processes

MDPI

Article

A machine learning based pipeline for the classification of CTX-M in metagenomics samples

Diego Ceballos [1-2], Diana López-Álvarez [3], Gustavo Isaza [2], Reinel Tabares-Soto [1], Simon Orozco-Arias[1-2] and Carlos D. Ferrin [4]

[1] Universidad Autónoma de Manizales, Manizales, Colombia
[2] Universidad de Caldas, Manizales, Colombia
[3] Universidad Nacional de Colombia - Palmira, Colombia
[4] Universidad de Valle, Cali, Colombia
* Correspondence: diego.ceballos@ucaldas.edu.co (D.C.); dianalopez430@gmail.com (D.L.A.); gustavo.isaza@ucaldas.edu.co (G.I.); rtabarez@autonoma.edu.co (R.T); simon.orozco.arias@gmail.com (S.O.A.); cdfbdex@gmail.com (C.D.F.)

Received: 16 February 2019; Accepted: date; Published: date

Abstract: Bacterial infections are a major global concern because they can lead to a public health problem. In that area, bioinformatics contributes extensively with the analysis and interpretation of "in silico" data by characterizing different individuals/strains from a genetic point of view, such as the bacteria mentioned above. However, the growing volume of metagenomic data requires of new infrastructures, technologies, and methodologies that support the analysis and prediction of this information from a clinical point of view as it is intended with this work. On the other hand, distributed computational environments allow the management of these large volumes of data, due to significant advances in processing architectures such as multicore CPU and GPGPU. For this purpose, we developed a bioinformatic flow from the filtering of the metagenomic data with the Duk tool. The formatting of these data was done through the emboss software and a prototype of a workflow. A pipeline was also designed and implemented in bash script based on machine learning. Then, Python 3 programming language was used to do the normalization of the training data of the artificial neural network and later implemented in the TensorFlow framework and the behavior of the Artificial Neural Networks was visualized in TensorBoard. Finally, the values from the initial bioinformatics process and the data generated during the parameterization and optimization of the Artificial Neural Networks are presented and evidenced with the most optimal result for the identification of the CTX-M gene group.

Keywords: machine learning; metagenomics; bioinformatics; CTX-M

1. Introduction

Within the area of bioinformatics, researchers use metagenomics that seeks to characterize microbial genomes using direct isolation from the environment (Katharina J Hoff et al, 2008). In the same way, new sequencing technologies generate a large volume of data which need to be analyzed, due to the abundant varieties of species that can be found and even more considering their length and complexity. Also, with the possibility of discovering new species, the problem of taxonomic assignment to reads of short DNA sequences becomes extremely challenging (Zeeshan Rasheed and Huzefa Rangwala, 2012). Having said that, meta-genomics is considered as the field of study of a large number of genomes in different environments that may even be areas or regions of living beings such as mucous membranes and intestines, among others. So, Metagenomics is a challenge for computer science researchers who seek to develop methods to understand such amount of genetic information (Soueidan, Hayssam, 2016). Concerning the area of computational intelligence, this work

deals with a technique already known and validated with artificial neural networks. According to (Soueidan, Hayssam, 2016), machine learning techniques currently offer a large set of promising tools to build predictive models for the classification of biological data. These tools are built under different frameworks offering the possibility of implementing supervised and unsupervised techniques (clustering) among others.

CTX-M-type enzymes are a group of class A extended-spectrum *β-lactamases* (ESBLs) that are rapidly spreading among *Enterobacteriaceae* worldwide. The first recognition of the appearance of *β-lactamases CTX-M* occurred almost simultaneously in Europe and South America in early 1989. The first publication to recognize a BLEE from the *CTX-M* group was a report presenting a species of *E. coli* resistant to cefotaxime but susceptible to ceftazidime isolated from the ear of a 4-month-old child suffering from otitis media in Munich. (Cantón, González, & Galán, 2012).

At the regional level, the Manizales Antibiotic Resistance Group (GRAM) is in charge of presenting the accumulated antibiotic resistance data of the main hospitals in the city. Within the total isolation of patients in intensive citizens, general hospitalization and emergencies, the main bacteria identified are *Enterobacteriaceae* such as *Escherichia coli, Klebsiella pneumoniae, Eneterobacter cloacae*, among others; all these species with the capacity to carry genes of BLEE of the *CTX-M* group. In addition, according to the antibiotic susceptibility analyses carried out by the different clinics in the city, resistance to cefotaxime (cephalosporin with a broad hydrolyzable spectrum by *CTX-M*) ranges between 15 and 35% (Salazar, JD. et al 2018). This means that in Manizales up to one out of every three isolations of this group of bacteria is suspected of carrying a *CTX-M* type BLEE; the high frequency of this type of BLEE in our context highlights the importance of this type of development for antibiotic surveillance processes that take into account metagenomic data.

