the randomness of peptide selection in the mass spectrometer, applying the high-throughput proteomic techniques can be a challenging issue, as the result is hard to reproduce. Therefore, predicting the detectability of peptides is very necessary in proteomics. At present, mainstream research focuses on how to improve the accuracy of peptide existence prediction, but rarely considers the interpretability of the model. Despite the importance of accuracy, interpretability is also an important feature that needs to be considered. Retaining interpretability when predicting the detectability of the peptide sequences is beneficial for understanding the detection process and providing a reference for optimization of the experiments.

Based on the issues mentioned above, this paper proposed an effective and interpretable peptide sequence detectability prediction model named PrefixSpan-DRN based on sequential pattern mining techniques and Decision Rule Network. In this model, PrefixSpan sequential pattern mining is used to extract sequential patterns. Subsequently, a contrast threshold is set to filter the effective discriminant patterns. Finally, feature vectors are generated according to the existing constraint of the remaining patterns. Then the decision rule network is trained and used to generate decision sets. Experimental results show that PrefixSpan-DRN can achieve the classification accuracy of the current mainstream algorithms under the premise of ensuring the interpretability of the model. In addition, the generated rules show a strong generalization ability in the cross-species transfer task, indicating that the rule itself may be a has research value.

Key Words: Peptide Detectability; PrefixSpan;Sequential Pattern Mining; Interpretable Decision Set

Catalogue

1 Introduction

1.1 Research Purpose and Motivation

1.1.1 Purpose

Proteomics, first proposed by Marc Wilkins in 1994[1], refers to the research method of studying proteins to obtain protein information about cells, tissues and organisms. At present, more and more research studies need to rely on proteomics research. For example, in the future genome project, in order to understand the logical framework of genetic language in the genome, it needs to study the expression of the genome and the function of protein products[2]. Two main proteomics research methods are "top-down" and "bottom-up". In the case of the bottom-up approach, protein sequences are firstly broken down into short peptide sequences. A variety of techniques can then be used to obtain information about the protein.

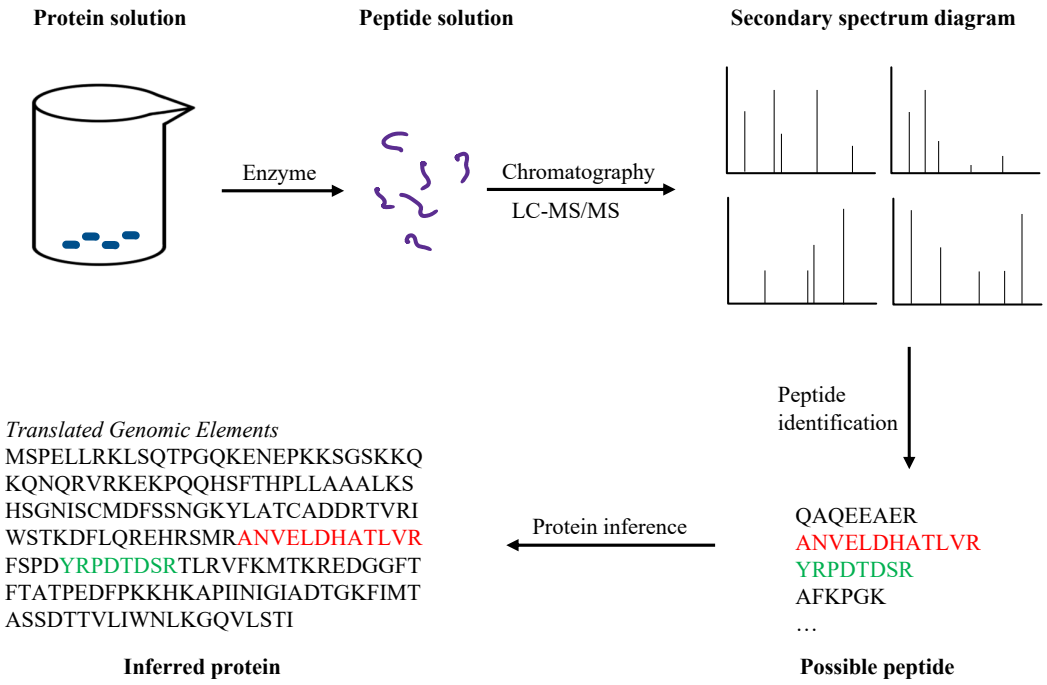


Figure 1.1 Example of a "bottom up" method of protein inference

Among the tasks of proteomics, protein identification is one of the most important research projects, the main purpose of which is to determine all proteins expressed in a

sample. Biologically, protein inference is key to understanding disease mechanisms and drug discovery. Take the shotgun method as an example of the bottom-up approach. First of all, it needs to pre-separate protein so that we are able to analyse it, then it is necessary to use the protease enzyme for digestion in order to separate the long pieces in to small peptide. Then the solution will be analysed with liquid chromatography-tandem mass spectrometry (LC-MS/MS). The last step is to apply computational analysis to identify the peptide, which mainly consists of database searching, spectral library searching, de novo sequencing, and hybrid methods. In MS detection of peptide sequences, some peptide sequences show the uncertainty of detection, which greatly affect the accuracy of peptide determination and protein inference. Therefore, it is an urge to develop an approach for measuring the detectability of peptides. However, due to numerous experimental variables, the mechanism behind peptide detection is still unclear. Therefore, detectability is generally not considered in the process of protein inference.

Sequence, a common data type, refers to an ordered linear list consisting of a set of elements. Essentially, a peptide can be regarded as a sequence of amino acids organized in a specific manner. An example of the peptide sequence can be found in Figure 1.2, the peptide sequence can be represented as $SGIYGGACLAALYPCPT$.

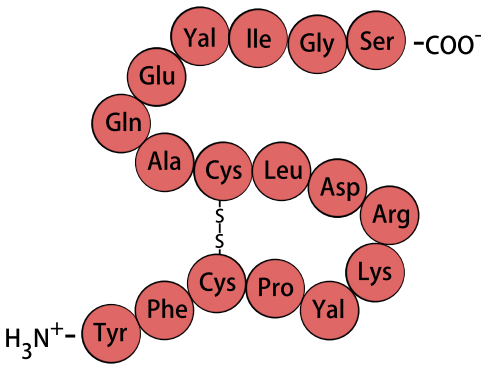


Figure 1.2 Examples of a peptide.

With the rapid development of bioinformation, a series of peptide databases have been established to support scientific researches, which provides the feasibility of using machine learning-based algorithms to analyse the peptide detectability[3,4]. This paper mainly focuses on interpretable peptide detectability prediction.

1.1.2 Motivation

In recent years, both traditional machine learning and novel deep learning methods have been presented to predict the detectability of MS spectrometry by defining it as a dichotomous problem, but these algorithms only focus on the accuracy of the model. Despite the importance of accuracy, interpretability is also a vital aspect that we need to consider as it may contribute to the process of understanding detection mechanisms. The algorithms presented before achieved high accuracy, but it can be argued that the model does not behave well across different systems. Even with good performance, the model's results are unconvincing due to the black-box nature of these models. Meanwhile, only parameters of the model accuracy cannot help researchers to understand the biological property behind the model.

Previous algorithms indicated that the peptide detectability was mainly dependent on the relative position of the amino acids. Thus, the frequent patterns in peptide sequence can be used to determine whether the peptide is detectable. According to this idea, this paper focuses on finding a set of rules based on the existence of patterns that can classify accurately while keeping readable. This paper presents a pure sequence framework to predict the detectability of peptides, which consists of two parts. Firstly features are generated with the sequential pattern mining module. In the next step, the classifier is trained in the decision rule sets learning module.

1.2 State-of-the-art

This section summarizes the relative works on peptide detectability and the sequence-based technique.

1.2.1 Relative Works on Peptide Detectability

The concept of peptide detectability was initially presented by Tang al[5]. With the rapid development of bioinformatics and data science, more and more biological databases are established, and many algorithms used in the field of computer science can be migrated to the field of bioinformatics, providing a method to analyse and predict detectability without wet experiments.

Early analysis[6] of peptide data started from physicochemical properties[7] and selected suitable features, such as aaindex-derived and sequence-derived, etc., and obtained results through training of classifiers. Building data from the molecular level is still prob-

lematic, and protein data analysis is high throughput, making it difficult to train models. It requires a lot of prior knowledge and a lot of work to screen from the perspective of physicochemical properties. In addition, the classifier trained in certain species has strong pertinence, that is, insufficient generalization ability. Consider AP3 algorithm[3] as an example, it can achieve high accuracy on its building data set, but it is not ideal on another unfamiliar data set. Moreover, from the perspective of text classification technology, the algorithm to achieve high classification accuracy based on sequence only was designed. Later, with the development of natural language processing technology, people began to use NLP based methods to study sequences[8]. Some deep learning methods combining physicochemical properties have also been proposed[9]. Subsequently, this chapter will introduce two algorithms, AP3, based on physicochemical properties, and Pepformer, which is based on the Siamese network, a deep learning algorithm.

AP3 is a feature-based algorithm, that integrates peptide digestibility and peptide detectability. For the source of the data set, AP3 chooses public large-scale yeast data set. There are two main workflows, which are peptide digestibility predictor and peptide detectability predictor. In case of confusion on the concept of peptide detectability and peptide digestibility, the dataset is divided into two parts, one for training peptide digestibility and one for training peptide detectability.

In the first part, a random forest classifier is used to predict probability of cleavage of each tryptic site. Then it follows the peptide digestibility calculation, the following formula can be applied to calculate the probability of peptide digestibility:

$$peptide\ digestibility = e_N * e_C * \prod_{i=1}^{n}(1 - e_i), \tag{1.1}$$

in which, e is the probability provided by the classifier, the subscript indicating the site of tryptic is N or C, e_i is the cleavage probability of i-th missing cleavage site, and n is the total number of missed cleavage sites.

The second part is the peptide detectability predictor, it combines the peptide digestibility and other aaindex features to predict the detectability of peptides.

PepFormer is a novel deep learning network based on the Siamese network structure. This algorithm does not need the peptide physicochemical properties and other properties, meanwhile, its gated recurrent unit ensures the ability to learn the context-sensitive em-

bedding representations. The datasets of Pepformer are come from GPMDB database[10]. The peptide sequences are ranked and labelled according to the observed times. Pepformer is mainly consists of 3 main modules, a sequence embedding module, a Siamese network module and an optimization module.

In the first module, the sequences are embedded into vectors in a "token" way, that is A(alanine) to 1, R(cysteine) to 2, For example, the sequence $LCYVALDFEQ$ can be converted to $[11, 5, 21, 22, 1, 11, 4, 14, 7, 6]$, if the length of the sequence is less than the set length, 0 will be added to the tail to ensure the vector has the same length. For the Siamese network module, Pepformer uses Transformer to extract the features. The position relationship is considered using an encoding layer. The second module is optimization. In this part, the backpropagation algorithm is used to update the parameters, and a cross-entropy is defined as the loss of the classifier.

