uc3m | Universidad Carlos III de Madrid

University degree in Biomedical Engineering
2019-2020

Bachelor Thesis

"Data mining of structural, genomic and proteomic annotations for biological macromolecules"

Paola Núñez Hernández

Supervisor
Mr. Carlos Óscar Sánchez Sorzano PhD.
Tutor UC3M
Arrate Muñoz Barrutia
Leganés, 2nd September 2020

ABSTRACT

Bioinformatics, understood as the application of computational tools on molecular data, has become a very important field. The evolution of the biology and the technology has supposed an exponential growth of the quantity of available data about genome and protein sequences. Because of the differences between these emerging data, there is not a single database that contains all this information.

The application of data mining techniques represents an enormous potential for the analysis of all these data that is accumulating at different databases. To do part of this analysis, it is presented two data mining techniques, association rules and prediction.

To find frequent patterns, manual association and A priori algorithm are applied to data contained in 3Dbionotes, a webserver that collects annotations from biological databases. From these two approaches, different frequent pattern rules have been extracted and revised.

Due to the implication of epitopes in the development of new types of vaccines and the already use of bioinformatics in this field, an epitope predictor is constructed using Natural language processing, based on Bidirectional Encoder Representations from Transformers (BERT) architecture. A model with an accuracy of 85% has been developed after pretraining and fine-tuning.

Key words: Biomedical informatics, Machine Learning, A priori, Association rules, Amino acids, protein, epitopes, vaccines.

AKNOWLEDGMENTS

TABLE OF CONTENTS

1.Introduction **1**

2. Theorethical background **2**

2.1 Proteins *2*

2.2. The immune system and immunotherapy *3*

2.3. Natural language processing *6*

2.3.1 BERT *7*

2.4. Association rules *12*

3. Objectives **15**

4. Materials **16**

4.1 Electron Microscopy Data Bank *16*

4.2 3DBionotes *16*

4.4 Immuno Epitope database and Analysis Resource *17*

4.5 BERT *17*

5. Methods **18**

5.1 Association rules *18*

5.1.1 Manual association *18*

5.1.2 A priori algorithm *20*

5.2 Epitope predictor *21*

6.Results and discussion **25**

6.1 Association rules *25*

6.2 Epitope predictor *29*

7. Conclusion and future work **35**

8. Regulatory framework **36**

9. Socio-economic impact **37**

11.Bibliography **40**

12. Supplementary material (annexes) **1**

12.1 Annex a *1*

LIST OF FIGURES

Fig. 1. (left side) Right-handed alpha helix in detail and (right side) symbolic representation. The dot lines in left side represent the hydrogen bonds that held together the alpha helix (source [7])....2

Fig. 2 Formation of parallel beta sheet and antiparallel beta sheet. The hydrogen bonds between oxygen (O) and hydrogen (H) are represented by the discontinuous lines (source [6])....3

Fig. 3 Graphical abstract of the development of bioinformatics tool for the design of epitope-based vaccines. The process starts at the computer level using the tools designed for epitope prediction and finished with the creation of the vaccine and its assessments (source [18])....6

Fig. 4 Overall architecture of Transformer. The encoder is at the left halved and the decoder is at the right halved of the image (source [22])....8

Fig. 5 Components of a single encoder. First, the input passes through the self-attention layer. The output of this layer is added with the input and normalize. After, it enters the feed forward layer and at the end it is again added and normalized (source [23])....8

Fig. 6 Mechanisms used at self-attention layer of the encoder. The basic architecture at left of the image and the multi-head attention mechanism construct based on several scaled dot-product units in parallel (source [22])....9

Fig. 7 BERT input representation as the sum of the token, segment and position embeddings (source [21])....10

Fig. 8 Summary of pre-training and fine-tuning procedures for BERT. In the pre-training steps, two tasks are performed, masked LM and NSP. In the fine-tuning, depending on the final task, the final layer changes according to question answering (SQuAD), classification or name entity recognition (NER) (source [25])....11

Fig. 9 Workflow of A priori algorithm from the selection of frequent items of length k to frequent itemsets of length k+1 based on the value of the support and the threshold selected (source [30])....14

Fig. 10 Distribution of epitopes length 22

Fig. 11 Transformation of protein sequences into spaced amino acids samples....22

Fig. 12 Division of the data set in the training, validation and test sets according to an 70%, 20% and 10% respectively. 22

Fig. 13 Evolution of accuracy and loss during the pretraining steps. 30

Fig. 14 Confusion matrix for BERT model after 1,800,000 steps of pretraining and 20 epochs of fine-tuning. The 0 means a negative prediction or prediction of a non-epitope sequence and the 1 means the prediction of an epitope sequence. In the upper left corner, the True Negatives; in the upper right corner, the False Positives. In the lower left corner, the False Negatives and in the lower right corner, the True Positives. The colormap indicates the number of predictions made in each case. 32

Fig. 15 Confusion matrix for BERT model after 3,000,000 of pretraining and 20 epochs of fine-tuning. 32

Fig. 16 Confusion matrix for BERT model after 5,000,000 of pretraining and 20 epochs of fine-tuning. 33

LIST OF TABLES

Table 1 Construction of confusion matrix from correct and incorrect predictions of model. 11
Table 2 Candidates to be association rule from the frequent itemset found. 15
Table 3 Technical specifications of the remote server used to train and test the model. 17
Table 4 Contingency table used for the calculation of risk ratio 20
Table 5 Bert based configuration parameters.....23
Table 6 Parameters used on create_pretraining.py.....24
Table 7 Contigency table of mutation's incidence in nucleotide binding 25
Table 8 Contigency table of mutation's incidence in epitopes. 25
Table 9 Contingency table of incidence of mutations in alpha helix, beta strand and turn. 26
Table 10 Contingency table of incidence of mutations in regulatory sites as exposed group. 27
Table 11 Contigency table of the incidence of epitopes in secondary structure types....27
Table 12 Parameters used for the construction of association rules derived from a priori. 28
Table 13 Results of accuracy and loss of bert pretraining. 29
Table 14 Results of fine-tuning after 1,800,000 steps of pretraining.....30
Table 15 Results of fine-tuning after 3,000,000 steps of pretraining.....30
Table 16 Results of fine-tuning after 5,000,000 steps of pretraining.....31
Table 17 Evaluation parameters corresponding to model 1 31
Table 18 Evaluation parameters corresponding to model 2 32
Table 19 Evaluation parameters corresponding to model 3 33
Table 20 Human resources cost 39
Table 21 Materials cost 39
Table 22 Summary 39

LIST OF ABBREVATIONS

AA: Amino acid

API: Application Programming Interface

AUC: Accuracy

BERT: Bidirectional Encoder Representation from Transformers.

DNA: Deoxyribonucleic acid

EMDB: Electron Microscopy Data Bank

FN: False Negative

FP: False Positive

FTP: File Transfer Protocol

IEDB: Immune Epitope Database and Analysis Resource.

JSON: JavaScript Object Notation

LM: Language Modeling

MHC: Major Histocompatibility Complex

NLP: Natural Language Processing

NSP: Next Sentence Prediction

PDB: Protein Data Bank

PREC: precision

RR: Risk Ratio

SARS-CoV: Severe acute respiratory syndrome associated coronavirus

TN: True Negatives

TP: True Positive

UniProt: Universal Protein Resource

1.INTRODUCTION

The success of the Human Genome Project sets the groundwork for the comprehensive analysis of the human genome from an evolutionary past to precision medicine against diseases [1]. Biological data is accumulating at a high pace and grows at increasing levels, due primarily to higher-throughput and lower-cost DNA sequencing. For this reason, the number of biological repositories that have been established to handle this data is rising at ever-faster rates.

Due to the amount of data available and the large amount of data that will be accumulated, it is necessary to analyze these data in a fast and efficient way in order to make the most of the biological knowledge that can be extracted [2]. The emerging bioinformatics area deals with the need to examine and interpret these data.

In the short term, more genes associated with diseases will be identified as a result of the development of new bioinformatics research methods, and new drug targets will be discovered in parallel. Bioinformatics will be used to classify risk genes and to know the pathogenic mechanisms implicated in the disease which will also offer an avenue for the development of targeted therapy [3].

In the longer term, the bioinformatics study of molecular, physiological and clinical data will expose possible harmful drug reactions in individuals through basic genetic examination. In the end, pharmacogenomics (using genetic knowledge to individualize medical treatment) is supposed to contribute to a new age of personalized medicine; patients may provide their own specific genetic profile for an individualized and focused medicine without side effects [4].

2. THEORETHICAL BACKGROUND

2.1 Proteins

Proteins are the most essential macromolecules in the organism, and they function in all biological processes. They serve as catalysts, keep and store other molecules like oxygen, provide immune protection, control growth and differentiation and many other functions within our body [5].

Protein structure

Protein structure exists at four distinct levels. The primary structure refers to the sequence of basic building blocks, which are primarily twenty amino acids that compose the protein. A protein is essentially an amino acids chain connected by peptide bonds.

The secondary structure is represented by the general 3D form of local regions of the protein. It arises from the formation of hydrogen bonds between atoms of the chain. There are two common folds, alpha-helix and beta sheet and one less common, turn helix [6].

An alpha helix is a right-handed helical coil that is bound together by each fourth amino acid by a hydrogen bond (see Fig. 1) [5].

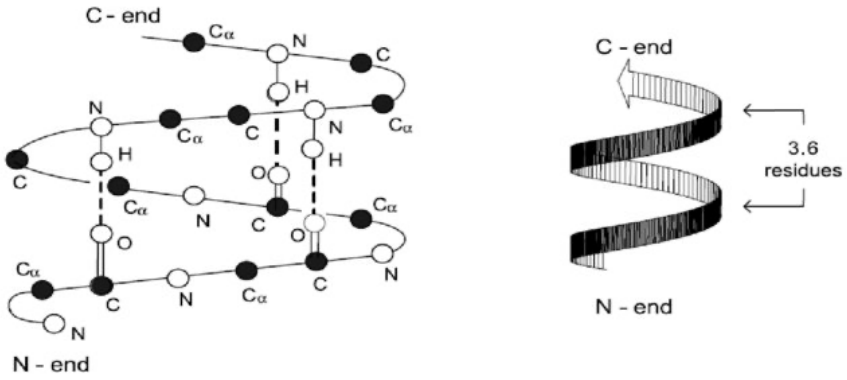

Fig. 1. (left side) Right-handed alpha helix in detail and (right side) symbolic representation. The dot lines in left side represent the hydrogen bonds that held together the alpha helix (source *[7]*).

Beta sheet is the other most common secondary structure. To form this type of structure, two different zones of the chain lie side by side and are bound by hydrogen bonds. There exist two types of beta sheet, parallel beta sheet and antiparallel beta sheet. The end of the chain can be named as N-terminus or C-terminus, depending on whether the amino group or the carboxyl group of the amino acid is free. If the chains are oriented in the

same direction, then it is a parallel beta-sheet. When they run in opposing directions, it is called antiparallel beta-sheet [5].