The validation of this pipeline allows us to point the way to the extension of the analysis for other important genes such as *TEM, SHV, metalloenzines, carbapenemases* type *KPC* or *OXA-48* that are probably prevalent in our regional context considering the characteristics of populations, clinical management protocols of patients and health, asepsis in operating rooms. Since this is a common problem, the development of a pipeline that allows the identification of resistance variants becomes a fundamental step in the establishment of a modern antibiotic surveillance system. The subsequent goal will be to test this development on metagenomic data derived from the surveillance process, in collaboration with research groups for this area of knowledge.

1.1.Metagenomics

According to NCBI (Torsten Thomas et al 2012), metagenomics is an area of bioinformatics that has evolved significantly in the last ten years contributing on a large scale to microbiology. In the same way, this relatively new "omic" science has made surprising discoveries in the microbial taxonomy, revealing new capabilities and functionalities of different biomes (J. Johnson, Kunal Jain and D. Madamwar 2017).

Metagenomics is analyzed through computation and bioinformatics, especially with the use of different information discovery techniques. From this field, we try to discover patterns within these data to extract information that may be relevant for biologists, pharmacologists, chemists or bioinformaticians. This information contributes to the solution of the different pathologies related to microbial attacks.

New techniques have been developed to analyze large volumes of information from a large amount of metagenomic data, being big data and machine learning the most widely used (Chuang Ma, Hao Helen Zhang, Xiangfeng Wang 2014). These techniques use distributed computational environments of large capacity that allow more efficient processing and reducing computing times in a significant way.

1.2. Machine Learning

Machine learning seeks to answer a very concrete question: How can we build computer systems that automatically improve with experience, and what fundamental laws govern this teaching process? (Tom M. Mitchel, 2016).

Through this discipline it is possible to implement new methods that help researchers in making new findings. Machine learning techniques are used, for example, to learn about models of gene expression in cells and other applications in bioinformatics, more specifically in metagenomics (Kevin Vervier et al. 2015). One can talk about three types of algorithms within the current machine learning techniques:

Supervised: Data training consists of labeled entries and known outputs that the machine analyzes while relabeling. There are many applications of supervised algorithms in bioinformatics to solve problems (Pinyi Lu et al., 2015), knowing that many of the genes are already adequately characterized.

Unsupervised: This type of analysis of unlabeled and categorized data is based on similarities that have been identified. In this case the machine can cluster the data based on shared characteristics. Techniques that use unsupervised algorithms are often used for problems in which the human being cannot infer patterns clearly, that is, it requires exhaustive observation to identify such patterns. It is also a technique that allows to determine behaviors based on different interpretations.

Semi-supervised: This analysis refers to a combination of the two previously mentioned techniques. It is used in large data sizes when the labels of some of these data are known. Unsupervised learning is based on the analysis of unlabeled data to group them, while techniques of supervised learning are used to predict the labels of this group formed by the first technique. Artificial Neural Networks (ANN) are a known approach to address complex problems, neural networks can be implemented at the hardware or software level and in turn can use a variety of topologies and learning algorithms.

2. Materials and Methods

2.1. Selection of the CTX-M and metagenome baseline reference database for the study

First, we based on Núñez previous work in 2016 (unpublished data) where all the *CTX-M* reported groups are already considered. Then, after a review of state of the art, we started from the *CTX-M* database previous filtered by the analysis of phylogenetic trees carried out by Núñez. The reference metagenome to be studied was selected from a review in the EBI-Metagenomics database (https://www.ebi.ac.uk/metagenomics/) considering the high probability that the *CTX-M* gene was present when reviewing four metagenomes related to the next entry to make the prototype. The metagenome selected for the training of the subsequent proposed neural network was:

Subsequently, the reference metagenome to be studied was selected by means of a revision in the EBI-Metagenomics database (https://www.ebi.ac.uk/metagenomics/) considering the high probability that the *CTX-M* gene was present by reviewing four metagenomes related below which are inputs to make the prototype, only one was selected for the study:

1. https://www.ebi.ac.uk/metagenomics/projects/ERP001506
2. https://www.ebi.ac.uk/metagenomics/projects/ERP020191
3. https://www.ebi.ac.uk/metagenomics/projects/ERP016968
4. https://www.ebi.ac.uk/metagenomics/projects/ERP009131

Antibiotic resistance within the preterm infant gut: https://www.ebi.ac.uk/ena/data/view/PRJEB15257 Next, the process of identifying the pipeline for filtering the data was proposed following these steps:

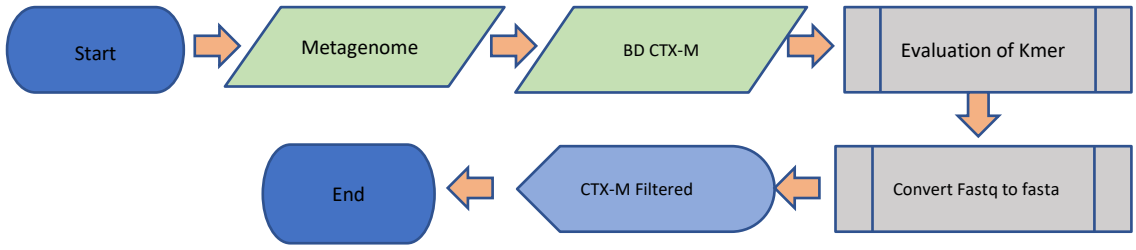


Figure 1. Details of the bioinformatic pipeline.

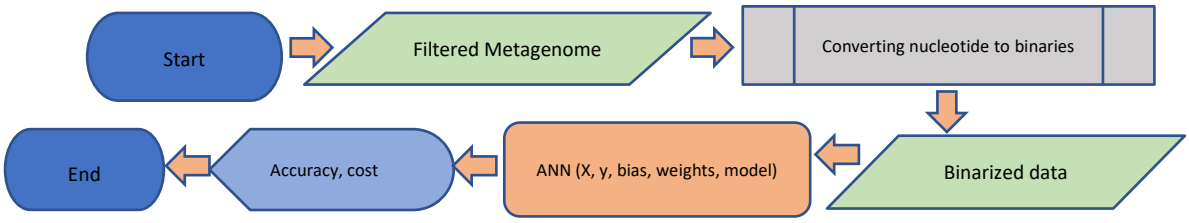


Figure 2. Details of the computational pipeline.

The figure 2 describe de steps of the computational pipeline where the data are prepared and technique of machine learning is applied, later is appraised the accuracy and the cost of the artificial neural network. A brief description is: Filtered metagenome, consist in the input data filtered in the first pipeline, converting nucleotide to binaries and Binarized data is the process to transform the data for the ANN, then the artificial neural network in execution and accuracy and cost is the metrics for be evaluated.

The *CTX-M* database is consolidated in a fasta format with a total of 211 reference sequences using the Duk tool (Li, Mingkun, et al., 2018) which allows mapping between the sample metagenome and the *CTX-M* reference database to eliminate information not relevant for the study. To achieve this, we used a kmer of 16 (default) and another of 63 as initial tests. The following recommendations were taken into account to validate these results (algorithm 1):

Parameterize the initial mapping with Duk using odd Kmers.
Execute tests using different K-mers.

Name: Pre-filter *CTX-M*
Start
For kmer values between 17 and 65
Do
Execute duk with each k-mer against the reference database
Save results in a single file “duk_results”
Finish do
Best_K-mer <- 0
Best P-value <- 0
For each line in “duk_results” file
Do
Find P-value of each k-mer
If (P-value found is larger than Best P-value)

```
                Best P-value <- P-value found
                Best_K-mer <- k-mer found
            End if
            Convert output file of best k-mer to fasta format
            Format the fasta file for the ANN (X, y)
            For each end of CTX-M sequence
            Do
                Separate CTX-M group from each sequence.
        Finish do
    End
```

Algorithm 1. Bioinformatic pipeline for filtering and formatting input data.