1.2.2 Sequential Pattern Mining Technology

The order of the itemsets in sequence reflects some potential information. Consider the example of shopping in the mall, a customer buying a novel phone implies that the phone shell is the latent demand. In biology information, especially for protein, the sequence determines the macro performance of the molecular characteristics. Sequential pattern mining refers to finding interesting, useful, especially discriminant patterns for classification from the dataset, and there are three main categories. The algorithms based on apriori-concepts were initially proposed and then developed as frequent sequential pattern mining. The second and the third are the discriminant sequential pattern mining and constraint-based algorithms. The recent research on these three aspects will be introduced in this section.

(1) Frequent sequential pattern mining

The task of frequent sequential pattern mining is finding all the sequential patterns that the support value satisfied the minimum number set by users. It can mainly be divided into two main parts, which are breadth-first search and depth-first search.

Consider breadth-first search approach GSP (Generalized Sequential Pattern) as an example[11]. Given that a sequence database $\mathcal{D}$ with support threshold value θ,the frequency of the sequence σ appears in the dataset $\mathcal{D}$ is:

$$Occ(\sigma, \mathcal{D}) = |\{t \mid s \subseteq t, t \in D\}|. \tag{1.2}$$

Task of GSP is mining all the frequent sequential patterns $\mathcal{P}$ that support is larger than threshold θ., in which the support is defined as:

$$Supp(\sigma, \mathcal{D}) = \frac{Occ(\sigma, \mathcal{D})}{|\mathcal{D}|}. \tag{1.3}$$

The algorithm is based on the idea that if $\mathcal{P}$ is frequent, then all the subsequences of $\mathcal{P}$ are frequent. Like apriori, GSP scan the database and find the patterns set that the length $|\sigma|$ of a sequence σ satisfied $|\sigma| = k$ and calculates their support value, patterns that support is larger than threshold θ will be the candidate for the next iteration. With time constraints and sliding window, the number of candidate patterns is greatly reduced.

(2) Discriminant sequential pattern mining

The target of discriminant sequential pattern mining is different from the frequent sequential pattern mining, as the former is a supervised pattern discovery algorithm, focusing on extracting patterns with significant differences between different classes. Up to today, many algorithms based on different contrast measures have been presented. There are two main approaches, one is based on the set threshold then mining all the satisfied, and another is the Top-k form. Compared with the set threshold, the Top-k form is more flexible, especially for the unfamiliar dataset. The only parameter necessary to determine is the value k, which provides the feasibility for pruning.

Based on the constraint pattern of minimal distinguish subsequence(MDS), Xiaonan Ji et al. described an efficient approach named ConSGapMiner[12]. It focuses on mining all the discriminant sequential pattern sets which satisfy the maximum gap constraint. What's more, ConSGapMiner filtrate the patterns based on the frequency threshold of each class. However, it also excludes some patterns where the frequency is high in one class while low in another class. Many algorithms expanded based on the ConSGapMiner, as its high efficiency and unity.

(3) Constraints-based algorithms

Currently, pattern generation by apriori-based algorithms inevitably to be redundant, it can be argued that plenty of the useless information is discovered tautologically. Discovering all the frequent patterns in a large database could be a challenging issue. Based on the purpose of reducing the redundant patterns, increasing the performance and mining more interesting patterns, the idea of constraints-based algorithms has been proposed[13].

Constraints are limitations for controlling the accuracy, quality and most of all number of discovered patterns. Generally, there are two main branches for constraints-based algorithms. One approach is filtrating the pattern discovered. Nevertheless, such operations still consume lots of time and memory. To address this problem, the second method applies the constraints during searching. By using this method, the consumption is reduced by orders of magnitude. GSP[11] also integrate constraints, including gap constraint, duration constraint. Zaki presented cSPADE algorithm[14], it reduces the search space greatly with the minimal or maximum gap constraint, a time window, etc. Many constraints-based algorithms and the criteria of constraints were presented subsequently[15–17].

1.2.3 Sequence Classification

There are many overlaps between sequential pattern mining and sequence classification, of which pattern mining is outlined in the previous section. It has been proved that sequence classification is involved in wide applications such as bioinformation analysis, health monitoring and anomaly detection. The challenge of this task is mainly for the reason of the latency of the features. Thus, many feature-based algorithms cannot be directly applied. Even with the feature selection technique, the result is not satisfactory. This section provides an overview of the sequence classification algorithms. In general, there are three kinds of sequence classification algorithms: feature-based, sequence distance-based, and model-based[18].

(1) Feature-based classification

For traditional numerical data, there exist widely classification methods to be choosen, while not suitable for symbolic data such as peptide sequence. So one way is extracting features from sequence or converting sequence into features. Naively, every element of symbolic sequence can be represented as a feature, while this operation loses its characteristics of time dependency. For example, $\langle A, C, G, H, I, L, P, S, T, Y \rangle$ is able to represent sequence $GIYCGGASLAACYPLPTTLHLA$ shows in Figure 1.1. Hence, a method named k-grams was proposed to address this issue[19]. k-grams refers to a short sequence segment with k number of elements. k-grams presence and absence or frequency leads to the feature representation of the sequence. Inexact matching operations are also considered an optional choice to enhance expression ability.

After a sequence is transformed into a feature vector, traditional machine learning methods such as decision tree, naive Bayes and support vector machine can be applied for

sequence classification. However, the k-grams method also has some disadvantages. For example, when the sequence length is too long, the size of the feature set will become very large, which leads to the difficulty of obtaining good classification results in the subsequent model training.

Therefore, another feature transformation method based on the pattern was proposed. Compared with k-grams, the pattern-based method can transform sequences into feature vectors in controllable dimensions, filter features more efficiently in a non-redundant manner, and generally maintain a higher classification accuracy than k-grams.

(2) Sequence distance-based classification

Another method is to classify sequences by comparing their similarity. To obtain the similarity between sequences, the definition of the distance function of sequences was proposed. With the sequence distance function, the sequence can be classified by some existing classification methods, such as KNN. It is worth noting that the effect achieved by the classifier depends heavily on the measures of sequence distance. Here are a few examples of sequential distance functions:

Euclidean distance[20] is a measure commonly used in the simple time series, which is defined as follows:

Assume that there are two time series s and s', Euclidean distance between them is,

$$dist(s, s') = \sqrt{\sum_{i=1}^{L}(s[i] - s'[i])^2}. \tag{1.4}$$

Euclidean Distance has a competitive performance on the 1NN classifier compared with other sequence distance measures. However, Euclidean distance also has several disadvantages. First, it requires that the two sequences have the same length, which is very demanding for the data. Second, Euclidean Distance is sensitive to distortions. To solve this problem, Dynamic time warping distance (DTW) was proposed[21]. Furthermore, dynamic time warping distances do not have the limitations of equal length.

For symbolic sequence, several algorithms were presented, such as Hamming distance, Levenshtein distance and longest common subsequence similarity. These algorithms mainly can be divided into edit based, Token based and sequence-based. Here is the description of Hamming distance:

Hamming distance[22] is the number of different symbols between the two sequences

in the corresponding position. In another word, it can be defined as the minimum number of changing one sequence to another by replacing the character one by one. Consider $karolin$ and $kathrin$ as an example, the Hamming distance between them is 3, because of replacing rol to thr.

(3) Model-based classification

Model-based classification assumes that a class of sequence data belongs to a specific underlying model M. M learns from a given data set to obtain the probability distribution of the sequence in each type. The main Model methods include the Naive Bayes sequence classifier, Markov Model and Hidden Markov Model.

1.2.4 Interpretable Decision Sets Learning

Intractability is a vital feature of a machine learning module, indicating the ability of humans to understand the cause of the module. To achieve the goal of high accuracy while keeping interpretable, a method of decision rules has been developed. According to Molnar[23], a decision rule is a simple IF-THEN statement consisting of a condition and prediction. The structure of the decision rule ensures its interpretability, as it is similar to the natural way of the human brain, more interestingly, like coding. The formation of an IF-THEN rule behaves like this,

IF condition THEN conclusion.

The following rule is an example:

IF *age* = *youth* AND *student* = *yes* THEN *buys_computer* = *yes*.

A decision set is an unordered combination of multiple rules, in which all rules are mutually exclusive, while a decision list is an ordered one. Compare with other rule-based algorithms, the simple structure of decision lists leads to high learning efficiency. In early research, the sequential covering is a common ideal for learning a single rule to create a decision list, and many algorithms[24–26] have been proposed based on this greedy ideal. Another school[27,28] originated from the associated rule in data mining, which generates frequent rules first then pruning.

Despite that decision lists already resemble human thoughts, it still has a deficiency, as it is a tree-like structure. Thus, the concept of interpretable decision sets has been presented[29], it is more comprehensible compared to the decision list due to its flat structure. The Figure 1.3 shows the core difference between the two methods. As it can be seen from the left one, the decision set does not rely on the order of features, contributing to the power

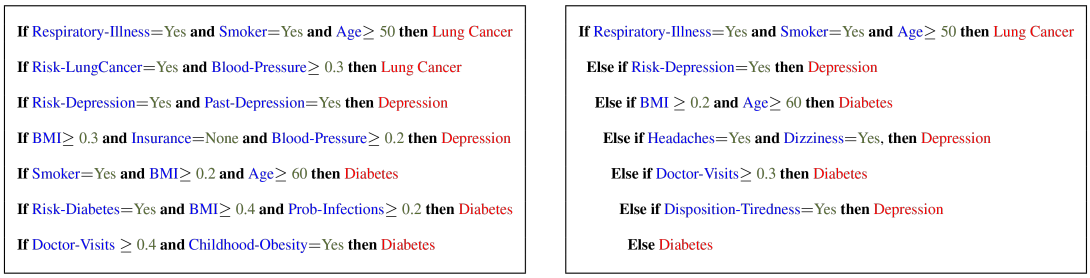


Figure 1.3 Difference between interpretable decision set and decision list[29].

of comprehension. The initial approach has the drawback that it consumes huge memory, which means that it can only deal with the task containing a few features. What's more, this algorithm runs slow, it would consume lots of computing resources. However, the framework itself is superior, as it provides a new sight of the rule-based method. Under the influence of this ideal, algorithms like GC[30], BRS[31] made some improvements and achieved high performance.

1.3 Main Contributions

Firstly, this paper proposes a sequence-based peptide detectability prediction model, which is competitive with other state-of-the-art models in terms of classification performance.

In addition, this paper is the first to study peptide detectability from the perspective of interpretability, which expands the research horizon and points out the direction for further development.

1.4 The Framework of this Paper

This paper can be divided into four sections, the structure and details of each section are followed as below:

In section 1, the research background is introduced, which describes the urgency of peptide detectability prediction. It also summarizes the research status on peptide detectability prediction, sequential pattern mining, sequence classification, and interpretable decision set.