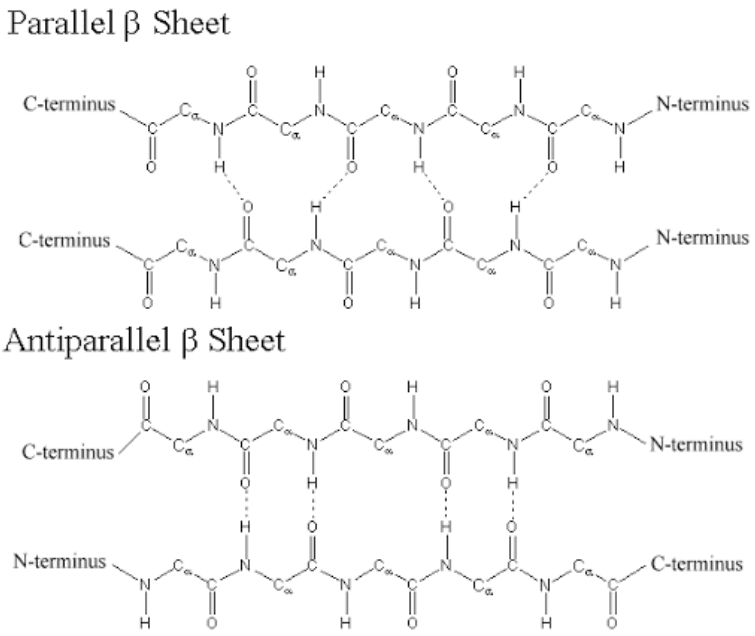

Fig. 2 Formation of parallel beta sheet and antiparallel beta sheet. The hydrogen bonds between oxygen (O) and hydrogen (H) are represented by the discontinuous lines (source *[6]*).

Finally, beta-turns are the third and less common secondary structure present in proteins. It is a region of the chain that involves four consecutive residues, and it results in a fold back on itself [8].

Proteins have different regions or sites of interest that will give us the most import features for understanding each different function and also diseases associated with these zones.

2.2. The immune system and immunotherapy

The immune system is the part of our body that is in charge of preventing or limiting an infection. It protects our body from foreign substances, germs, or viruses. To defend our body, there exist two mechanisms: innate immunity and adaptive immunity.

Innate immunity

The innate immunity is the nonspecific mechanism that the human body has to fight against an infection. It acts against all the substances in the same way and they provide rapid responses. In addition, this mechanism does not have a memory component, it does not remember a previous infection of the foreign molecule.

In the innate immunity, it is found two mechanisms of protection, the first and second line of defense. The first line of defense is composed of the physical barriers of the body, which includes the skin and mucous membranes. The second line of defense is formed by antimicrobial substances, natural killer cells and it includes the inflammation and fever responses [9].

Adaptive immunity

Adaptive immunity refers to the specific defenses. The microbe has breached the innate defenses and it involves a specific recognition and a specific response to it. Substances that are not recognized by the immune systems and generate an immune response are called antigens.

The components of the adaptive immunity are lymphocytes called T cells and B cells. B cells are responsible for producing antibodies. Regarding T cells, it can be made a functional subdivision into cytotoxic and helper. CD8+ cytotoxic T cells attack cells which have become infected with the antigen or tumour cells. CD4+ helper T-cells promote the normal functioning of the rest of immune cells [9].

The major histocompatibility complex is a group of genes whose role is the immunological identification of the cells of the organism and of exogenous cells from invading organisms. They bind to peptide-derived fragments of the pathogens and present them on the cell surface to be recognized by T cells [10]. Depending on where these major histocompatibility complexes are found, they can be of Class I or Class II. Class I MHC are in the plasma membrane of all cells except red blood cells and Class II MHC are localized on the surface of antigen-presenting cells.

Antigen-presenting cells are advanced cells capable of handling a protein antigen, splitting it into peptides, and presenting it on the cell surface along with MHC molecules, where it interacts with the specific T cell. If this is in combination with a Class II MHC, the antigen would be delivered to a CD4+ helper T cell. If it is in conjunction with Class I MHC, the antigen would be presented to a CD8+ cytotoxic T cell [11].

There exist a huge number of antigens in the environment that can provoke an immune response. But, typically, just a small part of the antigen molecules triggers the response. These small parts of the antigen are named as epitopes or antigenic determinants [9].

Epitopes can be classified in two types according to their structure: 1) continuous or linear, and 2) discontinuous or conformational. Continuous epitopes are linear peptide

sequences, usually amphipathic helices[1]. Discontinuous epitopes are structurally more complicated and nonlinear [12].

Immunotherapy

Immunotherapy is a medical treatment that uses the immune system to fight against diseases. Immunotherapy can elicit a change in the immune system work so that it can recognize and kill foreign cells. Inside the immunotherapy, there exists a treatment called therapeutic cancer vaccines that treats cancer by improving the immune system defenses against cancer. These vaccines contain tumor-associated antigens, that the immune system learn to recognize and react against those antigens, destroying cancer cells that contain them.

In general, the vaccines that are currently available are based on a natural form of the pathogen, which is made weakly or inactivate in same form to cause an immune response but not with exactly the same effect as it will cause the pathogen. This type of vaccines is seeking to replicate the response elicited by natural infection [13].

For many years, this has been the most successful, but it has not been able to treat most of cancer types or infectious diseases. Exploitation of sequencing technologies has led to a new concept in the development of vaccines. Epitope-based vaccines can be an alternative because isolated epitopes can have the ability of stimulate a specific immune response.

Bioinformatics and vaccine design

Reverse vaccinology is a method that employs bioinformatics techniques to recognize structures virus, cancer cells, allergens that may cause an immune response that can defend against a particular disease *[14]*. Bioinformatics approaches and algorithms can contribute to the design of epitope-based vaccines. The workflow that the majority of research follows is presented in Fig. 3.

Several methods have been proposed for the prediction of epitopes. Conformational Epitope Prediction server predicts epitopes based on the 3D structure data of protein antigens and the solvent accessibility of the amino acids. It has an accuracy of 75% [15].

[1] An amphipathic helix describes an alpha helix which has one side of the helix with hydrophilic amino acids and the other surface is composed of hydrophobic amino acids.

BepiPred-2.0[2] is a web server for the prediction of B-cell epitopes from antigen sequences [16]. All these tools available are based on Machine Learning techniques. Position-specific scoring matrices (PSSMs), support vector machines (SVMs), hidden Markov models (HMMs), or artificial neural networks (ANNs) are some of the most common ML techniques employed by the several epitope predictor available [17].

For many years, the use of bioinformatics for the development of models for the prediction of epitopes has been an initial step in the design of vaccines. In this bachelor thesis, natural language processing with BERT architecture is applied to the prediction of epitopes.

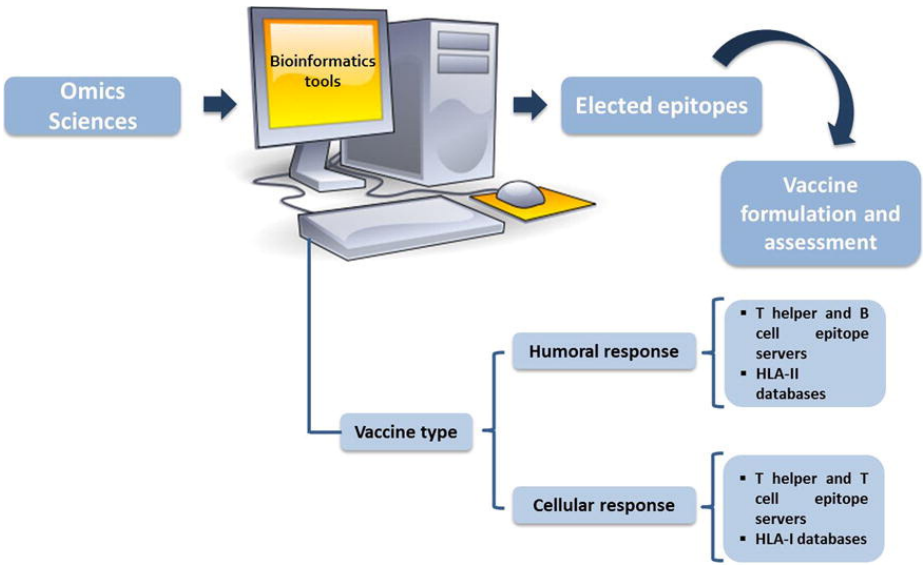

Fig. 3 Graphical abstract of the development of bioinformatics tool for the design of epitope-based vaccines. The process starts at the computer level using the tools designed for epitope prediction and finished with the creation of the vaccine and its assessments (source *[18]*).

2.3. Natural Language Processing

Natural language processing (NLP) is a discipline of Artificial Intelligence that gives the technology the ability to understand and read human languages. It has become particularly important in the last years in the biomedical industry because of its many applications.

Each protein, in its primary structure, can be thought as a chain of amino acids. But the representation of these amino acids is just a chain of discrete characters, then the language models that are used in natural language processing can be applied to them.

[2] The web server for BepiPred-2.0 can be found at this link: http://www.cbs.dtu.dk/services/BepiPred/

Some researchers have tried to introduce natural language processing to the field of bioinformatics in different ways. BioBERT is a pre-trained language model for the biomedical domain based on the same architecture of BERT. It has been used as a model for biomedical text mining [19]. BERTology makes used of the attention mechanism of BERT not only to learn about protein sequence amino acids, but also to capture global properties such as binding sites and tertiary structure of protein [20].

To accomplish the purpose of this thesis, an epitope predictor using BERT has been developed.

2.3.1 BERT

BERT, which stands for Bidirectional Encoder Representations from Transformers, is a model used in NLP. It was published in a paper from the researchers of Google AI Language [21].

In BERT, a model is first pre-trained with data that is not labeled. Once finished, the output of the pre-trained model is a dense representation of the input. To generate the final model with the specific purpose, the model is modified by simply adding a dense layer at the end. Then, the model is retrained with the data and labels specific to the task.

Transformers and model architecture

Transformers are a novel neural network architecture based on a self-attention mechanism. They just execute a limited number of steps. In each step, a transformer applies a mechanism of self-attention which, irrespective of the position, directly establishes the relationships between words in a sentence [22].

Neural networks are formed by an encoder and a decoder. The encoder reads and produces a representation of the input sentence. Then, the decoder produces the output sentence word by word while taking into account the representation that the encoder has done [22]. In the case of BERT, since the goal is the generation of a language representation model, it only needs the encoder part.

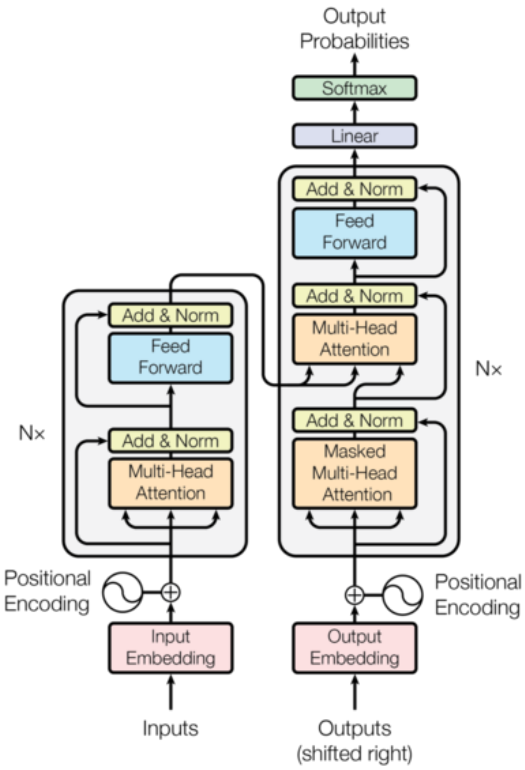

Fig. 4 Overall architecture of Transformer. The encoder is at the left halved and the decoder is at the right halved of the image (source *[22]*).