With the initial analysis, k-mer 17, 19 and 21 are the best detected. Additionally, validation of base against NCBI Blast of the contig obtained is executed after adjusting the kmer to 17 and 19 to conclusively verify that this fasta refers to bacteria with the *CTX-M* gene. The pipeline can be download here: https://github.com/dhcl1580/machinelearniginmetagenomicstesis

2.2. Defining an optimal neural network architecture

An exhaustive review was performed of the existing literature to define the architecture of the neural network for metagenomics. Different machine learning models focused on improving the precision of the techniques applied in neural networks such as random forest, or algorithms based on decision trees were evaluated too (Milko Krachunov et al., 2017). In none of these works a particular architecture is taken for granted, where it is understood that the main goal is to obtain a reduction in the cost function in order to guarantee that the neural net-work apprenticeship is being carried out taking into account the decreasing behavior of the function already mentioned. For all the above, this study proposes an architecture of a multi-layer perception neuronal network (Figure 3) because of the importance of the high sensitivity that different neurons show in each of their layers concerning the activation functions, weights, and epochs. This interaction allows to consider more parameters when training and validating such architecture taking into account its performance (Xiaoquin Zeng, Daniel S. Yeung. 2001).

In order to establish an appropriate dataset for the training of the proposed neuronal network, a routine was developed in Python 3 in charge of normalizing the data obtained where basically a binarization of the *CTX-M* nucleotide sequences is carried out.

In general terms, binarization consists of converting the nucleotides to their corresponding binary value by means of a function called vectorizeSequence and their correspondence to the binary value is established, thus defining the input values (X) for the neural network, to generate the Y (*CTX-M* groups) the data are discriminated after the comma separator (,).

All sequences are standardized to the value of the longest identified sequence and additional spaces are defined by the value N.

The result is the file "dataGen.csv" where a total of 3896 values are generated for X and the 10 *CTX-M* groups (Classes).

The 10 most representative classes were selected in order to ensure a uniform distribution of classes for stratified cross validation in stage 2 (validation). Initially there were 17 classes from which only those with sequences represented at least 4 times within the test and validation data set were selected,

Each of the 10 classes corresponds to the following *CTX-M* groups respectively

Group *CTX-M*	Class
1.0	0
9.0	1
14.0	2
15.0	3
22.0	4
24.0	5
27.0	6
55.0	7
59.0	8
65.0	9

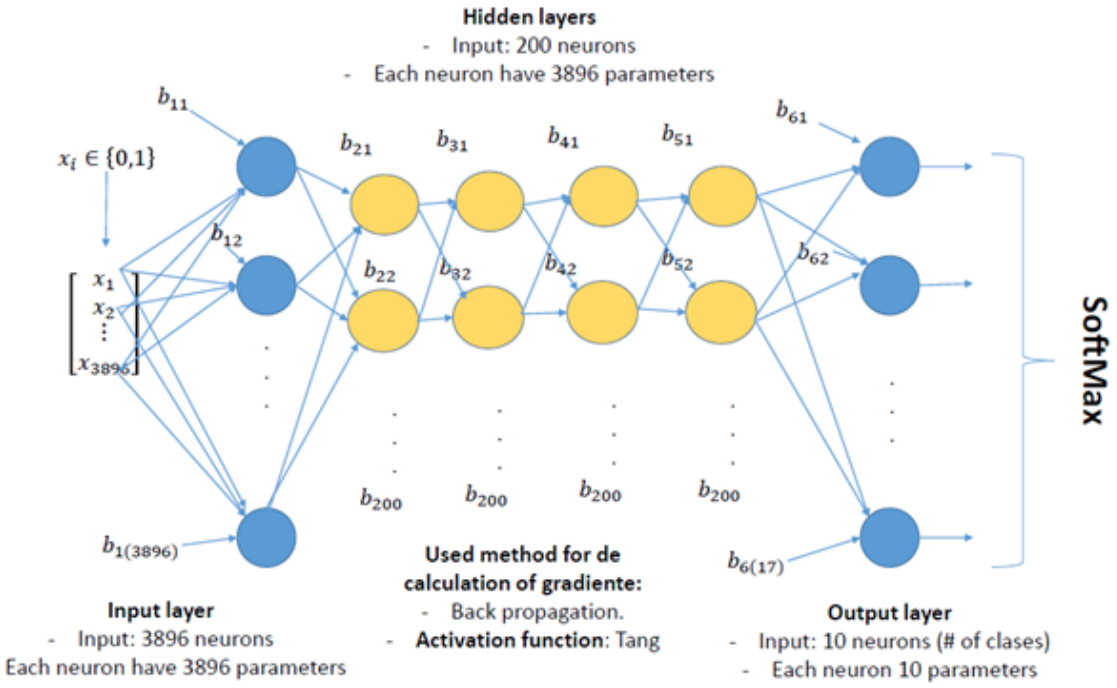


Figure 3. Details of the ANN components; the cost, accuracy, optimization and model definition tensors.