Section 2 provides details of the algorithms used in the interpretable peptide detectability prediction model presented in this paper, including classical and novel pattern mining techniques and the theory of interpretation decision set.

The process of the interpretable peptide detectability prediction model is described in section 3, it consists of the module of pattern mining and decision rule sets learning.

Section 4 describes the experimental details and results, also the comparisons of other state-of-the-art techniques.

Finally, Section 5 overviews the contributions and provides a holistic assessment of the module.

2 Relative Theories

2.1 Representation of Peptide Sequence

A protein sequence is a sequence consisting of basic amino acids, in which the amino acid is formed by one centre carbon atom connected with one azyl-terminal, one carboxyl-terminal, one hydrogen and one R group. Figure 2.1 shows the basic structure of amino acids.

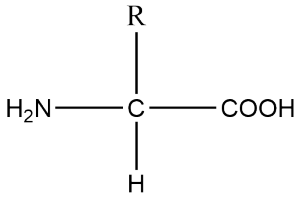


Figure 2.1 Structure of amino acid.

The choice of the R group determines different classes of amino acids. For example, the R group for glutamate is hydrogen while methyl for alanine. In this way, the basic amino acids can be classified into twenty kinds shown in Table 2.1. These twenty kinds of amino acids combine in a certain way and lead to a peptide sequence.

2.2 Sequential Pattern Mining Technologies

2.2.1 Definition and Symbol Description

Definition 2.1 Itemset refers to a non-empty set $I = \{i_1, i_2, i_3, \cdots, i_n\}$, in which element $i_k (1 \leq k \leq n)$. Every element is unique and unordered in the itemset.

Definition 2.2 Sequence is an ordered list of itemset, which can be defined as $a =< a_1, a_2, a_3, \cdots, a_m >$, in which $a_i \subseteq I (i = 1, \cdots, m)$.

2.2.2 k-mer

k-mer is a commonly used method in DNA sequence, while it can also be migrated for proteomics analyse. k-mers refer to all the k length substrings in a sequence.

The pseudocode for k-mers generation is provided in algorithm 2.1.

In bioinformatics, k-mer is widely used for the feature extraction process, which considered the near position relationship in an amino sequence. Consider sequence a with

Table 2.1 Twenty kinds of amino acids.

Amino acid	Abbreviation	Sign	Amino acid	Abbreviation	Sign
Alanine	Ala	A	Leucine	Leu	L
Arginine	Arg	R	Lysine	Lys	K
Asparagine	Asn	N	Methionine	Met	M
Aspartic acid	Asp	D	Phenylalanine	Phe	F
Cysteine	Cys	C	Proline	Pro	P
Glutamic acid	Glu	E	Serine	Ser	S
Glutamine	Gln	Q	Threonine	Thr	T
Glycine	Gly	G	Tryptophan	Trp	W
Histidine	His	H	Tyrosine	Tyr	Y
Isoleucine	Ile	I	Valine	Val	V

Algorithm 2.1 k-mers $(String\ Seq, Integer\ k)$

$L \leftarrow length(seq)$
$kmers \leftarrow$ new array of $L-k+1$ empty strings
for $n \leftarrow 0\ to\ L-k$ **do**
  $k-mers[n] \leftarrow$ subsequence of seq from letter n inclusive to letter $n+k$ exclusive
end for
return $k-mers$

length L.

Generally, $L-k+1$ numbers of k-mers will be generated for representing a sequence with length L. As for the possible k-mers in a dataset, it is decided by the size n of the symbol set. As for protein sequence, it can be known that the possible k-mers number will reach the number 20^k.

2.2.3 Prefixspan

PrefixSpan, short for Prefix-Projected Pattern Growth, refers to a pattern mining algorithm based on the prefix-projected method[32]. It adopted the ideal of divide and conquer, generating multiple smaller projected datasets and focusing on mining frequent patterns in the projected datasets. The main cost for PrefixSpan is on generating the projected datasets, while it has lower complexity compared with the algorithms that are based on the ideal of

Apriori. In the meantime, it also avoids the high consumption of massive candidates. Thus, PrefixSpan is a sequential pattern mining algorithm with relatively high efficiency. Following are the relative definitions of PrefixSpan.

Definition 2.3 Prefix: Given a sequence $\alpha =< e_1e_2e_3\cdots e_n >$ and a sequence $\beta =< e_1^{'}e_2^{'}e_3^{'}\cdots e_m^{'} > (m \leq n)$, where b is the prefix of a if for every $i(1 \leq i \leq m)$ that satisfied $e_i^{'} = e_i$.

Definition 2.4 Suffix: Given a sequence $\alpha =< e_1e_2e_3\cdots e_n >$, the prefix of sequence α is $\beta =< e_1e_2e_3\cdots e_m > (m \leq n)$, then $\gamma =< e_1^{'}e_2^{'}e_3^{'}\cdots e_m^{'} >$ is the suffix of a about β.

Consider sequence $\alpha = KFVADGIFK$, if there does not exit other constrain, then $< K >, < KF >, < KFV >$ is the prefix of α, while $< KFA >, < FVA >$ is not the prefix. $< ADGIFK >$ is the suffix of α about $< KFV >$.

Definition 2.5 Projection: Given sequence α and β, if sequence β is the subsequence of α, then the projection $\alpha^{'}$ of sequence α about β must satisfy that: β is the prefix of a and $\alpha^{'}$ is the subsequence with max length of α.

For example, consider sequence $< KFVA >$, the projection of subsequence $< F >$ is FVA, and $< KFVA >$ for subsequence $< KF >$.

Definition 2.6 Projected database: Assume a is the sequential pattern in sequence dataset D, then projected database a^{-} refers to all the suffix about prefix a denote as $D|_{\alpha}$.

The procedure of PrefixSpan followed as below:

(1) Scan the original sequence database so that to find all the frequent items with $length = 1$.

(2) Construct the projected databases based on the corresponding sequence space that is divided from the frequent pattern with length equal to 1.

(3) Repeat the same procedure on the generated projected databases, till no frequent sequential patterns with $length = 1$ can be found.

(4) Repeat the same procedure to different projected databases till no new sequential patterns that length equal to 1.

The pseudocode of PrefixSpan can be found in Algorithm 2.2.

Algorithm 2.2 PrefixSpan(α, D)

FL = find_ frequent_1_sequence_patterns $(D|_\alpha)$
if FL = $\emptyset$ **then**
  return NULL
end if
for each $\gamma \in$ FL **do**
  if $\gamma =$ _ γ **then**
    $\alpha' \leftarrow \alpha \bowtie_i \gamma$
    out_put $(a^{'})$
  else
    $\alpha' \leftarrow \alpha \bowtie_s \gamma$
    out_put $(a^{'})$
  end if
  $D|_{a'} \leftarrow$ function _contract_project_Database $(\alpha', \mathcal{D}|_\alpha)$
  PrefixSpan($\alpha^{'}, D|_{\alpha'}$)
end for

Some researchers have pointed out that PrefixSpan is not suitable for biological sequence mining. They believe that PrefixSpan is first proposed for transaction data. In another word, for transaction data, the form of patterns only needs to be frequent, not continuous. In biological data, it is necessary to meet the requirement of frequent and continuity of mining patterns. For example, in the processing of mining protein sequential patterns, the two amino acids that are too far apart can be considered to have no interaction, that is to say, the pattern mined with these two elements as prefixes is meaningless and may even be noisy data, which is very difficult for the subsequent model training. However, for the detectability prediction of peptide sequences focused in this paper, PrefixSpan pattern mining technology meets the needs because most peptide sequences are short.

2.3 Decision Sets for Classification

2.3.1 Interpretable Decision Sets

Interpretable decision sets (IDS)[29] is a framework for generating rules for classification with high accuracy and interpretability. In this paper, the author also provides an approach for training the model.

IDS do not have a hierarchical structure. In other words, the rules obtained by them can be independently applied to classification. Meanwhile, compared with existing classification models, IDS can achieve similar classification results in problems with a small model volume. Learning IDS from data can be difficult, as it needs to find a model with

high precision in an Interpretable space. The overlapping rules are also considered in IDS by optimizing a joint objective to find a near-optimal set of rules.

The decision set is proposed as a model class for description boundaries and predicted the results. The intuitive difference between decision set and decision list is that the rule in the decision set is not connected with the $else$ statement, in this way, the rule in the decision set can be single applied. It can be argued that the expression ability of decision set, decision list and decision tree is at the same level due to their mutual representability.

Table 2.2 Notation description.

Notation	Definition	Term
$\mathcal{D}$	Input set of data points $\{(X_1, y_1), \ldots, (X_N, y_N)\}$	Dataset
X	Observed attribute values of a data point	
y	Set of class labels in $\mathcal{D}$	
p	(attribute, operator, value) tuple, e.g., $existence(MK)$	Predicate
s	Conjunction of one or more predicates, e.g., $existence(MK)$	
S	Input set of itemsets	Itemset
r	Itemset-class pair (s, c)	Rule
$\mathcal{R}$	Set of rules $\{(s_1, c_1), \ldots, (s_k, c_k)\}$	

The decision set is defined with *itemset* s, which refers to a filter for data points and can be defined as a conjunction of several *predicates* of the forms such as attribute, operate and value, specifically for peptide sequence $existence(MK)$.

For an attribute X, if all predicates in the conjunction are true for X, it can be denoted by $X\ satisfies\ s$. In a *rule* tuple (s, c), it refers to that predicates itemset s and a class label s. The formal definition of the decision set followed as below:

Definition 2.7 Decision set $\mathcal{R}$ refers to a set of tuples (s, c) consisting of itemset s and class c. The classify process behaves in a way that:

- If X satisfied the itemset s_i then it is assigned with label c_i.
- If X satisfied none of the itemset, it will then be assigned with default label.
- If X satisfied multiple itemset, then tie-breaking function is used to assign its class.

Users can manually specify the default label and tie-breaking function, while the choice of the default labels will commonly be the majority class as this will achieve higher accuracy. This make is also vital for certain areas, for example, disease prediction. In this case, the smallest minority classes can be chosen as the default class and tie-breaking function.

The decision set is naturally human-readable, as it is similar to the way people think and make decisions. However, it also exits the cognitive limitations as the complexity of the whole model. For compared and control the interpretability of the model, four different measurements that can be used to evaluate the interpretability are defined in IDS.

Size: Naturally, the length of the decision set itself is formalized as size. The number of rules indicates the difficulty of understanding the decision behaviour of the model.

Definition 2.8 Size($\mathcal{R}$) refers to the number of rules exiting in a decision set $\mathcal{R}$.

Length: Focusing on the rule itself, if there exiting too many predicates, the rule will be difficult for understanding. Term length is used to describe the size of the rule.

Definition 2.9 Length(r) is the number of predicates in a rule $r = (s, c)$.

Cover: This parameter describes the scope range of the data points in the dataset.

Definition 2.10 cover(r) refers to the set of data points in $\mathcal{D}$ with attribute X satisfied the itemset s.

Overlap: It tells the clarity of the rule in a decision set.