The encoding component is formed of a stack of 6 identical encoders. In each encoder, there are two sublayers. The input first passes through the self-attention layer, and the outputs are passed to a feed-forward neural network. After these sublayers, there are a residual connection and a normalization layer. That is, the output of each sublayer is *LayerNorm (x + Sublayer(x))* where *Sublayer(x)* is the function created by the sublayer itself [22].

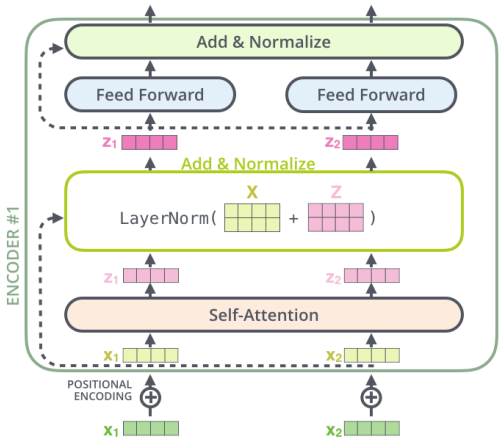

Fig. 5 Components of a single encoder. First, the input passes through the self-attention layer. The output of this layer is added with the input and normalize. After, it enters the feed forward layer and at the end it is again added and normalized (source [23]).

An attention function can be defined as mapping a query and a series of key-value pairs to an output where the query, keys values and output are all vectors or matrices. The output is calculated as the weighted sum of the values, where the weight assigned to each value is calculated by the compatibility function of the query with the corresponding key [22].

The Transformer's basic building blocks are scaled dot-product attention units. The input consists of queries and keys of dimension d_k, and values of dimension d_v. The dot products of the query with all keys are computed. They are divided each by $\sqrt{d_k}$ and after that, it is applied a SoftMax function to obtain the weights on the values [22]. The SoftMax function turns a vector of K real values into a vector of K real values that add 1. In this form, the values can be interpreted as probabilities [24].

In the real practice, the input embeddings are packaged into a matrix, and it is multiplied by the weight matrices that are being trained to get the Q, K and V matrices. The output matrix which corresponds to the attention matrix, is calculated as

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})\mathrm{V} \quad (6)$$

Rather than only computing the attention once, the multi-head mechanism runs through the scaled dot-product attention multiple times in parallel. The attention outputs are concatenated and linearly transformed into the proper dimensions.

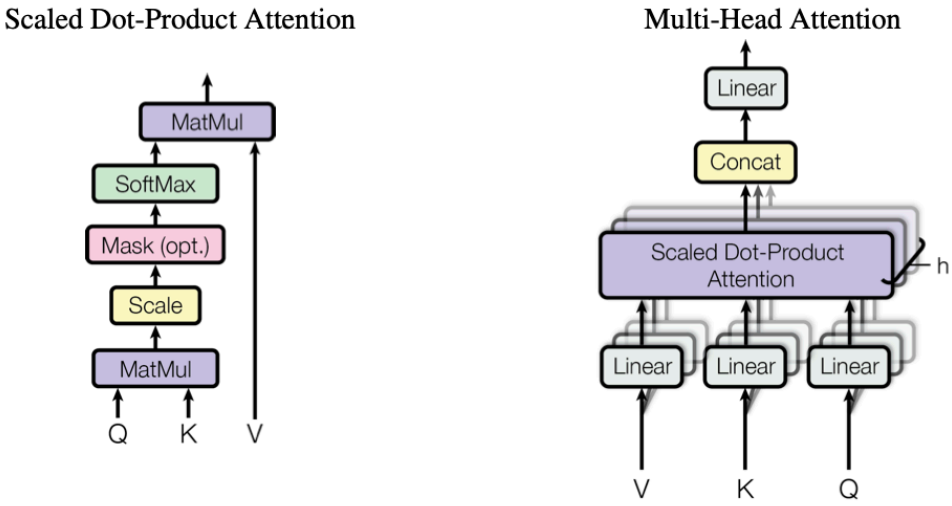

Fig. 6 Mechanisms used at self-attention layer of the encoder. The basic architecture at left of the image and the multi-head attention mechanism construct based on several scaled dot-product units in parallel (source *[22]*).

Input representation

Before starting the pre-training, the input data is converted into the sum of three different embeddings: token embeddings, segment embeddings and position embeddings. The token embeddings transform the words into vector representations of a fixed dimension.

The input text before it passes to the Token embedding is tokenized. Tokenization is the form of separating a piece of text or a word into smaller units that can be word, sub-words or characters. Extra tokens are added at the beginning ([CLS]) and end ([SEP]) of the tokenized sentence. The [CLS] token is a special one used for classification token. If the input has two sentences, as is the case of Question Answering tasks, both pair of sentences are divided by the SEP token.

Segment embedding tokens are also used to separate both sentence A and B in case there are two sentences for the question answering tasks. They are just learned embeddings E_A or E_B concatenated to every token of sentence A or B depending. Finally, the position embeddings represent the location of each token in the input. A visualization of the construction of the input representation in BERT can be seen in the *Fig. 7*.

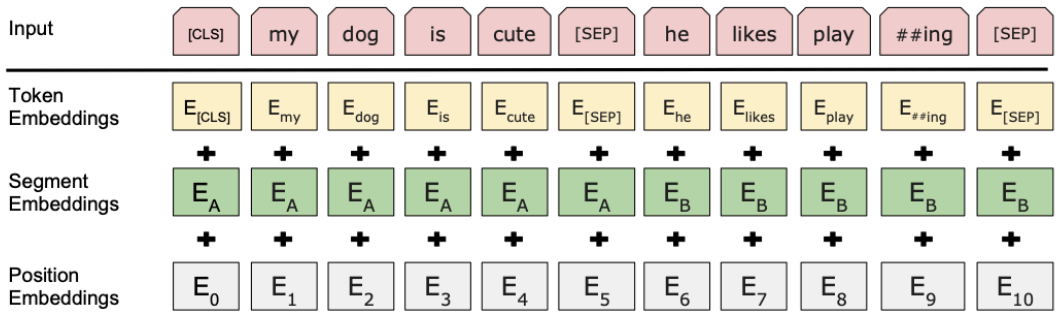

Fig. 7 BERT input representation as the sum of the token, segment and position embeddings (source *[21]*).

As is mentioned above, to accomplish the task, BERT has two stages: pre-training and fine-tuning.

Pre-training

The first step is the pre-training. The language model is trained on unlabeled data over two different tasks:

Task 1: Masked Language Modeling (Masked LM). To train a deep bidirectional representation, a given percentage of the input tokens are masked, and then some of the masked tokens are estimated.

Task 2: Next sentence prediction (NSP). In order to perform tasks such as Question Answering, in which the understanding of the relationships between two sentences is needed, the pre-train needs next sentence prediction. In this case, the model gets pairs of sentences as input and it learns to predict if the second sentence is the subsequent pair in the original data.

Fine-tuning

Fine-tuning is the second step and it is a directed step since the attention mechanism permits the realization of many different downstream tasks. It involves the change in model architecture but minimal, it only affects the last layers of the model. These layers are modified according to the task that it is going to perform, depending whether the task is a classification, question answering, or any other.

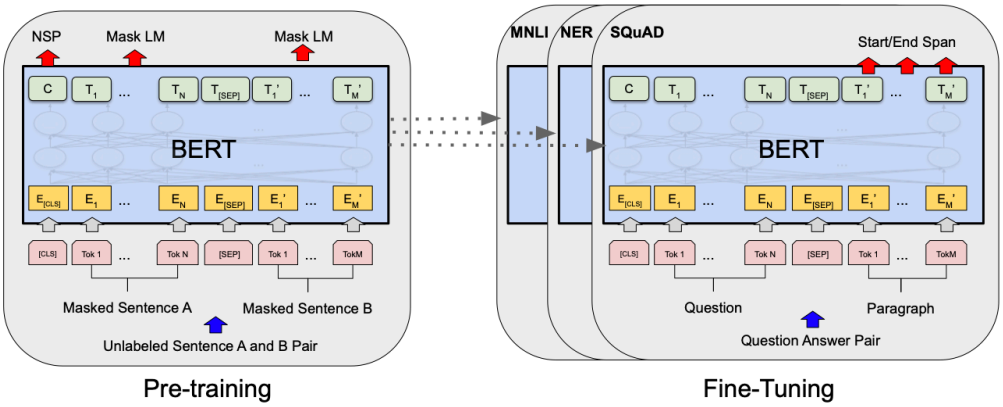

Fig. 8 Summary of pre-training and fine-tuning procedures for BERT. In the pre-training steps, two tasks are performed, masked LM and NSP. In the fine-tuning, depending on the final task, the final layer changes according to question answering (SQuAD), classification or name entity recognition (NER) (source [25]).

Evaluation methods

To analyze and asses the prediction of a model, there are some common metrics that can be used to describe its performance.

- **Confusion Matrix** is a table based on the test data. It contains the true positives and true negatives, where the prediction was negative or positive but correct. And it also contains the false positives and false negatives, where the prediction was incorrect (see *TABLE 1*).

TABLE 1 CONSTRUCTION OF CONFUSION MATRIX FROM CORRECT AND INCORRECT PREDICTIONS OF MODEL.

True	**Negative**	TN	FP
	Positive	FN	TP
		Negative	**Positive**
		Predicted	

- **Accuracy** is a calculation of how much a model properly classifies a data point. It is calculated as the number of predictions correct divided by the total number of prediction or from the confusion matrix as in formula (6).

$$AUC = \frac{TP+TN}{TP+TN+FN+FP} \quad (6)$$

- **Recall,** also known as sensitivity, tests the proportion of positive data points which were correctly estimated in comparison to the positive data points in the dataset.

$$Recall = \frac{TP}{TP+FN} \quad (7)$$

- **Precision** is calculated as the number of correct positive predictions divided by the total number of positive predictions.

$$PREC = \frac{TP}{TP+FP} \quad (8)$$

- **Loss** is measured on training and validation datasets and refers to how good the model is predicting on such two datasets. Loss is not a percentage like accuracy value. It is the sum of the errors made in each example from the prediction of training or validation dataset [25]. From these definitions, the loss is expected to decrease with the iterations of the models.
- **F1 score** is used to compare the performance out of different models. It is calculated as a harmonic average between precision and recall. The higher is the F1 score, the better is the model.

$$F1\ score = 2 * \frac{1}{\frac{1}{PREC}+\frac{1}{Recall}} \quad (9)$$

2.4. Association rules

Association rules are used in data mining to analyze the contents of a database and extract rules, the so-called association rules. The number of rules that from very simple database may be produced is actually very high. A tool for determining which rules to reject and which to maintain when implementing association rules is needed [26].

When applying this kind of algorithms to a data set, transactions and items are required. Items can be defined as the different classes or objects that appear in the data set and transactions as the way items are being grouped. A single transaction can contain one or more items.

Different algorithms for frequent pattern detection have been used in the biology field. For example, frequent pattern identification on sequence of proteins that are associated with a neurodegenerative disease has been develop [27].