2.3. Data standardization for the neural network

To establish an appropriate dataset for the training of the proposed neuronal network, we developed a routine in Python 3 in charge of normalizing the data obtained where basically a binarization of the *CTX-M* nucleotide sequences is carried out. All sequences are standardized to the value of the longest identified sequence, and additional spaces are defined by the value N. The result is the file "dataGen.csv" where a total of 3896 values are generated for X and the 10 groups of *CTX-M* (Classes).

3. Analysis of results

3.1. Analysis of the graph resulting from ANN

The figure 4 show how the graph of the ANN is build, in this graph is possible observe how the nodes are distributed and the interaction of this for the process data.

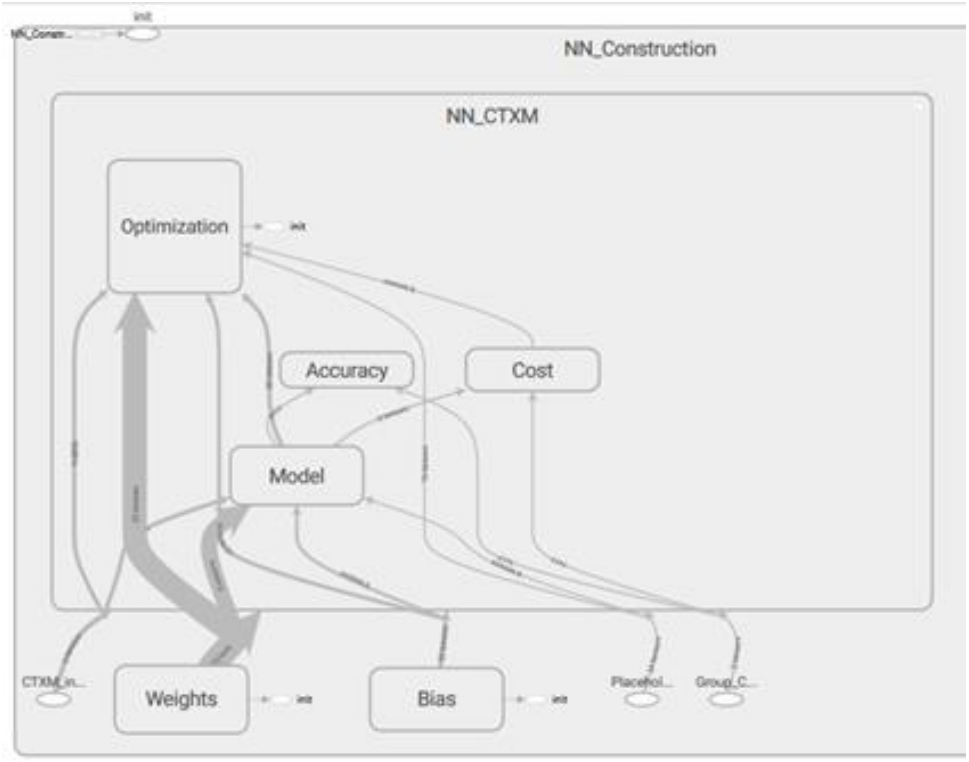


Figure 4. Details of the ANN components and the cost, accuracy, optimization and model definition tensors.

3.2. Training stage over CPU

The activation functions tanh and sigmoid were experimented with RELU where the parameters LEARNING_RATE, TRAINING_EPOCHS, HIDDEN_SIZE were varied, obtaining the results presented below for each function. Each of the 10 classes corresponds to the following CTX-M groups respectively (taking into account the metrics presented for ROC Analyses):

Table 1. Summary of target values during the raining stage under CPU.

Activation function	LEARNING_RATE	TRAINING_EPOCH	HIDDEN_SIZE	Initial cost value	Final cost value	Accuracy of initial training	Accuracy of final training	Precision test
Tanh	0.001	400	200	2.17	0.80	0.260	0.960	**0.879**
Sigmoid	0.001	400	200	2.19	1.61	0.030	0.680	**0.698**
RELU	0.001	300	200	2.19	0.00	0.110	1	**1**

The best values were obtained using the tanh activation function.