Definition 2.11 Overlap($r, r^{'}$) describes the set of data points that both satisfy the itemset s in $r = (s, c)$ and itemset $s^{'}$ in $r^{'} = (s^{'}, c^{'})$.

$$overlap(r, r^{'}) = cover(r) \cap cover(r^{'}). \tag{2.1}$$

To measure the accuracy of the decision set, correct-cover(r) and incorrect-cover(r) are defined.

$$\text{correct-cover}(r) = \{(x, y) \in \text{cover}(r) \mid y = c\}, \tag{2.2}$$

$$\text{incorrect-cover}\ (r) = \text{cover}(r) \backslash \text{correct- cover(r)}\,. \tag{2.3}$$

The full learning objective is,

$$\arg\max_{\mathcal{R}\subseteq\mathcal{S}\times\mathcal{C}} \sum_{i=1}^{7} \lambda_i f_i(\mathcal{R}), \tag{2.4}$$

in which the f_i is consists of interpretability and accuracy. To optimize the objective, smooth local search (SLS) was proposed.

3 Interpretable Peptide Detectability Prediction Model

This section mainly describes the Interpretable Peptide Detectability Prediction Model. Firstly, the workflow of the model is introduced. Then, it accounts for the pattern mining process of the peptide sequence and the process of filtering the patterns with the discriminant value. Finally, it describes the procedure for using a rule-based interpretable classifier to learn and predict peptide sequence detectability.

3.1 Workflow of Model

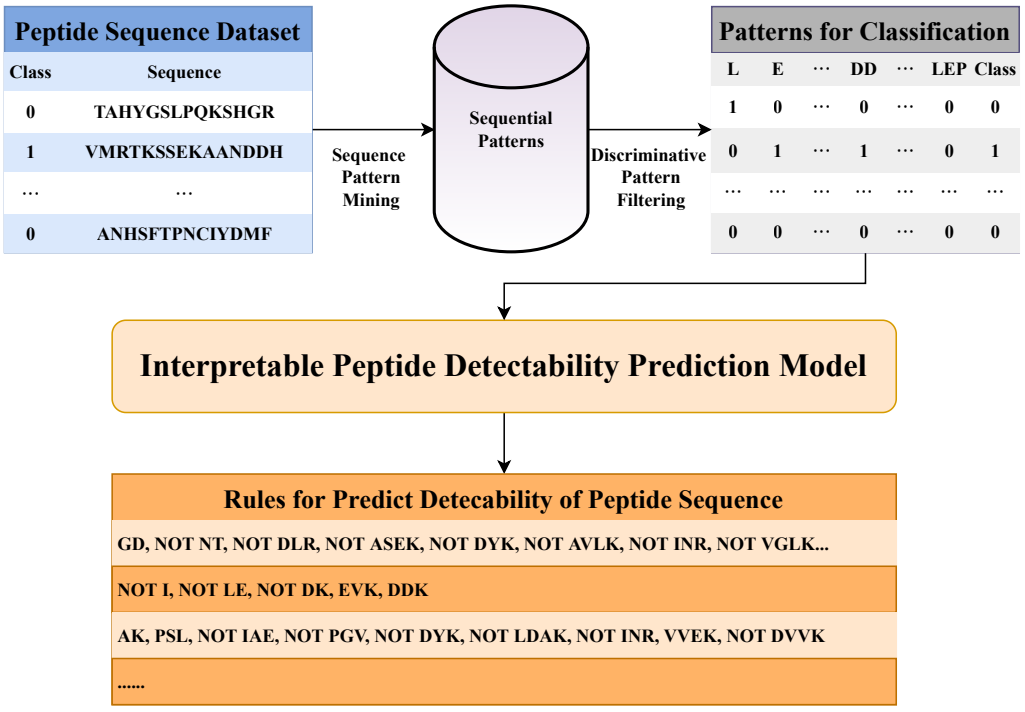


Figure 3.1 Workflow of Interpretable Peptide Detectability Prediction Model.

At present, many methods focusing on predicting peptide detectability have been proposed from different perspectives. Part of these methods are based on physic-chemistry while some are pure sequence-based, but they all have a problem in common, which is that they do not take into account the interpretability of peptide sequence in the problem of detectability prediction. Aiming at this phenomenon, this paper proposes an interpretable peptide sequence detectability prediction model. From the perspective of pure sequence

analysis, this method is the first to predict the detectability of peptide sequences by analyzing LC-MS/MS data with an interpretable model.

The interpretable Peptide Detectability Prediction Model is mainly composed of two parts: pattern mining and interpretable decision set learning. The flow chart of the model can be seen in Figure 3.1. First, the sequence data is read for generating a sequence database, and the sequence database is subsequently mined for patterns. Finally, features are trained in interpretable classifiers and readable rules will be output.

3.1.1 Sequential Pattern Mining Module for Feature Mining

In this section, two methods of pattern mining for peptide sequences are introduced, which are k-mer and PrefixSpan.

(1) k-mer

For peptide sequences, the k-mer method is a good method to extract the features, as k-mer is a position-based sequential pattern method. Studies have shown that the probability of peptide cleavage is strongly correlated with the relevant position between the items in the sequence, and this probability is one of the most important features in the process of detectability prediction, for the reason that it reflects the strength of the bond. Importantly, the k-mer method can just get relevant information from continuous substrings. Therefore, it is reasonable to use the k-mer method to obtain the features in the sequence.

Taken peptide sequence used for classification as an example:

$$S = KFVADGIFK.$$

IF $k = 2$ then, it generates the following patterns:

$$P = \{FKV, FVA, VAD, ADG, DGI, GIF, IFK\}.$$

Table 3.2 shows all possible k-mers of the peptide sequence. It can be considered that the distribution of k-mers spectrum of genes of most species is an unimodal distribution, that is, the number of modes first increases with the increase of k value, and then decreases with the increase of k value after reaching a maximum value.

The flow chart of using k-mer as the sequential pattern mining method is shown in Figure 3.2. Firstly, it needs to set the k value, and then mine the k-mers of each sequence in turn. At the same time, compare whether there are new patterns in the k-mers set and add

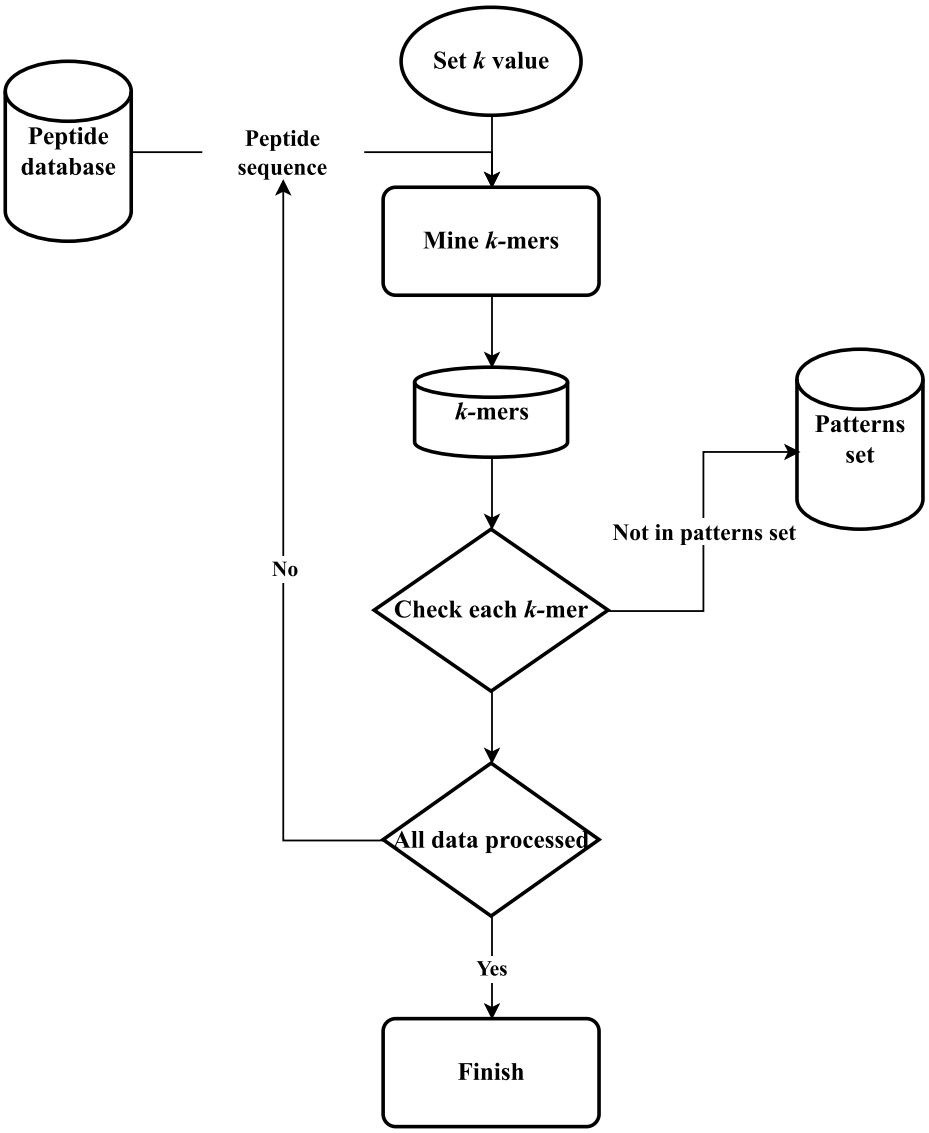


Figure 3.2 Flow chart of mining k-mers from database.

the new k-mers to the set. Therefore, the k-mers in the set are all unique. After extracting all the k-mers from each sequence, a set of patterns containing all k-mers in the sequence database can be obtained.

$$PS = \{a_1, a_2, a_3, \ldots a_n\}. \tag{3.1}$$

Subsequently, it needs to establish the feature vector for the next module, decision set

Table 3.1 k-mers for S.

k	k-mers
1	K, F, V, A, D, G, I
2	$KF, FV, VA, AD, DG, GI, IF, FK$
3	$KFV, FVA, VAD, ADG, DGI, GIF, IFK$
4	$KFVA, FVAD, VADG, ADGI, DGIF, GIFK$
5	$KFVAD, FVADG, VADGI, ADGIF, DGIFK$
6	$KFVADG, FVADGI, VADGIF, ADGIFK$
7	$KFVADGI, FVADGIF, VADGIFK$
8	$KFVADGIF, FVADGIFK$
9	$KFVADGIFK$

learning. The method of generating feature vectors is described as follows. In this module, the feature vector is defined as the existing response of the patterns, in other words, if a pattern exists in the sequence, the expression of the pattern in the vector is 1 otherwise it will be 0. The formal definition is described as follows:

For a given sequence, its feature vector is generated in a way that if the pattern a_i from sequential patterns set PS is exiting in the given sequence, then the i-th columns of this sequence will be 1, on the contrary, i-th columns will be 0.