An A priori algorithm has been used to find dominating amino acids patterns [28] and to identify new viral-host protein-protein interactions in HIV-human proteins [29].

A priori algorithm

This algorithm is included in the frequent item set mining algorithms. It uses the downward closure property in order to speed up the search for frequent item sets. This principle states that all the subsets included in a frequent item set must be also frequent to count them.

For the evaluation of the rules generated by the Apriori algorithm, the next three parameters are used:

- **Support** corresponds to the proportion of the database to which the rule successfully applies, i.e. the proportion of counts in which the item appears at the data set. Being R a possible item in the data set, the support of R is calculated as:

$$Support\ (\mathrm{R}) = \frac{\mathrm{count(R)}}{number\ of\ transactions} \quad (1)$$

- **Confidence** is defined as the proportion of transaction or records for which the rule is satisfied. Being the rule $L \rightarrow R$, meaning that when *item L* happen, it can be predicted that *item R* also happen, the confidence can be calculated as

$$\mathrm{Confidence(L \rightarrow R)} = \frac{\mathrm{count(L \cup R)}}{\mathrm{count(L)}} \quad (2)$$

or

$$\mathrm{Confidence(L \rightarrow R)} = \frac{\mathrm{Support(L \cup R)}}{\mathrm{Support(L)}} \quad (3)$$

- **Lift** refers to how likely item *R* happens when item *L* happens, while controlling how frequent item *R* is.

$$\text{Lift } (L \rightarrow R) = \frac{\text{Support}(L \cup R)}{\text{Support}(L) \times \text{Support}(R)} \quad (4)$$

A lift value equal to 1 means that no association between items exist.

If lift value is greater than 1 means that R is likely to happen if item L happens.

The A priori algorithm first generates itemsets of length *k* and counts their supports, and it eliminates candidates whose support is less than the minimum support established. With the resulting frequent k-item sets, the (k+1) itemsets are generated by combination of all the k-itemset. The supports are again calculated and the not frequent itemsets are discarded. This is how the downward closure property is used to restrict the number of candidates of (k+1) length. In this way, all the subsets that can be generated by a bigger set are all frequent.

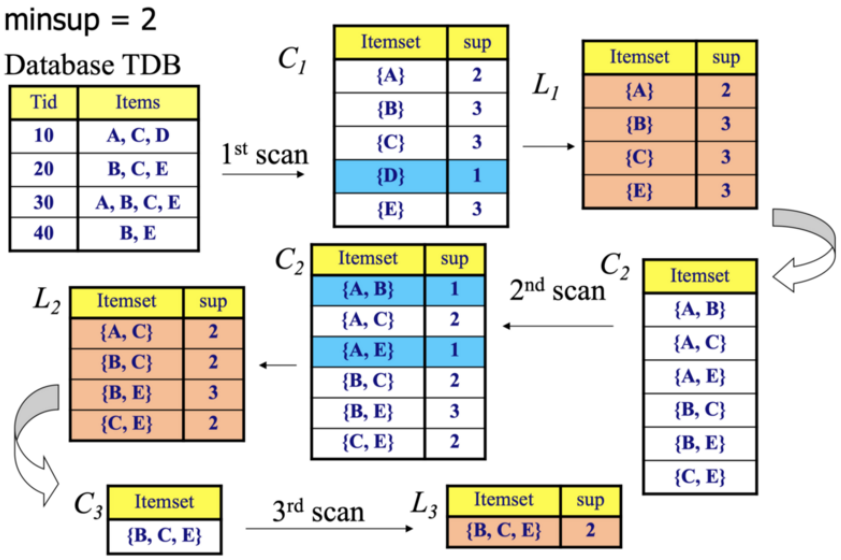

Fig. 9 Workflow of A priori algorithm from the selection of frequent items of length k to frequent itemsets of length k+1 based on the value of the support and the threshold selected (source *[30]*).

With this workflow seen in Fig. 9, the frequent itemsets have been generated, and now the association rules are going to be generated by using the confidence parameter. From the final frequent itemset generated, the candidates to be association rules are created as it can be seen in *TABLE 2*. These candidates must have a confidence greater than the minimum confidence established.

TABLE 2 CANDIDATES TO BE ASSOCIATION RULE FROM THE FREQUENT ITEMSET FOUND.

Association rules
BE-->C
BC-->E
CE-->B
B-->CE
C-->BE
E-->BC

Finally, the lift value is calculated as a measure of the importance of a rule. The greater the lift, the better the association rule found.

3. OBJECTIVES

This thesis has been developed at the Biocomputing Unit of the Natl. Center of Biotechnology (CSIC). The main purpose behind it is the use of data mining techniques to be applied in the biological databases to generate new available tools in this field.

The data will be extracted mostly from 3DBionotes, a web maintained at the Natl. Center of Biotechnology (CSIC) that integrates all the annotations from specific databases.

The first technique we have explored in our work is Association rules mining. This is done with two approaches, the simplest one which is finding relationships manually and the second approach is using A priori Algorithm. This algorithm uses various metrics to measure the strength of the association rules that it finds.

The second technique that we have used is Prediction. The objective is to develop an algorithm that is able to predict if a sequence inserted, is an epitope or not. The algorithm is using BERT for binary classification. The data used to run it is downloaded from IEDB, Immune Epitope Database and Analysis Resource.

4. MATERIALS

4.1 Electron Microscopy Data Bank

EMDB is a public repository for electron microscopy density maps of macromolecular complexes and subcellular structures. From EMDB, a list of the identification name of all the structures is downloaded via FTP[3].

4.2 3DBionotes

3DBionotes is a web framework used for the acquisition of part of the data that is used during the project. It integrates biological annotations and protein structure from several databases such as UniProt, Protein Data Bank, PhosphoSitePlus, IEDB, Biomuta and dSysMap [31]. UniProt database contains annotations for over 120 million proteins of all branches of life. It contains structural annotations, information about the location and the topology of the protein in the cell, post-translational modifications, variants and mutations [32].

Protein Data Bank (PDB) contains the 3D structures of proteins and complex assemblies.

PhosphoSitePlus is a database where the annotations of post-translational modifications like phosphorylation, acetylation are found.

Biomuta contains single-nucleotide variation associated with disease [33] and dSysMap contains Human disease-related mutations [34].

The annotations from all these databases are integrated at the 3DBionotes web server available at https://3dbionotes.cnb.csic.es.

4.3 A priori

An implementation of A priori algorithm was downloaded from https://pypi.org/project/apyori/. It is a module available for Python language and downloaded through pip[4].

3 FTP (File Transfer Protocol) link to download the identification name of the structures is ftp://ftp.ebi.ac.uk/pub/databases/emdb

[4] Pip is a standard manager for the packages available at Python. The A priori package is installed using pip install apyori.

4.4 IEDB database

The Immune Epitope database catalogs experimental data on antibody and epitopes. It is used to download the sequences of epitopes. The full list of epitopes was downloaded but then, only the linear epitopes were processed. A dataset with 908,655 epitopes from human and not human organism was formed.

4.5 BERT

BERT code was cloned from https://github.com/google-research/bert. It contains the files for TensorFlow code to do both pre-training and fine-tuning for a general task. In our case, to generate the protein language model.

4.6 Speed Benchmarking

For the first part of the thesis, in which the manual associations and the A priori algorithm were implemented, a 15 “MacBook Pro version 2018 was used.
To improve the performance of the training, a remote server running on Linux 4.15.0 with Ubuntu 18.04.1 LTS was used. Training was performance through CUDA from Nvidia.

TABLE 3 TECHNICAL SPECIFICATIONS OF THE REMOTE SERVER USED TO TRAIN AND TEST THE MODEL.

CPU:	Intel(R) Xeon(R) CPU E5-2630 v4 @ 2.20GHz - 40 cores
GPU:	4 x NVIDIA GeForce GTX 1070
RAM:	252 GB
SSD:	894.3 GB

5. METHODS

5.1 Association rules

As mentioned above, association rules forms part of data mining technique whose purpose is the generation of new knowledge based on frequent patterns found. For this aim, during the thesis, two methods have been applied: manual association and A priori algorithm.

In every data mining project, after defining and identifying the data needed, the data should be cleaned and given the format necessary for the main purpose. Then, the algorithms are selected and built. Then, the training and testing of the performance of the algorithm are done and finally, the model is verified.

For both methods in association rules, the same data is used, so the step of data extraction is common.

Data extraction

From EMDB, a list of all the identification names were downloaded. An EMDB identification contains information and structure of macromolecular complexes. This identification fits with one or several PDB models which is consulted through one of the API. Afterwards, the PDB models fit with the Uniprot sequences which are the ones that contains more annotations.

Among the APIs available in 3DBionotes, each database can be queried independently, and the annotations are extracted in JSON format. To get the data required, Python scripts were written to consult each database for each Uniprot sequence. All the data extracted was saved in different MySQL tables depending on the database.

5.1.1 Manual association

With the manual association approach, pairs of frequent patterns are intended to be found. Once the data is extracted, the algorithm is constructed.

Algorithm construction

In the case of manual association, this algorithm is very simplistic. In general, the first step is the establishment of two items that want to be related. Once it is defined, the data that is necessary is retrieved from the SQL tables through an advanced consult.

The following steps are the mathematical ones of the algorithm. It involves the comparison of start and end positions of the annotations of both item in the relation and counting the number of amino acids that coincide in both and which not.

Finally, as evaluation methods the risk ratio is calculated and both conditional probabilities.

Cases of study

Five different cases were studied.

- Mutations in nucleotide binding.
- Mutations in epitopes.
- Mutations relationship with protein secondary structure.
- Mutations in regulatory sites.
- Epitope relationship with protein secondary structure.

In all the cases, the calculation of the number of amino acids that take part of each of the items was restricted to the proteins that have annotations of at least the exposed group. For the first case, where the exposed group is the nucleotide binding, the proteins used for the calculations have at least nucleotide binding annotations. Also, before calculations and during the annotation extraction, a program was written to avoid overlaps while counting amino acids, as there are annotations whose start and end overlap in some intervals.

To evaluate this association, the mutations were extracted from the BioMuta and dSysMap databases, which contains mutations associated with diseases. The nucleotide biding annotations were extracted from Uniprot database. The epitopes annotations from IEDB were used to check this rule. The annotations related with the secondary structure are from Uniprot database. PhosphositePlus was used to extract the regulatory sites annotations.

In the cases where the exposed group is one of the secondary structure types, inside the protein sequence, the number of amino acids that are counted are the ones that have annotations on an alpha helix, beta strand or turn.

Evaluation methods

- **Risk Ratio (RR)** of an event is the probability of its occurrence following exposure to a risk variable compared to the probability of its occurrence within a control group.

TABLE 4 CONTINGENCY TABLE USED FOR THE CALCULATION OF RISK RATIO

Exposure Status	**Events Occurred**	
	Yes	No
Exposed	a	b
Not exposed	c	d

$$\text{Risk Ratio} = \frac{\frac{a}{a+b}}{\frac{c}{c+d}} \quad (5)$$

As a measure of effect size, a risk ratio is clinically significant when it is less than 1/3, meaning that the risk is lower in the exposed group; or more than 3.00 which means that the risk is increased in the exposed sample. [35]

5.1.2 A priori Algorithm

To improve the performance of manual association and eliminate the necessity of establishing the relation that we want to calculate, the A priori algorithm was used. Once the data was extracted in MySQL tables, data should have the format needed for the execution of the algorithm.

In A priori algorithm, data should be put in transactions, that means that in the same line, items that happen at the same time should be put, i.e. at position 23 of the sequence, there exist a structural annotation and a mutation. In the algorithm, this relation should be introduced as a line as *{Mutation, Helix}*.

To construct the list of transactions needed, an iterative process is done. A list of transactions of a single Uniprot is constructed, then it is added to the final list of all transactions.

First, it is constructed a table where the number of rows is the length of the protein and the number of columns is variable depending on the number of items that contains each position in the protein sequence. The different annotations for each protein are consulted through an advanced search on MySQL, we search for the annotations in every table of the same Uniprot.

Once, MySQL retrieved the annotations, an iterative process started to fill the list of transactions of each Uniprot. When an annotation is read, the program takes the start and end positions, and it fills the columns of the table with the description of the annotation from start position to end position rows.

When all the annotations for an Uniprot are read and the table for such is filled, that table is added to the final one and the process starts again for the next Uniprot. The number of resulting transactions is 2,968,009. Finally, the list that is obtained is introduced in A priori.

The A priori implementation of Python needs four parameters to be selected: minimum support, minimum confidence, minimum lift and minimum length. Minimum length refers to the number of items that forms part of the transaction.

The selection of the appropriate parameters was done by trial and error. It started by setting a value for the parameters and we changed it depending on the number of transactions that the algorithm.

5.2 Epitope predictor

Data preprocessing

The data set used for training the algorithm was constructed with two data subsets. One of the subsets were the negative samples, which was constructed from sequences of proteins downloaded from Uniprot. The other subset was the positive samples, which were the epitope sequences, downloaded from IEDB as described in materials section.
One of the principal objectives when generating the data set is the construction of data set with the same length distribution as in *Fig. 10*.

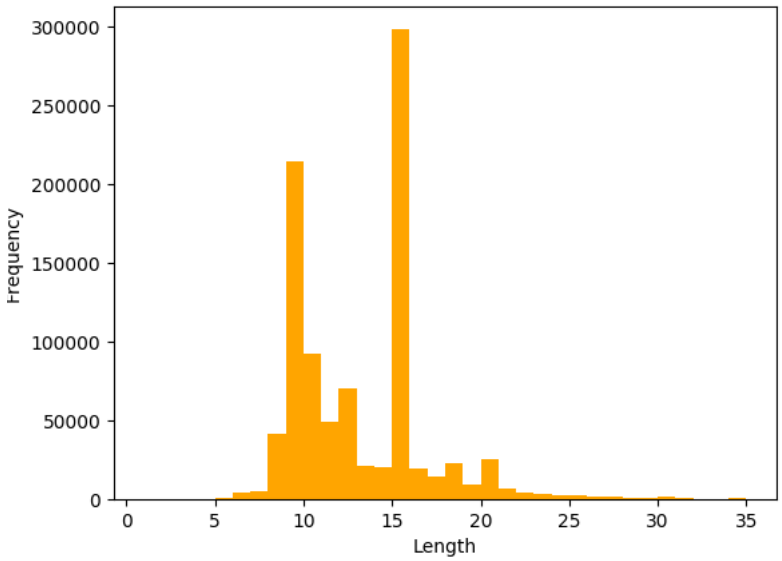

Fig. 10 Distribution of epitopes length

To generate the negative samples, the complete protein sequences were downloaded from Uniprot, and they were cut according to the same distribution of epitopes and length. The sequences of negatives and positives samples were separated in amino acids as different words in a sentence. Within the line of a sequence, there is also a label, written as a 0 or 1, meaning non epitope or epitope sequence.

Fig. 11 Transformation of protein sequences into spaced amino acids samples.

Finally, a data set is formed with 1,817,310 sequences that were later divided into three different groups for doing the training, the validation and the test of the algorithm. The data set split is done according to a 70%, 20% and 10% respectively.

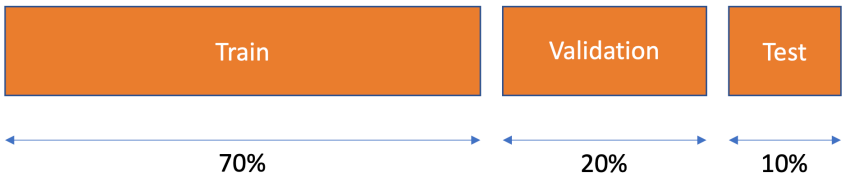

Fig. 12 Division of the data set in the training, validation and test sets according to an 70%, 20% and 10% respectively.

BERT architecture

As was described in materials, the code for the development of the epitope predictor was downloaded from BERT GitHub[5]. It contains a set of scripts that are intended for the different tasks that can be performed using BERT. For the specific purpose of developing an epitope predictor, two main steps were performed: pre-training and binary classifier.

Before starting the pre-training, the necessary files were created.

- Samples.txt: Text file used to do the pre-train. In each line of the file, a different sequence is written without labels.
- Vocab.txt: Vocab file created to map WordPiece to word id. To create this file, BertWordPieceTokenizer is applied to the samples.txt to generate a dictionary of the different tokens in which the samples are divided. In this case, the dictionary that results contain the different letters that represent the amino acids and the special tokens used.
- Bert_config.json: Configuration file that specifies the hyperparameters of the model. For the development of the epitope predictor, our configuration of BERT is shown in *TABLE 5*.

TABLE 5 BERT BASED CONFIGURATION PARAMETERS

Attention_probs_dropout_prob	0.1
Hidden_act	GELU: Gaussian Error Linear Unit
Hidden_dropout_prob	0.1
Hidden size	768
Initializer_Range	0.02
Intermediate_size	3072
Max_position_embeddings	512
Num_attention_heads	12
Num_hidden_layers	12
Type_vocab_size	2

[5] https://github.com/google-research/bert

Vocab_size	30

Pre-training

After the tokenization, the script create_pretraining.py was run on Python using the parameters shown in .

TABLE 6. The parameters selected are all the default ones of the algorithm except for the maximum sequence length and the maximum predictions per sequence. This last parameter is calculated by multiplying the *Masked_lm_prob by Max_seq_length* [36].

TABLE 6 PARAMETERS USED ON CREATE_PRETRAINING.PY

Max_seq_length	27
Max_predictions_per_seq	4
Masked_lm_prob	0.15
Random_seed	12345
Dupe_factor	5

The next step was run_pretraining.py, which is the main part of the pretraining. The *batch size* is the number of training examples used in each iteration. In order to improve the performance and reduce hours of training, the *batch size* should be as high as possible but taking into account the out-of-memory errors. For both processes, a batch size of 32 was chosen as the maximum that the GPUs supported. The pretraining takes the longest time and consumes the majority of resources of the GPUs.

Fine-tuning

For the development of an epitope predictor, a binary classification is the task for which the pretrained model has to be fine-tuned. That task is called CoLA, “The Corpus of Linguistic Acceptability”, used for single sentence classification into two classes. The architecture of BERT is the same as in the pre-training but with a single linear layer on top for classification.

To accomplish the task, run_classifier.py is run for training and evaluation. When the fine-tuning is finished, the prediction is done with a dataset that only contains the sequences and not the labels.

6. RESULTS AND DISCUSSION

The following chapter explains the results of the data mining techniques developed during this work: association rules and epitope predictor based on BERT architecture.

6.1 Association rules

Manual association

Six different associations were studied in order to check and understand if a relation between both could be established. To do that, the risk ratio was calculated for the six cases.

The first case was the study of the incidence of mutations in nucleotide binding (NP_binding). The number of amino acids (aa) for each of the cases is shown in *TABLE 7*.

TABLE 7 CONTIGENCY TABLE OF MUTATION'S INCIDENCE IN NUCLEOTIDE BINDING

	Mutation	NOT mutation
NP_binding	232 aa	14287 aa
Not NP_binding	3536 aa	727337 aa

The result of the calculation of risk ratio from the contingency table was 3.302. This means that the risk of having a mutation in an amino acid that forms part of a nucleotide binding site is three times higher than in the rest of the protein sequence. From the biological point of view, this is a correct association because the binding sites of a protein are key part of most of the functional process in which the protein can be involved. Thus, these mutations can inactivate promoters or regulatory sequences, resulting in a loss of function and with the potential of generating a disease.

The second case was the analysis of the frequency of mutations in epitopes sequences. The results of the contingency table for this case is shown in TABLE 8.

TABLE 8 CONTIGENCY TABLE OF MUTATION'S INCIDENCE IN EPITOPES.

	Mutation	Not Mutation
Epitope	213 aa	50947 aa

Not epitope	1699 aa	372698 aa

The risk ratio is 0.916 which indicates a decrease in the risk of a mutation in the amino acid that forms part of epitopes. This means that out of the proteins that have epitopes annotated in the sequence, there are more mutations annotated in the amino acids that are not epitopes. However, the difference with respect to a risk ratio of 1 is so small, that this result cannot be considered significant (risk ratios are considered to be significant if they fall below 1/3 or above 3).

The third case was the study of occurrence of mutations in the secondary structure annotations, whether a mutation occur more frequently out of alpha helix, beta strand or turn. For these case, three different risk ratios are calculated, and the results are shown in *TABLE 9*.

TABLE 9 CONTINGENCY TABLE OF INCIDENCE OF MUTATIONS IN ALPHA HELIX, BETA STRAND AND TURN.

	Mutation	Not Mutation
Alpha Helix	3945 aa	585386 aa
Turn	440 aa	57839 aa
Beta strand	3282 aa	400979 aa

The risk ratio values are 1.207 for Beta strand, 0.838 for alpha helix and 1.045 for turn. From these three values, it can be said that the risk of having a mutation is almost equally likely to happen in any of the secondary structure types.

Mutations associated with diseases can be caused by a change in secondary structure. In that case, the risk of being produced by a mutation in a beta strand is greater because of the structure associated with it. It contains less contacts between inter-residues than alpha helices, therefore the mutation can be more disruptive in beta strand causing a bigger change in protein folding and with that, in protein function. However, we did not verify this hypothesis.

The fourth case is the analysis of the incidence of mutations in regulatory sites. In *TABLE 10* it is shown the number of amino acids for each of the cases. The value of the risk ratio calculated is 1.931.

This result shows an important increase in the risk of regulatory sites to suffer a mutation associated with a disease. This is because of the key function that regulatory sites have in most of the biological processes. However, the statistical significance of this finding is yet to be confirmed.

TABLE 10 CONTINGENCY TABLE OF INCIDENCE OF MUTATIONS IN REGULATORY SITES AS EXPOSED GROUP.

	Mutation	Not Mutation
Regulatory site	61 aa	5168 aa
Not regulatory site	6361 aa	1059058 aa

The last case was the study of the incidence of epitopes out of the secondary structure types. In this case, the exposed group is the secondary structure. The risk ratio values are 0.0 for Turn, 0.145 for beta strand and 8.689 for the alpha helix, which are calculated from *TABLE 11*.

TABLE 11 CONTIGENCY TABLE OF THE INCIDENCE OF EPITOPES IN SECONDARY STRUCTURE TYPES.

	Epitope	Not Epitope
Turn	0 aa	58279 aa
Beta strand	243 aa	404018 aa
Alpha helix	2689 aa	586642 aa

The high risk of incidence of epitopes in alpha helix shows that epitopes are more concentrated in these regions of the secondary structure of the protein.