(2) PrefixSpan

Like k-mer, PrefixSpan is a sequential pattern mining algorithm that extracts patterns based on the relationships between the item positions in the sequence. Take the sequence database in Table 3.3 as an example:

Setting that the minimum support threshold $\theta = 50\%$, the prefixes with length equal to 1 are: $\langle I\rangle, \langle M\rangle, \langle K\rangle, \langle S\rangle, \langle P\rangle, \langle H\rangle, \langle L\rangle, \langle Q\rangle, \langle R\rangle$. Table 2.4 shows the result of counting the support value.

It can be observed that only $\langle I\rangle$,$\langle K\rangle$,$\langle P\rangle$ reach the standard of the set threshold. Thus, in order to search the frequent patterns that the prefix equal to $\langle I\rangle$,$\langle K\rangle$ and $\langle P\rangle$, it only needs to search the projected database based on these three prefixes.

Table 3.2 A sequence database.

Sequence_id	Sequence
10	$\langle IIMK \rangle$
20	$\langle SPPPP \rangle$
30	$\langle HIIK \rangle$
40	$\langle LQPR \rangle$

Table 3.3 Result of counting the support value.

Frequent pattern	$\langle I \rangle$	$\langle M \rangle$	$\langle K \rangle$	$\langle S \rangle$	$\langle P \rangle$	$\langle H \rangle$	$\langle L \rangle$	$\langle Q \rangle$	$\langle R \rangle$
Counting	2	1	2	1	2	1	1	1	1
Support	0.5	0.25	0.5	0.25	0.5	0.25	0.25	0.25	0.25
Satisfied	✓		✓		✓				

If we focus on the first level of recursion, it can be found that the projected database $D|_{\langle K \rangle}$ with $\langle K \rangle$ as the prefix does not have any sequence data, so PrefixSpan will output $\langle K \rangle$ as a frequent pattern.

The next step is to search the frequent patterns with length equal to 1 in the projected database $D|_{\langle I \rangle}$, and the result is followed in Table 2.6 below.

As it can be seen in the table, it can be observed that all the support values satisfied the threshold, so three projected databases in the second level needs to be created.

Looking at the second level of recursion, it can be found prefix $\langle K \rangle$ does not have any sequence in the projected sequence databases, so the frequent pattern $\langle IK \rangle$ can be output.

In this manner, all the prefixes in the projected database will be searched in a recurred way, the recursion ends when there is no sequence in the projected sequence database, in the meantime, it outputs the frequent pattern.

The example above illustrates the workflow of PrefixSpan to extract sequential patterns from a sequence database. For this module, the workflow of sequential pattern extraction is shown below in Figure 3.3. As can be seen from the figure, this paper adopted a “two-step” approach in the process of feature extraction of peptide sequences.

In the process of mining frequent sequential patterns, it may occur data leakage if the database that is mined contains the data from the test set. Data leakage refers to operating

Table 3.4 The projected database on first level of recursion.

$D\|_a$	Prefix	Projected database
$D\|_{\langle I\rangle}$	$\langle I\rangle$	$\langle IMK\rangle$,$\langle IK\rangle$
$D\|_{\langle K\rangle}$	$\langle K\rangle$	$\langle\rangle$
$D\|_{\langle P\rangle}$	$\langle P\rangle$	$\langle PPP\rangle$,$\langle R\rangle$

Table 3.5 Result of counting the support value.

Frequent pattern	$\langle I\rangle$	$\langle M\rangle$	$\langle K\rangle$
Counting	2	1	2
Support	1	0.5	1
Satisfied	✓	✓	✓

on the whole data set before evaluating the performance of the model. The information of the test set will be leaked to the training process so that the performance of the model will be wrongly estimated when the new data is predicted.

Therefore, in the first step of pattern mining, this paper divides the training set and test set and uses the PrefixSpan algorithm to mine the patterns of peptide sequences respectively on positive and negative classes. The second step is to filter the pattern set by the discriminant value. The method chosen in this paper is the contrast threshold, that is, the discriminant value of the preserving patterns has to be larger than the minimal threshold θ. The definition of discriminant is as follows:

$$Disc(s, D) = \frac{Occ(s, D_{positive})}{|D_{positive}|} - \frac{Occ(s, D_{negative})}{|D_{negative}|}, \tag{3.2}$$

where $Disc(s, D)$ refers to the discriminant ability of pattern s to positive and negative classes in a database D. $Occ(s, D_{positive})$ and $Occ(s, D_{negative})$ respectively refers to the mode in positive and negative class frequency, $|D_{positive}|$ and $|D_{negative}|$ refers to is set and the size of the counter example set. Only patterns that satisfy the condition of $Disc(s, D) \geqslant \theta$ will be reserved.

Then, the training set and test set are respectively converted to the feature vector based on the reserved frequent patterns. Like k-mer, the sequence is transformed by the exis-

Table 3.6 The projected database on second level of recursion.

$D\|_a$	Prefix	Projected database
$D\|_{\langle I\rangle}$	$\langle I\rangle$	$\langle MK\rangle$,$\langle K\rangle$
$D\|_{\langle M\rangle}$	$\langle M\rangle$	$\langle K\rangle$
$D\|_{\langle K\rangle}$	$\langle K\rangle$	$\langle\rangle$

tence response. The difference here is that the object being checked is the subsequence of a sequence. Therefore, this paper uses an algorithm to detect whether a sequence contains subsequences. Finally, the feature vector of the training set and test set are generated respectively.

3.1.2 Decision Rule Sets Learning Model to Predict Peptide Detectability

In this section, how to learn and train decision sets according to the features extracted from the peptide sequence is introduced.

First, this paper was carried out in accordance with the method of IDS, but after the experiment, it found that IDS is not able to be applied to large data sets. Once the number of features is slightly increased, the memory consumption of IDS will increase sharply. Meanwhile, the training time is too long. Through literature research, this study pointed out that the IDS algorithm does have this problem in other studies[33]. Finally, this paper employed the method named Decision Rule Network, which is proposed by Qiao[34], a neural network-based training algorithm for an interpretable Decision set model. The model is described as follows:

(1) Decision Rule Network

The Decision Rule Network is focusing on training a Boolean classifier based on binarized input features. The formal definition is:

For training set $(X_n, y_n), n = 1, \ldots, N$, where N is the number of data samples and X_n is consists of D dimensions features $X_{n,i} \in \{0, 1\}, i = 1, \ldots, D$, and labels $y_n \in \{0, 1\}$. The results are the final decision rule set $C = \{c_1, c_2, \ldots, c_m\}$ which is a combination of rules c, in which the definition of rule is same as the Table 2.2 above. If there is no predicate associated with the input characteristics in the rule, this means that feature is excluded from this rule.

The main architect and the workflow of the model can be found in the following figure.

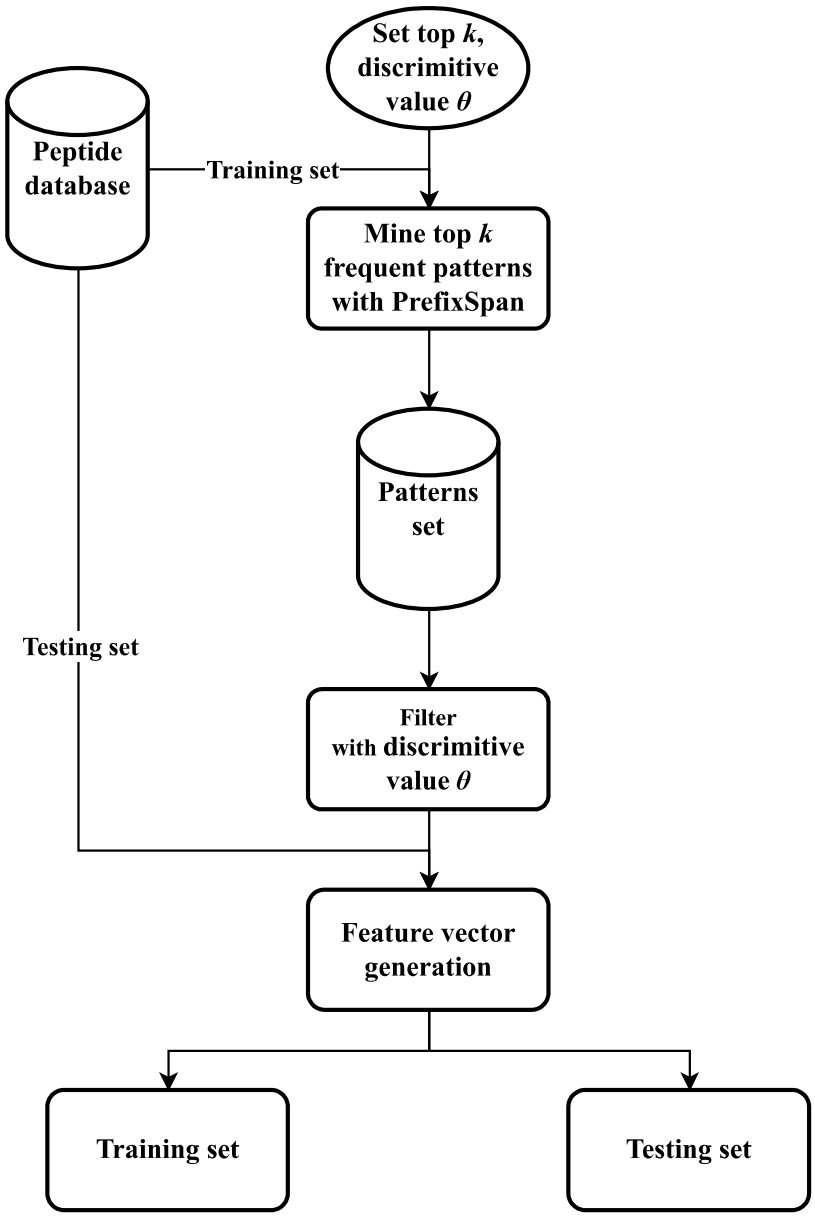


Figure 3.3 Flow chart of mining patterns with PrefixSpan and discriminative value.

As it shows, the network is consisting of two layers, which are the rule layer and the OR layer. The rule layer is aimed to train the neurons to generate the rules for the decision set, and the OR layer is focusing on disjunction and selecting the rule generated by the first layer.

For a decision rule network, the input features must be binary vectors, but for most

of the data, there are numerical features. Therefore, the preprocessing of input features is very important. Decision Rule Network adopts quantile discretization to obtain a set of thresholds for each feature. In this way, features are transformed into input vectors. However, this process does not exist for the existing response feature data in this paper, so the features can be simply input into the Decision Rule Network in the form of one-hot encoding. For example,

$$existence\ EE, not\ existence\ DK, existence\ LD \ldots, existence\ FDR,$$

can be represented as a feature vector $1, 0, 1, \ldots, 1$. In this way, the feature vector can be the input of the first layer.