A priori algorithm

As it was described in the Methodology section, for the correct construction of the association rules in A priori algorithm four parameters were selected based on the quantity of transactions. For this case, as the number of transactions is of almost three million, the

initial value for the support was set to 0.001. This means that the minimum number of times that a rule must appear is 3,000 times.

In *TABLE 12* we report the parameters introduced in A priori algorithm for the generation of the association rules. In the last column, it appears the final number of rules that the algorithm constructed. The number of rules doubles as the minimum support is decreased.

TABLE 12 PARAMETERS USED FOR THE CONSTRUCTION OF ASSOCIATION RULES DERIVED FROM A PRIORI.

Case	Min_support	Min_confidence	Min_lift	Number of association rules found
(1)	0.001	0.700	3.000	11
(2)	0.0005	0.700	3.000	26

All the resulting rules for cases 1 and 2 are found in **12.1 Annex A** . After the rules are generated, they were revised manually to check if they were corrected or not from a biological point of view.

A huge part of the rules that are generated are just a relationship between a protein and its localization in the cell. This is because if a chain of protein is annotated as extracellular for example in its hole chain, the number of transactions that the A priori algorithm receives is the length of the protein. This increases the support for such rule.

It is found that alpha-1 is likely to be in an extracellular region. This was revised at Uniprot [37] and it is a correct rule with 96% of confidence.

Another rule found is the N-linked (GlcNAc...) asparagine at extracellular which is a partially wrong rule. Because the N-linked asparagine is just a post-translational modification, where an oligosaccharide binds to an asparagine residue of the protein. This process starts at the endoplasmic reticulum of the cell but maybe or not the protein goes to the extracellular space [38].

It was also found that NR LBD, which stands for nuclear receptor ligand-binding domain, is likely to happen with Helix. This is a totally correct rule, because the structure of these transcriptional receptors is composed of 11 to 13 alpha helices [39].

N-terminal globular head is likely to happen in Helix was another rule found by A priori. The structure of this domain is exclusively composed of helices and loops [40], so the rule was correct.

When the min_support is decreased, more rules were generated which must also be revised.

It was found that GS domain is likely to be cytoplasmic, which is a true rule. The GS domain which forms part of a receptor, is located in the cytoplasmic portion of the receptor [41].

Although the rules generated do not add new knowledge about proteins, this application has a huge potential for the new prediction of biological annotations. There exist a lot of sequences of protein without annotations, without structure. This kind of rules can help when there exists similar protein to annotate, knowing that there are domains that have a characteristic secondary structure for example.

6.2 Epitope predictor

Pre-training

The first part of the construction of the epitope predictor was the pretraining. It was trained on two GPUs for almost two months to get to the desired number of steps.

TABLE 13 RESULTS OF ACCURACY AND LOSS OF BERT PRETRAINING.

Global_step[6]	Loss	Masked_lm_accuracy	Masked_lm_loss
165,000	2.598	0.209	2.598
1,800,000	2.241	0.327	2.239
3,000,000	1.885	0.438	1.884
5,000,000	1.574	0.529	1.574

In the evaluation perform of BERT, two parameters more are at the output, which are the accuracy and loss for the next sentence prediction task. For our case, as that task is not used, the next sentence loss does not contribute to the total loss and the next sentence accuracy is 1.0 as there is not a second sentence. As can be seen in *TABLE 13* and in *Fig. 13* graphically, the accuracy increases over the steps and the loss decreases. This is the expected result in a normal machine learning model.

[6] Global_step is the number of groups of examples of dimensions as the batch size that has passed through the model.

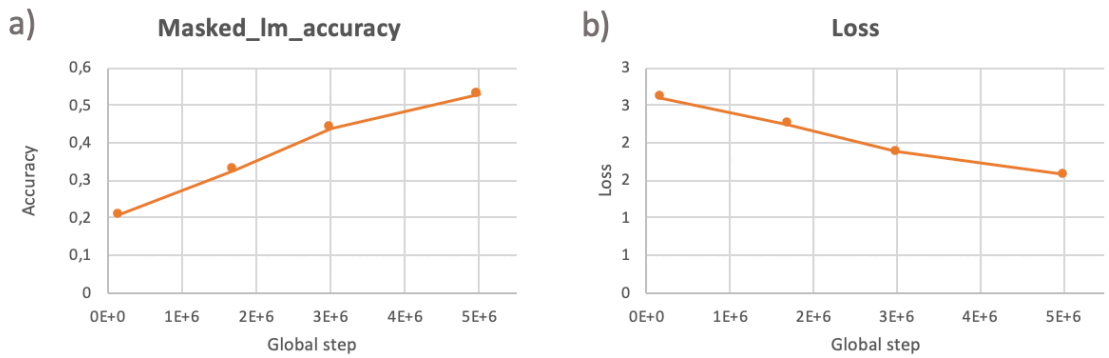

Fig. 13 Evolution of accuracy and loss during the pretraining steps.

To get a better result, and as it was a time-consuming process, steps of pretraining and fine-tuning were intercalated in order to improve accuracy values and not to incur in excessive pretraining time. The results of fine-tuning are shown in next section.

Fine-tuning

When the global step of the pretraining was 1,800,000, the fine-tuning process started, and the results are shown in table below. The number of epochs of fine-tuning refers to the number of complete times that the training dataset passes through the model.

TABLE 14 RESULTS OF FINE-TUNING AFTER 1,800,000 STEPS OF PRETRAINING

	Eval_accuracy	Eval_loss
1 epoch	0.8174	0.404
5 epochs	0.835	0.447
10 epochs	0.834	0.656
20 epochs	0.838	1.076

Then, the pretraining got to 3,000,000 steps and the result of the fine-tuning are better than before for 20 epochs.

TABLE 15 RESULTS OF FINE-TUNING AFTER 3,000,000 STEPS OF PRETRAINING

	Eval_accuracy	Eval_loss
1 epoch	0.831	0.381
5 epochs	0.842	0.467
10 epochs	0.843	0.788
20 epochs	0.846	1.152

The pretraining continued until 5,000,000 where the accuracy increased with respect of 3,000,000 steps, and in fine-tuning, the accuracy increased slightly.

TABLE 16 RESULTS OF FINE-TUNING AFTER 5,000,000 STEPS OF PRETRAINING.

	Eval_accuracy	Eval_loss
1 epoch	0.829	0.390
5 epochs	0.843	0.541
10 epochs	0.844	0.811
20 epochs	0.850	1.138

As it can be observed in *TABLE 14*, in *TABLE 15* and in *TABLE 16*, the accuracy of the models have increased and the evaluation loss increases with the epochs. This is probably due to an overfitting of the models. This can happen because the model does not generalize well on data that has not been seen before, which means that the model has learned patterns that are specific to the training dataset and when doing the validation, these patterns are irrelevant.

As the accuracies values for the three models are almost the same, the final test was used to compare their performance.

The result of the prediction was a text plain file, with two columns corresponding to the two labels of the classification, epitope or non-epitope and the length of the rows equal to the number of test examples introduced. In a single row it is found the probability of each class for each example.

The first model that was tested was the resulting model after 20 epochs of fine-tuning and 1,800,000 steps of pretraining.

The methods to do the evaluation described before are shown in *TABLE 17* and in the confusion matrix in *Fig. 14*. This table was calculated with validation data that was never seen by the algorithm.

TABLE 17 EVALUATION PARAMETERS CORRESPONDING TO MODEL 1

Accuracy	0.838
Recall	0.813
Precision	0.856
F1-score	0.833

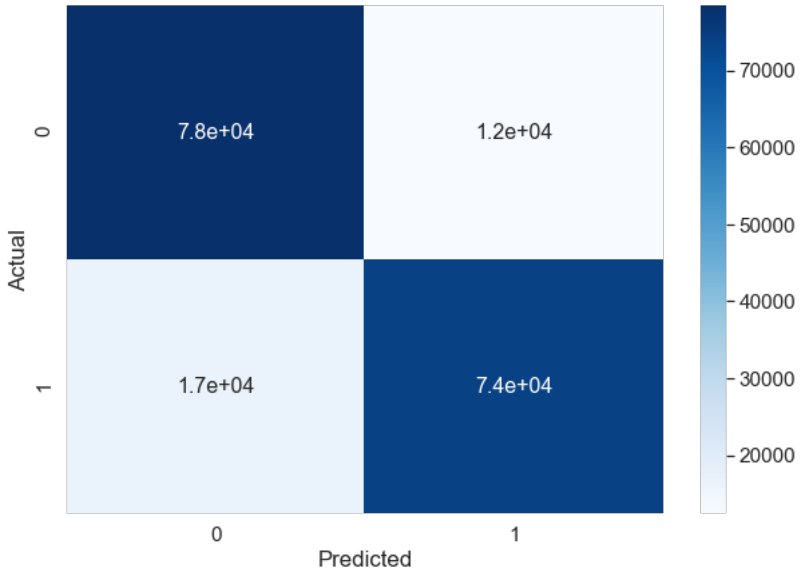

Fig. 14 Confusion matrix for BERT model after 1,800,000 steps of pretraining and 20 epochs of fine-tuning. The 0 means a negative prediction or prediction of a non-epitope sequence and the 1 means the prediction of an epitope sequence. In the upper left corner, the True Negatives; in the upper right corner, the False Positives. In the lower left corner, the False Negatives and in the lower right corner, the True Positives. The colormap indicates the number of predictions made in each case.

The second model that was tested was the resulting model after 20 epochs of fine-tuning and 3,000,000 steps of pretraining. The results of the evaluation are shown in *TABLE 18*

TABLE 18 EVALUATION PARAMETERS CORRESPONDING TO MODEL 2

Accuracy	0.845
Recall	0.810
Precision	0.870
F1-score	0.838

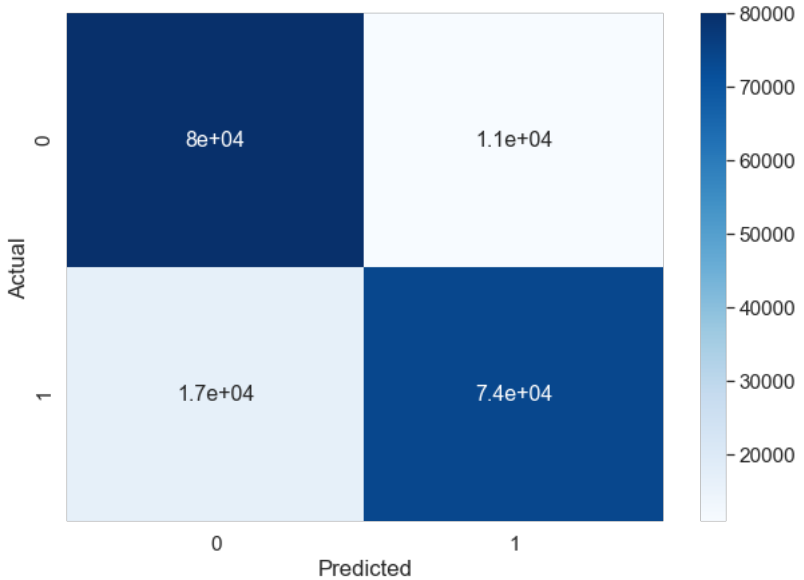

Fig. 15 Confusion matrix for BERT model after 3,000,000 of pretraining and 20 epochs of fine-tuning.

The third and last model used to do the evaluation is the model derived after 5,000,000 steps of pretraining and 20 epochs of fine-tuning.

TABLE 19 EVALUATION PARAMETERS CORRESPONDING TO MODEL 3

Accuracy	0.848
Recall	0.800
Precision	0.885
F1-score	0.840

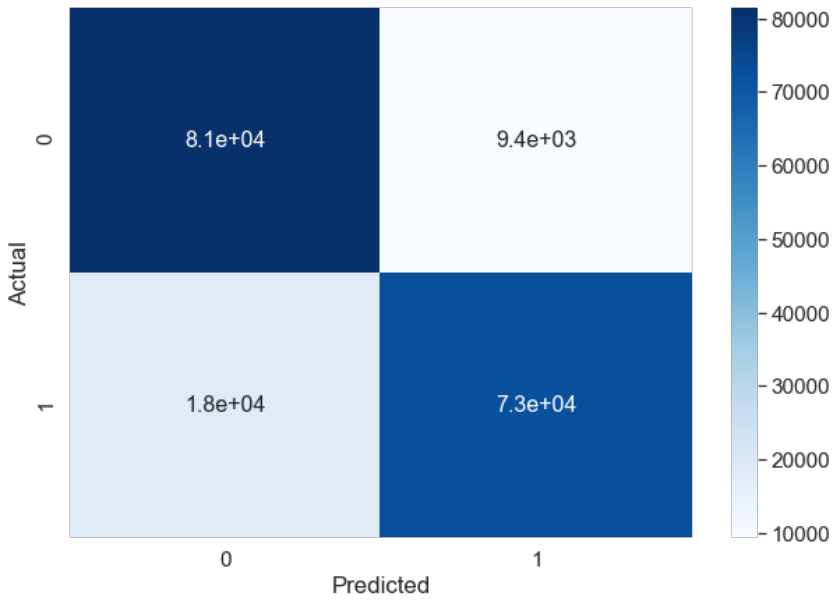

Fig. 16 Confusion matrix for BERT model after 5,000,000 of pretraining and 20 epochs of fine-tuning.

From both three models, it can be seen at their confusion matrix that the models predicted better the non-epitope sequences than the epitope ones and there has been an increase in the number of True negatives from model 1 to model 3.

At model 3, there has been a decrease in the number of epitopes predicted well in favor of the prediction of epitope sequences as non-epitope sequences.

When comparing the parameters, it can be seen that the model 3 has the better performance based on the F1-score, because it has the highest value and the highest accuracy also.

This binary classification has good results for the accuracy, it would be improved with a bigger training set in order to avoid an overfitting of the models. With a higher number of steps and this increase in data, this predictor could have better results and it could be compared with some of the predictors that already exist. However, this bigger training set

is not possible at the moment as we have used the whole IEDB. On the other side, we can make the BERT model smaller in order to prevent the overfitting. This means to decrease the complexity of the architecture of BERT.

7. CONCLUSION AND FUTURE WORK

Nowadays data mining and computational methods applied to the biology and healthcare industry represents an emerging area of research with a huge potential. This big prospect comes from all the data available generated in part from large scale experiments.

The application of these data mining techniques for this work had the purpose of discovering new patterns or being able to detect them to create newly available tools. The development of these new biomedical applications can help to reduce the cost of research, reduce time, and even save lives. The use of bioinformatics as initial steps in research reduces the different approaches, for example reducing the number of proteins to be studied in a selection process. It speeds up the process and reduces the cost at the same time because of the decrease in the number of experiments to be performed.

The use of manual associations is limited due to the necessity of having an expert biological knowledge for the evaluation of the rules generated. One of the interesting rules that was extracted is the high risk of having alpha helices in epitopes sequences. This association can be helpful in future work during the development of epitope predictor.

The construction of the epitope predictor represents the beginning of a tool for fighting against many infectious diseases and even cancer, which has not cure in 21st century. The epitope predictor can help in the selection of epitopes and in the design of these vaccines.

The use of NLP to understand the language of proteins has shown good results. This model based on BERT architecture can be improved to get not only higher accuracy but also to be able to do a multiclass classification according to the different types of epitopes that exist: β-cell epitopes, T-cell epitopes.

The association rules related with epitopes can be also used for improvement of the detection or the evaluation of the epitopes predicted. If new frequent patters are found, they can be incorporated as extra models of the predictor when taking into account the structure, the solvent accessibility. These biological parameters based on physicochemical properties of proteins can be introduced in a hybrid model, in which NLP processing and other machine learning techniques could be applied.

8. REGULATORY FRAMEWORK

The techniques used in this study are not regulated by any legislation or subject to the defense of any intellectual property, and do not compromise any legal code of ethics.

The methods implemented in this research are not governed under any law or subject to any intellectual property protection and do not break any ethical code of ethics. Nevertheless, several programs and programming language were used to implement the techniques linked to their regulations.

With regard to the programming language used to write the scripts, Python is an open source.

A priori Algorithm is licensed under an MIT license. This is the most permissive free software licenses; the only condition is the addition of a copy of the original MIT license in the distribution of new product.

For the development of the predictor model, BERT is licensed under an Apache License 2.0. This license allows to freely use, change or distribute the product.

9. SOCIO-ECONOMIC IMPACT

Biological databases have evolved exponentially over the last 20 years. It is estimated that this amount of data doubles every 10 months [42]. This growth was attributed primarily to the accumulation of data from molecular sequences [43].

The high value of the data stored in these databases has increased the demand of data mining tools. Data mining tools are helpful in extracting useful knowledge. It may either be used to test a particular hypothesis or to identify patterns automatically [42]. This discovery has many advantages in the understanding of human biology and all the processes caused by diseases which scientist not know yet.

The incorporation of bioinformatics approaches such as epitope predictors for the development of vaccines possesses many advantages over the traditional approaches. With the aid of epitope predictive software, the number of wet experiments to be performed while vaccine development is reduced, as a consequence the cost of the research and the time are reduced significantly [44].

It is estimated that the cost of producing a vaccine, from development and testing to drug registration, is between US $200 million and US $500 million per vaccine. And the overall time that it can take before launch the vaccine is 10 years [45].

It is important to highlight the fight against the SARS-CoV-2, which the whole world is facing. Scientific groups and pharmaceutical companies are looking to develop a treatment or vaccine to end this pandemic in record time. Vaccines takes years of study and experimentation before supplying the market, however by 2021, laboratories are promising to develop a safe and reliable vaccine.

There are 36 vaccines currently in clinical trials on humans [46]. One of the vaccines in a more advanced phase is Moderna vaccine which has received almost 1$ billion from the Biomedical Advanced Research and Development Authority of the U.S. government [47].

Bioinformatics also plays a key function in the ongoing COVID-19 epidemic by identifying effective drugs against viral targets and then, these are now being evaluated for confirmation in a laboratory. The rapid publication of the genome sequence and protein sequences at the NCBI database has had the potential to be used in these bioinformatics tools as in China where the used BepiPred server to identify the epitopes

candidates to develop a vaccine and further test within both in vitro and in vivo models [48].

By identifying a few possible ways from hundreds of alternatives, bioinformatics will severely reduce research costs. Bioinformatics may evaluate a vast volume of data to arrive at a hypothesis that can be further tested in a laboratory by experiments.

There is no doubt about the potential that the bioinformatic field has in the 21st century and the applications it has for improving the health of the people.

10. BUDGET

The estimation of the cost of the project is calculated in two different sections: the human resources associated, and the technical equipment needed.

TABLE 20 HUMAN RESOURCES COST

Element	**Cost (€/hour)**	**Time investment (hours)**	**Cost**
Student	20.00 *€*	400	**7600 *€***
Tutor	55.00 *€*	25	**1375 *€***
TOTAL			**8975 *€***

TABLE 21 MATERIALS COST

Element	***Description***	***Cost***	***Months***	***Amortization***
MacBook Pro	*15" version 2018, Intel Core I7 2.6 GHz, 16 GB RAM SSD, 512 Gb Disk SSD, Radeon Pro 560X 4Gb*	2349*€*	5	**978.75 *€***
T-Series SP Intel Xeon	*Intel(r) Xeon(r) CPU e5-2630 v4 @ 2.20ghz - 40 cores, Nvidia GeForce GTX 1070 379x4 (x4)*	*4500 €*	*3*	***1125 €***
TOTAL				***2103.75 €***

TABLE 22 SUMMARY

Element	***Cost***
Human resources	**8975** *€*
Materials	**2103.75** *€*
Indirect costs (materials and human resources) 15%	**1661.81** *€*
TOTAL	12,740.56 ***€***

11.BIBLIOGRAPHY

[1] D. Zou, L. Ma, J. Yu and Z. Zhang, "Biological Databases for Human Researcg," *Genomics Proteimics Bioinformatics,* pp. 55-63, Feb 2015.

[2] P. Sevon, L. Eronen, P. Hintsanen, K. Kulovesi and H. Toivonen, "Link Discovery in Graphs Derived from Biological Databases," *HIIT Basic Research Unit, Department of Computer Science,* 2006.

[3] A. Bayat, "Bayat A. Science, medicine, and the future: Bioinformatics.," *BMJ,* vol. 324(7344), pp. 1018-1022, 2002.

[4] P. Katara, "Role of bioinformatics and pharmacogenomics in drug discovery and development process," *Network Modeling Analysis in Health Informatics and Bioinformatics 2,* p. 225–230, 2013.

[5] "Chapter 3, Protein Structure and Function," in *Biochemistry 5th edition*, New York, 2002.

[6] I. Rehman, J. Jiang and S. Botelho, in *Biochemistry, Secondary Protein Structure*, StatPearls, 2020.

[7] J. Abbass, "Secondary structure-based template selection for fragment-assembly protein structure prediction.," 2018.

[8] L. Liu, Y. Fang and M. Li, "Prediction of Beta-Turn in Protein Using E-SSpred and Support Vector Machine," *Protein J 28,* pp. 175-181, 2009.

[9] G. J. Tortora and B. Derrickson, Principles of Anatomy and Physiology 12th Edition, John Wiley & Sons, Inc.

[10] C. A. Janeway, P. Travers, M. Walport and M. J. Shlomchik, Immunobiology: The Immune System in Health and Disease. 5th edition., New York: Garland Science, 2001.

[11] J. M. Cruse, R. E. Lewis and H. Wang., "7-Antigen Presentation," in *Immunology Guidebook*, Academic Press, 2004, pp. 267-276.

[12] X. Yang and X. Yu, "An introduction to epitope prediction methodsand software," *Reviews in Medical Virology,* pp. 77-96, 2008.

[13] A. Sette and J. Fikes, "Epitope-based vaccines: an update on epitope identification, vaccine design and delivery," *Current Opinion in Immnulogy,* vol. 15, pp. 461-470, 2003.

[14] R. M. Ribas Aparicio, J. A. Castelán Vega, A. Jiménez Alberto and G. P. Monterrubio López, "The Impact of Bioinformatics on Vaccine Design and Development," in *Vaccines*, 2007.

[15] U. Kulkarni-Kale, S. Bhosle and A. S. Kolaskar, "CEP: a conformational epitope prediction server," *Nucleic Acids Res.,* 2005.