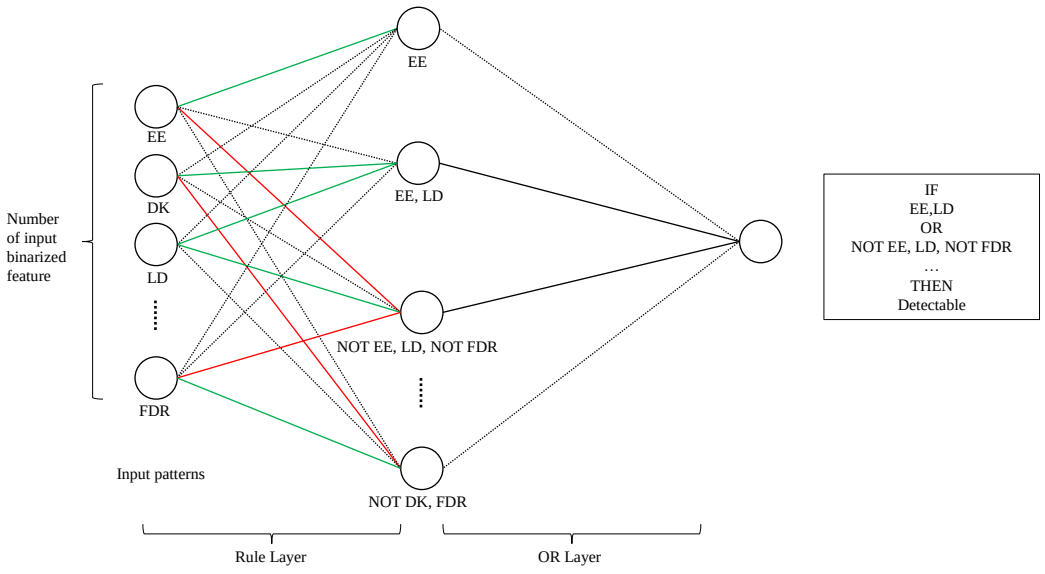


Figure 3.4 Workflow in decision rule network of generating rules for peptide prediction.

Rule layer in Decision Rule Network is focusing on extracting rule c by training the neurons with binary step activation function. In this way, the output of the first layer can be mapping into a rule set $C = \{c_1, c_2, \ldots, c_m\}$ in which the rule c_i is connected by the

predicate “AND”. The operation of the neuron is described as follows:

$$y = \sum_{i=0}^{D} w_i x_i - \sum_{w_i>0} w_i + 1. \tag{3.3}$$

As for the logical AND, the binary step function is applied:

$$f(x) = \begin{cases} 1 & \text{if } x = 1, \\ 0 & \text{otherwise.} \end{cases} \tag{3.4}$$

For this equation, it ensures that only when the input is equal to 1 does the neuron will be activated. In this case, the neuron can be mapping into the logical IF-THEN statement, in which the positive weights indicate that the input feature is a positive association, vice-versa.

To make the activation function in the rule layer differentiable, the gradients are computed with the following equation:

$$g_{\hat{y}_i} = \begin{cases} 0 & \text{if } y_i < 0 \text{ or } y_i > 1 \frac{\partial L}{\partial y_i} < 0, \\ g_{y_i} & \text{otherwise ,} \end{cases} \tag{3.5}$$

in which $g_{\hat{y}_i}$ and g_{y_i} are the gradients of loss.

As for the OR layer, only one output is inclded, while the weights needs to be a binary function:

$$\hat{w}_i = \begin{cases} 0 & \text{if } w_i \leq 0, \\ 1 & \text{otherwise .} \end{cases} \tag{3.6}$$

By applying the negative bias $-\epsilon$ on the output neuron, the dot product formation is:

$$y = \sum_{i=1}^{D} \hat{w}_i x_i - \epsilon. \tag{3.7}$$

The existence of the bias made the output neuron become an OR gate, which means that as long as there is one input that is activated, then the output value y will be the positive value.

To minimize the rule number, a method similar to the L_0 regularization was proposed to maximize the sparsity. With the maximum sparsity, the number of neurons that can be activated will be minimized. Thus, it also reduces the number of rules. In order to eliminate the input feature, the author makes all the weights of the parameter be replaced by the product with the corresponding mask variables.

The author also applied the method of a two-phase training strategy to reduce the difficulty of training. In the first phase, the OR layer is frozen, and the module is focusing on training the rule layer, while in the second phase, the rule layer is frozen to optimize OR layer. This is an effective approach, reflected in the process of training. First, search for the best-matched feature, and then select the most appropriate rule, move in circles.

In the module of the training decision rule set, the model takes in the feature generated from the sequential patterns mining module and provides the rule for classification.

4 Experiment and Discussion

In this section, the effectiveness of the Interpretable Peptide Detectability Prediction Model was validated, and details like setting parameters and the performance of the model are discussed. Besides, it also provides a comparison between the Interpretable Peptide Detectability Prediction Model and the state-of-the-art algorithm. The model itself is implemented in Python, it was then tested on an AMD Ryzen 5 5600X CPU with 16 GB DDR4 memory. To limit the running time, the features mined in the first module should be limited, but also because of its memory consumption.

4.1 Setup

4.1.1 Datasets

The model was trained using the peptide data of Homo sapiens and Mus musculus, in which both datasets contain 90000 pieces with the label of half positive and half negative of peptide sequence data. As described in the research[4], the datasets were constructed from the GPMDB database. The peptide sequence was ranked according to the observation times, then they were selected the top-N was positive and the last N sample was negative.

4.1.2 Peptide Sequential Pattern Mining Module

On the peptide sequential pattern mining module, the parameter that needs to be set is the k number for k-mers and the top-k for mining the number of patterns with the method of PrefixSpan. This paper experimented with determining the parameters of these two approaches, the detail of the experiment will be introduced below.

For the part of k-mer, the mined pattern number is mainly because of the setting parameter k. It is known that the number of k-mers is versus with k, and its distribution is unimodal. To guarantee the model accuracy and limit the time consumption, the setting parameter for featurization is very important. Figure 4.1 shows the number of the pattern in the peptide sequence of two species versus with k.

It can be found in Figure 4.1 that the number of k-mers increases sharply with the increase of k value. What's more, tens of thousands of features are not suitable for training and testing. More importantly, this is not good for module interpretability. Thus, it is necessary to control the number of k-mers selected for the next module. With the elbow method,

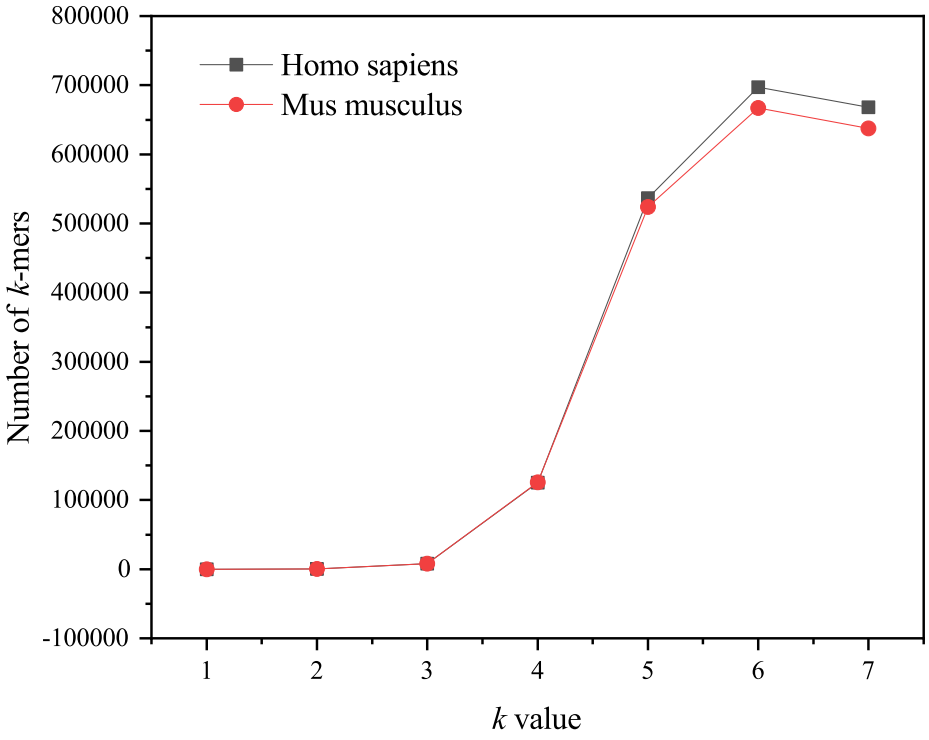
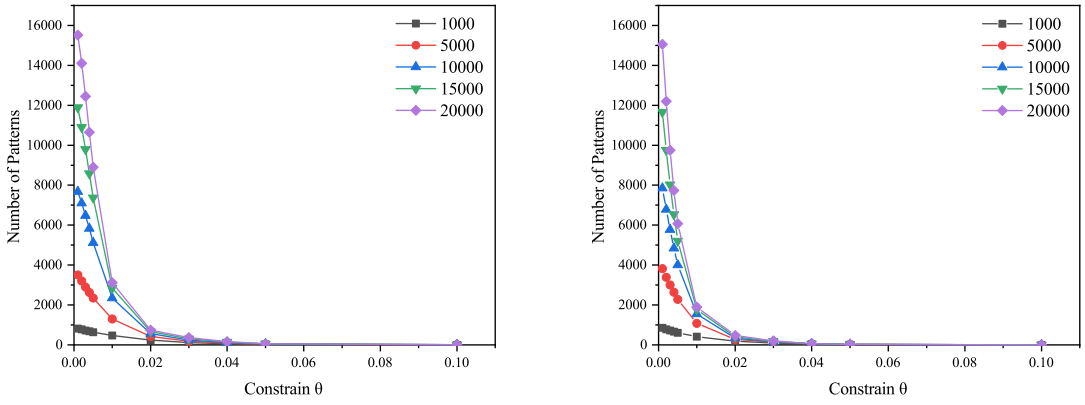


Figure 4.1 Number of k-mers versus k value.

it can be regarded that $k = 3$ is the best choice as it is the changing point. However, when $k = 3$ there are 7997 for Homo sapiens and 8005 for Mus musculus, this number is beyond the limitation because it will need nearly 32 GB of memory. What's more, when putting the feature matrix into the classifier, the classifier even cannot converge, and the sparsity will increase to 1 rapidly. That means, the classifier cannot learn any useful information from the feature. So, $k = 2$ is considered to be the setting parameter.

As for PrefixSpan, the part of setting parameters is more complex because it has two parameters that need to be confirmed, which are the threshold value θ, and the Top-k. Figure 4.2 shows the trend of increasing the number of pattern changes with the decreasing the threshold θ under five conditions. Five conditions of top-k were tested on two species respectively on the positive set and negative set. As it can be found in Figure 4.2, it can be regarded that the pattern number is increasing at an exponential rate, and more importantly, 0.02 is an elbow point. With the threshold θ getting smaller, the potential pattern number

increases sharply, while $\theta = 0.02$ is considered to be a reasonable choice as it remains a certain amount of discriminant patterns for classification. If $\theta = 0.01$, the first module will produce more than thousands of features, which does not show a distinct performance difference compared with 0.02. Nevertheless, it will lead to larger difficulty in training the module, especially the aspect of time consumption. To maximize the discriminant feature number, here $k = 20000$ is chosen.



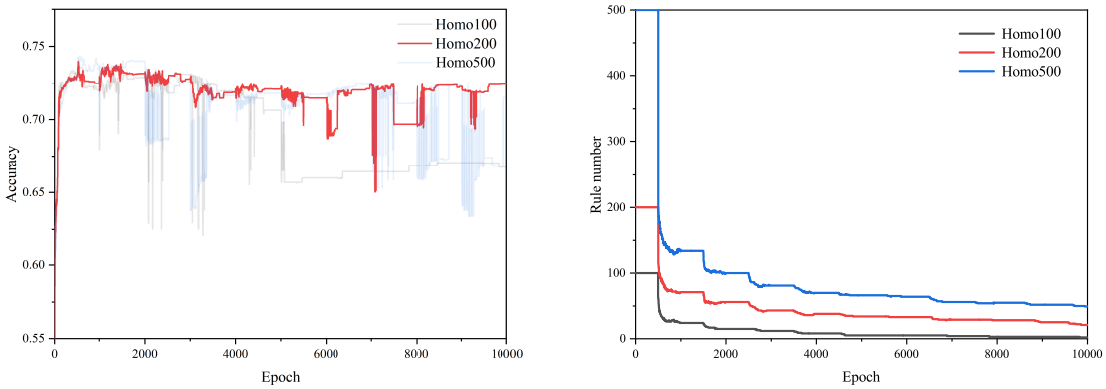
(a) Homo sapiens, (b) Mus musculus,

Figure 4.2 Number of patterns versus constrain θ.

4.1.3 Decision Set Learning Module

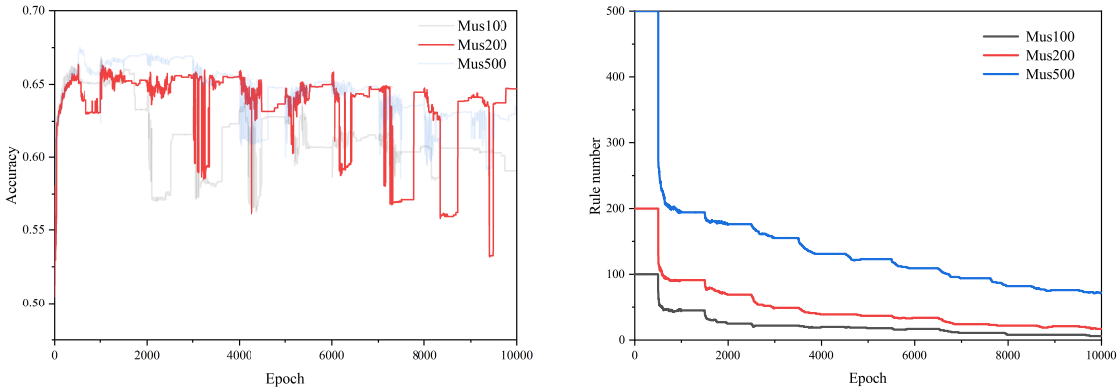
For the module of the learning decision set, the method that this paper adopts needs to set mainly just the epochs and the initial rule numbers. The working mechanism of the Decision Rule Network sets an initial rule number first as its neuron number in the output of the Rule Layer. With the training process, some neurons will not be considered as a utility one as the regularization term, finally, the rules will be reduced to a short number that guarantees some classification capability. Thus, the initial rule number indicates the range of the searching space and also influences the final performance of the model. As for the epochs, this hyperparameter influences the times that the algorithm work through training datasets. However, this parameter is connected to the degree of regularization. In other words, larger epochs number will result in longer time training. In the meantime, the regularization term affects the sparsity and makes the rule number reduced.

With the experiment result, $rule\ number = 200$ is determined to be the best parameter.



(a) Accuracy versus epoch. (b) Rule number in decision set versus epoch.

Figure 4.3 Experiment results with different number of rule number on Homo sapiens.



(a) Accuracy versus epoch. (b) Rule number in decision set versus epoch.

Figure 4.4 Experiment results with different number of rule number on Mus musculus.

For one reason, there exist nearly 400 features, so the initial potential rule should be a small number. However, if the searching space is too limited, the model may not find the best rule for classification, this phenomenon happened when the initial feature number is equal to 100. As it can be found in the Figure, its accuracy is significantly lower than other conditions. Reciprocally, if the initial searching space is set to be very large, there also exist drawbacks. Taking the condition of 500 initial rules for example, even with 10000 epochs, there are still about100 rules, which means that the model is not interpretable. To obtain a briefer result, the model needs to do more pruning and optimization, implying that extra time is needed. Generally, the potent rule in a peptide should be brief due to the natural

mechanism, with 200 as the initial rule number, the model shows the best performance with brief rules.

4.2 Results and Evaluation

In this section, the detail of the model result is discussed, and this section also provided a comparison between the main current algorithms.

4.2.1 Accuracy of Classification

The detail of the classification performance of the model is shown in Figure 4.1 and 4.2. In the PrefixSpan module, the sequential patterns mined for the next step are 20000 for both positive and negative respectively, and then the discriminant threshold θ is set to 0.02. As for the decision rule learning module, the initial rule is set to 200, the training epoch for 2-mer-DRN is set to be 1000 due to its convergence speed, and 10000 for PrefixSpan.

Table 4.1 Experiment results on Homo sapiens.

Models	ACC	SP	SN	MCC
iBCM+RF	0.6360	0.6249	0.6445	0.2741
AP3 [1]	0.6416	0.5949	0.6881	0.2843
2-mer-DRN	0.6800	**0.7664**	0.5973	0.3692
SeqDT	0.7151	0.7089	0.7213	0.4345
PrefixSpan-DRN	0.7201	**0.7873**	0.6470	0.4414
PepFormer [1]	0.8066	0.7213	0.8915	0.6221

[1] The comparison result are from the work of Pepformer[4].

From the experiment results, it can be regarded that the model that this paper proposed can reach the state-of-the-art performance while keeping interpretable. It's worth noting that when the model is trained on the dataset Mus musculus, the accuracy of the model is nearly 0.65, but if the rules are pruned to lean, the model accuracy will then converge to around 0.64. This may be explained that the way of obtaining the feature is not enough for extracting the inner mechanism of the biological sequence. Thus, large numbers of rules are necessary to be used to ensure the performance of the model. More importantly, PrefixSpan-DRN has the highest specificity, which means that this model is very sensitive to the negative, which means it can discover nearly the majority of the peptide sequence

Table 4.2 Experiment results on Mus musculus.

Models	ACC	SP	SN	MCC
iBCM+RF	0.5767	0.6190	0.5215	0.1573
2-mer-DRN	0.5956	**0.8864**	0.3047	0.2349
PrefixSpan-DRN	0.6447	**0.8127**	0.4799	0.3099
AP3 [1]	0.6462	0.5993	0.6928	0.2934
SeqDT	0.6537	0.6467	0.6568	0.3035
PepFormer[1]	0.7521	0.6421	0.8629	0.5176

[1] The comparison result are from the work of Pepformer[4].

which are undetectable. From another perspective, the interpretable model itself provides a chance for researchers to be able to find the mechanism of peptide discovery and thus improve the performance of analyzing mechanism.

4.2.2 Cross-Species Transfer Accuray

To evaluate the cross-species transfer learning ability of this model, this paper did the experiment only change the epoch number, as it does not affect the performance very much, it mainly influences the output rule number. Firstly, one dataset is used to train, and then the trained model is tested on the other dataset to measure its cross-species transfer performance. Table 4.3 shows the result of the experiment, and it also provides a comparison with the latest algorithms.

Table 4.3 Cross-species transfer comparison.

Accuracy	AP3	**PrefixSpan-DRN**	Pepformer
Train on Mus, test on Homo	0.5221	0.7011	0.7917
Train on Homo, test on Mus	0.5049	0.6556	0.7451

From the table, it can be concluded that the model proposed in this paper shows a satisfactory performance on the cross-species transfer task. It can be found in the table that when the model was trained on Homo sapiens and tested Mus musculus, the accuracy is higher than even trained and test on the Mus musculus with even fewer rules. Research pointed out that this may due to the learning ability of the model. As for Prefixspan-DRN,

the model can reach the same level as the model that trains and test on the same dataset, but after pruning the rules, the accuracy will drop a little. This may be because of the complexity of the dataset. This can also be proved when operating the first module. In the first module, after fliting the pattern that mined both positive and negative patterns, the remaining patterns of Mus musculus only reach half of Homo sapiens. Besides, when the model trained on Mus musculus and tested on Homo sapiens, the model also reached the performance of training and testing on the Homo sapiens only. This performance may provide an important insight that can help people understand the complex principle of whether the peptide sequence is detectable or not.

4.2.3 Interpretable Decision Sets

This section provides an analysis of the decision rules learned from the section module. Table 4.4 provided the details of the experiment results of interpretable decision sets.

Table 4.4 Details of rules in decision sets.

Models	Rule numbers	Rule length	N. conditions[1]	P. conditions[2]	Accuracy
Homo100	6	10.33	50	12	0.6483
Homo200	17	11.76	168	32	0.7201
Homo500	49	10.53	372	144	0.7150
Mus100	2	5.5	7	4	0.5184
Mus200	22	8.45	130	56	0.6447
Mus500	94	14.11	453	309	0.6050

[1] Negative conditions.
[2] Positive conditions.

As it can be found in the table, nearly twenty rules are enough for the model to make a considerable accurate decision for predicting the detectability of peptide sequence. The experiment result of cross-species measurement, also proved that the rules in two datasets are having the same efficiency. In addition, the negative conditions in the dataset are far more than the positive conditions. In other words, these negative conditions may be the most important features that will lead to the undetectable character of the peptide sequence.

4.2.4 Future Work

Though this model reaches state-of-the-art performance, it does have some defects. Firstly, some peptide sequences cannot be represented using the current way. For example, the sequence representation does not consider that the peptide actually has a three-dimensional structure. Thus, some important information cannot be discovered in this way, specifically, the disulfide bond in Figure 1.2 is being ignored. Therefore, a way that can store abundant information needs to be developed. The section point is that the model takes lots of time for convergence. In order to prune the rules in the decision set, the model takes lots of time, while the model will only output about twenty rules. This phenomenon may be improved by changing the way of mining features. For the model to make a decision, only several powerful features are necessary, while even with a discriminated threshold, there still exists hundreds of features. Moreover, when mining the sequential pattern, the model does not concern with the gap constrain. In this condition, even if the powerful sequential patterns were mined, the existence feature vector will also be generated incorrectly as the distance at which amino acids interact is finite. Under these conditions, the way of mining sequential patterns is urgently needing to be improved, and it is the direction to be considered next in this following work.