[16] M. Jespersen, B. Peters, M. Nielsen and P. Marcatili, "BepiPred-2.0: improving sequence-based B-cell epitope prediction using conformational epitopes," *Nucleic Acids Research,* pp. 24-29, 2017.

[17] L. Backert and O. Kohlbacher, "Immunoinformatics and epitope prediction in the age of genomic medicine," *Genome Medicine 7,* vol. 119, 2015.

[18] R. E. Soria-Guerra, R. Nieto-Gomez, D. O. Govea-Alonso and S. Rosales-Mendozab, "An overview of bioinformatics tools for epitope prediction: Implications on vaccine development," *Journal of Biomedical Informatics,* vol. 53, pp. 405-414, 2015.

[19] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So and J. Kang, "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," *Bioinformatics,* vol. 36, pp. 1234-1260.

[20] J. Vig, A. Madani, L. R. Varshney, C. Xiong, R. Socher and N. F. Rajani, "BERTology Meets Biology: Interpreting Attention in Protein Language Models," 2020.

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

[22] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser and I. Polosukhin, "Attention Is All You Need," in *31st Conference on Neural Information Processing Systems*, Long Beach, CA, USA, 2017.

[23] J. Alammar, "The Illustrated Transformer," 2018. [Online]. Available: http://jalammar.github.io/illustrated-transformer/.

[24] T. Wood, "DeepAI-Softmax Function," [Online]. Available: https://deepai.org/machine-learning-glossary-and-terms/softmax-layer.

[25] H. Bommana, "Loss Functions Explained," 2019. [Online]. Available: https://deeplearningdemystified.com/article/fdl-3.

[26] M. Braner, Principles of Data Mining 3rd Edition, Springer, 2016.

[27] M. Shahedul Islam, S. Saha and M. S. Rahman, "Pattern Identification on Protein Sequences of Neurodegenerative Diseases Using Association Rule Mining," in *Seventh International Conference on Advances in Computing, Electronics and Communication*, Kuala Lumpur, 2018.

[28] S. Dhumale, "Predicting Patterns over Protein Sequences Using Apriori Algorithm," *International Journal of Engineering and Computer Science,* vol. 4, no. 7, 2015.

[29] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay and R. Eils, "Mining association rules from HIV-human protein interactions," *2010 International Conference on Systems in Medicine and Biology, Kharagpur,* pp. 344-348, 2010.

[30] "Association rule mining and Apriori algorithm - Develop Paper," 24 2 2020. [Online]. Available: https://developpaper.com/association-rule-mining-and-apriori-algorithm/.

[31] J. Segura, R. Sanchez-Garcia, C. O. S. Sorzano and J. M. Carazo, " 3DBIONOTES v3.0: crossing molecular and structural biology data with genomic variations," *Bioinformatics,* vol. 35, pp. 3512-3513, 2019.

[32] The Uniprot Consortium, "UniProt: a worldwide hub of protein knowledge," *Nucleic Acids Research,* vol. 47, pp. D506-D515, 2019.

[33] H. Dingerdissen, J. Torcivia Rodriguez, Y. Hu and R. Mazumder, "BioMuta and BioXpress: mutation and expression knowledgebases for cancer biomarker discovery.," *Nucleic Acids Research,* 2017.

[34] R. Mosca, J. Tenorio Laranga, R. Olivella and V. Alcalde, "dSysMap: exploring the edgetic role of disease mutations," *Nature Methods,* vol. 12, pp. 167-168, 2015.

[35] C. Andrade, "Understanding Relative Risk, Odds Ratio, and Related Terms: As Simple as It Can Get," *Clinical and Practical Psychopharmacology,* July 2015.

[36] [Online]. Available: https://github.com/google-research/bert.

[37] "Uniprot: Alpha-1-antitrypsin," 12 August 2020. [Online]. Available: https://www.uniprot.org/uniprot/P01009.

[38] H. Lodish, A. Berk and S. Zipursky, "Section 17.7, Protein Glycosylation in the ER and Golgi Complex.," in *Molecular Cell Biology. 4th edition*, 2000.

[39] Nuclear receptor (NR) ligand-binding domain (LBD), "PROSITE," 2017. [Online]. Available: https://prosite.expasy.org/PDOC51843#references.

[40] Y. An, C. Chen and B. Moyer, "Structural and Functional Analysis of the Globular Head Domain of p115 Provides Insight into Membrane Tethering," *journal of molecular biology,* Vols. 399,1, pp. 26-41, 2009.

[41] J. Massagu, J. Kuriyan, Y. Chen and M. Huse, "Crystal structure of the cytoplasmic domain of the type I TGF beta receptor in complex with FKBP12," vol. 96, pp. 425-436, 1999.

[42] D. A. Abraham, P. A.-E. Hassanien and P. A.́. P. d. L. F. d. Carvalho, "Volume 4: Bio-Inspired Data Mining Theoretical Foundations and Applications," in *Foundations of Computational Intelligence*, 2009.

[43] R. Luethy and C. Hoover, "Hardware and software systems for accelerating common bioinformatics sequence analysis and algorithms," *Drug Discovery Today BIOSILICO,* 2004.

[44] R. M. Ribas-Aparicio, J. A. Castelán-Vega, A. Jiménez-Alberto, G. P. Monterrubio-Lopez and G. Aparicio-Ozores, "The Impact of Bioinformatics on Vaccine Design and Development.," 2017.

[45] I. Serdobova and M.-P. Kieny, "Assembling a Global Vaccine Development Pipeline for Infectious Diseases in the Developing World," *Am J Public Health,* vol. 96, no. 9, pp. 1554-1559, 2006.

[46] J. Corum, D. Grady, S.-L. Wee and C. Zimmer, "Coronavirus Vaccine Tracker," *The New York Times,* 28 August 2020.

[47] Biotech and Pharma, "Moderna gets further $472 million U.S. award for coronavirus vaccine development.," *CNBC,* 27 July 2020.

[48] H.-Z. Chen, L.-L. Tang, J. Zhou, Y.-F. Chang and X. Wu, "Bioinformatics analysis of epitope-based vaccine design against the novel SARS-CoV-2," *Infectious Diseases of Poverty,* vol. 9, no. 88, 10 July 2020.

[49] A. Kumar, S. Chalal and F. Hussain, "Development of biomimetic electrospun polymeric biomaterials for bone tissue engineering.," *Journal of biomaterials science, polymer edition.,* 2019.

[50] J. Mohajon, "Confusion Matrix for Your Multi-Class Machine Learning Model," Towards Data Science, [Online]. Available: https://towardsdatascience.com/confusion-matrix-for-your-multi-class-machine-learning-model-ff9aa3bf7826.

[51] M. Hoffman and S. Tenny, "Relative Risk," 2020. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK430824/figure/article-28324.image.f1/.

[52] Principles of Epidemiology in Public Health, "Centers for Disease Control and Prevention," 2012. [Online]. Available: https://www.cdc.gov/csels/dsepd/ss1978/lesson3/section5.html.

12. SUPPLEMENTARY MATERIAL (ANNEXES)

12.1 Annex A

Case (1) rules:

Rule: Extracellular -> Alpha-1
Support: 0.001
Confidence: 0.965
Lift: 9.159
==========
Rule: Extracellular -> C-type lectin
Support: 0.001
Confidence: 0.819
Lift: 7.775
==========
Rule: Extracellular -> Cadherin 1
Support: 0.002
Confidence: 1.0
Lift: 9.486
==========
Rule: Extracellular -> Fibronectin type-III 1
Support: 0.002
Confidence: 0.827
Lift: 7.849
==========
Rule: Extracellular -> Ig-like C1-type
Support: 0.001
Confidence: 0.832
Lift: 7.896
==========
Rule: Extracellular -> Ig-like C2-type
Support: 0.002
Confidence: 0.968
Lift: 9.187
==========
Rule: Extracellular -> Ig-like V-type
Support: 0.001
Confidence: 0.947
Lift: 8.984
==========
Rule: Extracellular -> N-linked (GlcNAc...) asparagine
Support: 0.001
Confidence: 0.781
Lift: 7.415
==========
Rule: Extracellular -> Sema
Support: 0.001
Confidence: 1.0
Lift: 9.486
==========
Rule: NR LBD -> HELIX
Support: 0.001
Confidence: 0.597
Lift: 3.008

Case (2) rules:

Rule: Alpha-1 -> Extracellular
Support: 0.001
Confidence: 0.965
Lift: 9.159
==========

Rule: Beta-1 -> Extracellular
Support: 0.0008
Confidence: 0.922
Lift: 8.750

Rule: C-type lectin -> Extracellular
Support: 0.001
Confidence: 0.819
Lift: 7.775
==========

Rule: Extracellular -> Cadherin 1
Support: 0.002
Confidence: 1.0
Lift: 9.486
==========

Rule: Extracellular -> Cys-rich
Support: 0.0007
Confidence: 0.972
Lift: 9.224
==========

Rule: Cytoplasmic -> GS
Support: 0.0005
Confidence: 1.0
Lift: 12.383
==========

Rule: TIR -> Cytoplasmic
Support: 0.0007
Confidence: 1.0
Lift: 12.383
==========

Rule: Tyrosine-protein phosphatase 1 -> Cytoplasmic
Support: 0.0007
Confidence: 0.900
Lift: 11.145
==========

Rule: Extracellular -> Cytosolic Ser/Thr-rich junction
Support: 0.0006
Confidence: 1.0
Lift: 9.486
==========

Rule: EGF-like 1 -> Extracellular
Support: 0.0006
Confidence: 0.767
Lift: 7.283
==========

Rule: Eph LBD -> Extracellular
Support: 0.00067
Confidence: 1.0
Lift: 9.486
==========

Rule: Extracellular -> Fibronectin type-III
Support: 0.0005
Confidence: 0.835
Lift: 7.927
==========

Rule: Fibronectin type-III 2 -> Extracellular
Support: 0.001
Confidence: 0.844
Lift: 8.016
==========

Rule: Ig-like C1-type -> Extracellular
Support: 0.001
Confidence: 0.832
Lift: 7.896
==========

Rule: Ig-like C2-type -> Extracellular
Support: 0.0009
Confidence: 0.940
Lift: 8.925
==========

Rule: Ig-like V-type -> Extracellular
Support: 0.001
Confidence: 0.947
Lift: 8.984
==========

Rule: Extracellular -> Interchain (between HA1 and HA2 chains)
Support: 0.0007
Confidence: 1.0
Lift: 9.486
==========
Rule: LRRCT -> Extracellular
Support: 0.0006
Confidence: 0.895
Lift: 8.498
==========
Rule: N-linked (GlcNAc...) asparagine -> Extracellular
Support: 0.001
Confidence: 0.781
Lift: 7.415
==========
Rule: Sema -> Extracellular
Support: 0.001
Confidence: 1.0
Lift: 9.486
==========
Rule: VWFA -> Extracellular
Support: 0.0008
Confidence: 0.873
Lift: 8.285
==========
Rule: N-terminal globular head -> HELIX
Support: 0.0005
Confidence: 0.708
Lift: 3.568
==========
Rule: Ig-like C1-type -> Alpha-1
Support: 0.0007
Confidence: 0.956
Lift: 9.073
==========
Rule: Alpha-1 -> Extracellular
Support: 0.0008
Confidence: 0.964
Lift: 9.146
==========
Rule: DISULFID -> Ig-like C2-type 1
Support: 0.0006
Confidence: 0.976
Lift: 9.263
==========