Conclusion

In proteomics, the detectability of peptide sequence is an important problem, as the detection accuracy of peptide sequence directly affects the correctness of protein identification. However, due to the complex sampling process in the experiments, the randomness in identification leads to the recurrence problems of mass spectrometer results. Therefore, predicting the detectability of peptides is a key problem in proteomics.

However, much of the recent research focus on the accuracy of the model, but rarely considers its interpretability of the model. Retaining interpretability when predicting the detectability of the peptide sequences is beneficial for understanding the detection process and providing a reference for optimization of the experiments.

Based on the above research and problems, this paper proposed an effective and interpretable peptide sequence detectability prediction model named PrefixSpan-DRN based on sequential pattern mining techniques and Decision Rule Network. In this model, PrefixSpan sequential pattern mining is used to extract sequential patterns, subsequently, a contrast threshold is set to filter the effective discriminant patterns. Finally, feature vectors are generated according to the existing constraints of the remaining patterns. Then the Decision Rule Network method is used to train and generate decision sets. Experimental results show that PrefixSpan-DRN can achieve the classification accuracy of the current mainstream algorithms under the premise of ensuring the interpretability of the model.

Reference

[1] WASINGER V C, CORDWELL S J, Cerpa-Poljak A, et al. Progress with gene-product mapping of the Mollicutes: Mycoplasma genitalium[J]. Electrophoresis, 1995, 16(1): 1090-1094.

[2] CATHERMAN A D, SKINNER O S, KELLEHER N L. Top Down proteomics: Facts and perspectives[J]. Biochemical and Biophysical Research Communications, 2014, 445(4): 683-693.

[3] GAO Z, CHANG C, YANG J, et al. AP3: An Advanced Proteotypic Peptide Predictor for Targeted Proteomics by Incorporating Peptide Digestibility[J]. Analytical Chemistry, 2019, 91(13): 8705-8711.

[4] CHENG H, RAO B, LIU L, et al. PepFormer: End-to-end transformer-based siamese network to predict and enhance peptide detectability based on sequence only[J]. Analytical Chemistry, 2021, 93(16): 6481-6490.

[5] TANG H, ARNOLD R, ALVES P, et al. A computational approach toward label-free protein quantification using predicted peptide detectability[J]. Bioinformatics, 2006, 22: e481-8.

[6] GURUCEAGA E, Garin-Muga A, PRIETO G, et al. Enhanced Missing Proteins Detection in NCI60 Cell Lines Using an Integrative Search Engine Approach[J]. Journal of Proteome Research, 2017, 16(12): 4374-4390.

[7] KAWASHIMA S, POKAROWSKI P, POKAROWSKA M, et al. AAindex: Amino acid index database, progress report 2008[J]. Nucleic Acids Research, 2007, 36(Database): D202-D205.

[8] SERRANO G, GURUCEAGA E, SEGURA V. DeepMSPeptide: Peptide detectability prediction using deep learning[J]. Bioinformatics, 2020, 36(4): 1279-1280.

[9] YU M, DUAN Y, LI Z, et al. Prediction of Peptide Detectability Based on CapsNet and Convolutional Block Attention Module[J]. International Journal of Molecular Sciences, 2021, 22(21): 12080.

[10] CRAIG R, CORTENS J P, BEAVIS R C. Open Source System for Analyzing, Validating, and Storing Protein Identification Data[J]. Journal of Proteome Research, 2004, 3(6): 1234-1242.

[11] SRIKANT R, AGRAWAL R. Mining sequential patterns: Generalizations and performance improvements[C]. APERS P, BOUZEGHOUB M, GARDARIN G. Advances in Database Technology — EDBT '96. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996: 1-17.

[12] JI X, BAILEY J, DONG G. Mining minimal distinguishing subsequence patterns with gap constraints[J]. Knowledge and Information Systems, 2007, 11(3): 259-286.

[13] FOURNIER VIGER P, LIN C W, RAGE U, et al. A survey of sequential pattern mining[J]. Data Science and Pattern Recognition, 2017, 1: 54-77.

[14] ZAKI M J. Sequence mining in categorical domains: Incorporating constraints[C]. CIKM '00: Proceedings of the Ninth International Conference on Information and Knowledge Management. New York, NY, USA: Association for Computing Machinery, 2000: 422-429.

[15] PEI J, HAN J, WANG W. Constraint-based sequential pattern mining: The pattern-growth methods[J]. Journal of Intelligent Information Systems, 2007, 28(2): 133-160.

[16] JABBOUR S, MANA F E, DLALA I O, et al. On Maximal Frequent Itemsets Mining with Constraints[M]. HOOKER J. Principles and Practice of Constraint Programming: volume 11008. Cham: Springer International Publishing, 2018: 554-569.

[17] DE SMEDT J, DEEVA G, DE WEERDT J. Mining Behavioral Sequence Constraints for Classification[J]. IEEE Transactions on Knowledge and Data Engineering, 2020, 32(6): 1130-1142.

[18] XING Z, PEI J, KEOGH E. A brief survey on sequence classification[J]. ACM SIGKDD Explorations Newsletter, 2010, 12(1): 40-48.

[19] DONG G, PEI J. Advances in database systems: volume 33 sequence data mining[M]. Kluwer, 2007.

[20] KEOGH E, KASETTY S. On the need for time series data mining benchmarks: A survey and empirical demonstration[C]. KDD '02: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Association for Computing Machinery, 2002: 102-111.

[21] KEOGH E J, PAZZANI M J. Scaling up dynamic time warping for datamining applications[C]. KDD '00: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, NY, USA: Association for Computing Machinery, 2000: 285-289.

[22] MACLEOD M D. 14 - coding[M]. MAZDA F. Telecommunications Engineer's Reference Book. Butterworth-Heinemann, 1993: 14-1-14-13.

[23] MOLNAR C. Interpretable Machine Learning[M]. 2021.

[24] COHEN W W. Fast Effective Rule Induction[M]. PRIEDITIS A, RUSSELL S. Machine Learning Proceedings 1995. San Francisco (CA): Morgan Kaufmann, 1995: 115-123.

[25] QUINLAN J R, Cameron-Jones R M. FOIL: A midterm report[C]. ECML '93: Proceedings of the European Conference on Machine Learning. Berlin, Heidelberg: Springer-Verlag, 1993: 3-20.

[26] CLARK P, NIBLETT T. The CN2 induction algorithm[J]. Machine Learning, 1989, 3(4): 261-283.

[27] LI W, HAN J, PEI J. CMAR: Accurate and efficient classification based on multiple class-association rules[C]. Proceedings 2001 IEEE International Conference on Data Mining. 2001:

369-376.

[28] YIN X, HAN J. CPAR: Classification based on predictive association rules[M]. Proceedings of the 2003 SIAM International Conference on Data Mining (SDM). 2003: 331-335.

[29] LAKKARAJU H, BACH S H, LESKOVEC J. Interpretable Decision Sets: A Joint Framework for Description and Prediction[C]. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco California USA: ACM, 2016: 1675-1684.

[30] DASH S, GUNLUK O, WEI D. Boolean decision rules via column generation[C]. BENGIO S, WALLACH H, LAROCHELLE H, et al. Advances in Neural Information Processing Systems: volume 31. Curran Associates, Inc., 2018.

[31] WANG T, RUDIN C, Doshi-Velez F, et al. A bayesian framework for learning rule sets for interpretable classification[J]. Journal of Machine Learning Research, 2017, 18(70): 1-37.

[32] Jian Pei, Jiawei Han, Mortazavi-Asl B, et al. PrefixSpan,: Mining sequential patterns efficiently by prefix-projected pattern growth[C]. Proceedings 17th International Conference on Data Engineering. Heidelberg, Germany: IEEE Comput. Soc, 2001: 215-224.

[33] YANG F, HE K, YANG L, et al. Learning interpretable decision rule sets: A submodular optimization approach[J]. Advances in Neural Information Processing Systems, 2021, 34.

[34] QIAO L, WANG W, LIN B. Learning accurate and interpretable decision rule sets from neural networks[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2021, 35 (5): 4303-4311.

Modification Record

(1)Graduation project (thesis) content important modification record

Modify the record for the first time:

Page 1, Figure 1.1,

Before modification:

Lack of detailed description of the whole experiment process.

After modification:

Added a description of the entire bottom-up process.

Modify the record for the first time:

Page 24, Equation 3.2,

Before modification:

$$Disc(s, D) = \frac{Occ(s, D_{positive}) - Occ(s, D_{negative})}{|D|}$$

After modification:

$$Disc(s, D) = \frac{Occ(s, D_{positive})}{|D_{positive}|} - \frac{Occ(s, D_{negative})}{|D_{negative}|}$$

第三次修改记录：

Page 31, Figure 4.1; Page 32, Figure 4.2; Page 33, Figure 4.3, 4.4.

Before modification:

Does not accord with the paper picture format, at the same time, there is an error in the name.

After modification: Added the outer border of the picture, modified the format of the picture, and corrected the name error.

（2）Graduation project (thesis) formal detection of repetition ratio：10 %

Note-taker (Signature)：董俊杰

Supervisor (Signature)：何炳蔚

Acknowledgement

Firstly, I want to acknowledge the painstaking care of Professor He, he leads me the way to the world of academics. When I decided to switch my major to software engineering, it was he that kept telling me not to be hurried and comforting me that he will be standing with me. I know that I am a guy full of anxiety and that is the reason why I frequently seek his help, while he never feels bored. His warm instruction and rigorous academic attitude enlightened me, which make it possible for me to know the methodology of learning and researching. I trust that we will have an unforgettable experience in the future.

I would also like to thank my friends. Shen Jinghan, you are the person who always supports me, cheering me up when I was down, I can't imagine my life without you. My roommates are careful and kind, and it is my honor to meet them, we experienced day and night, and we have to say goodbye, I wish you both a bright future. Little Guan, you will be living a simple life, and little Li, your logical mind will bring you to a new stage. Specially mention for Xia Kaiyang, we together spent our fourth year, you are the person who teach me the social experience most, and I believe that one day, we will meet again and have fun together. I also want to appreciate all members in Chemistry 1801, we are a big warm family and we together tide over difficulties, best wishes to all of you.

Finally, I am grateful to my parent, for their sacrifice in all these twenty years. They lift me, providing me with the best that they have. I know that I did not do well in daily life, bad temper, habits, etc., while you two always tolerate me, understand me, and teach me to be a better person.

2022 is a special year because I am going to say goodbye to my old days, and start my new journey in another major. Remember the first day I came to this university, I speak to myself that I wish to be the best person in whatever the area that I am focused on. No doubt that it is ridiculous, as the world is huge, while I am only an ant-like person. However, I still want to make a difference, at least, I want myself to be useful, and meaningful to this world, even just a little.