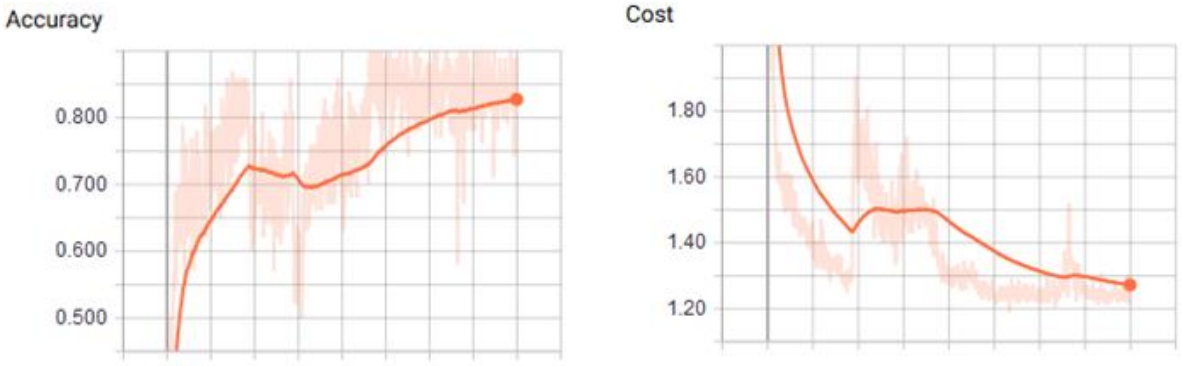


Figure 5 and 6. Values of accuracy and cost using tanh function over CPU.

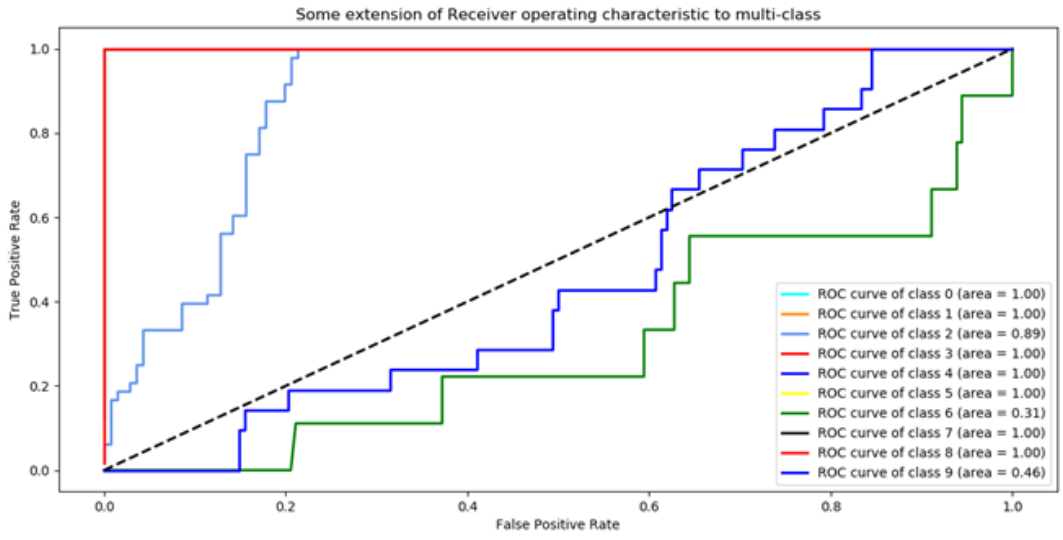


Figure 7. ROC analysis for the tanh activation function over CPU.

Table 2. Summary of target values during the raining stage under GPU.

Activation function	*LEARNING_RATE*	*TRAINING_EPOCH*	*HIDDEN_SIZE*	*Initial cost value*	*Final cost value*	*Accuracy of initial training*	*Accuracy of final training*	*Precision test*
Tanh	*0.001*	*400*	*200*	*2.16*	*0.84*	*0.380*	*0.920*	***0.909***
Sigmoid	*0.001*	*400*	*200*	*2.20*	*1.67*	*0.440*	*0.560*	***0.628***
RELU	*0.001*	*300*	*200*	*1.90*	*1.00*	*0.590*	*1*	***1***

The best values were obtained using the tanh activation function.

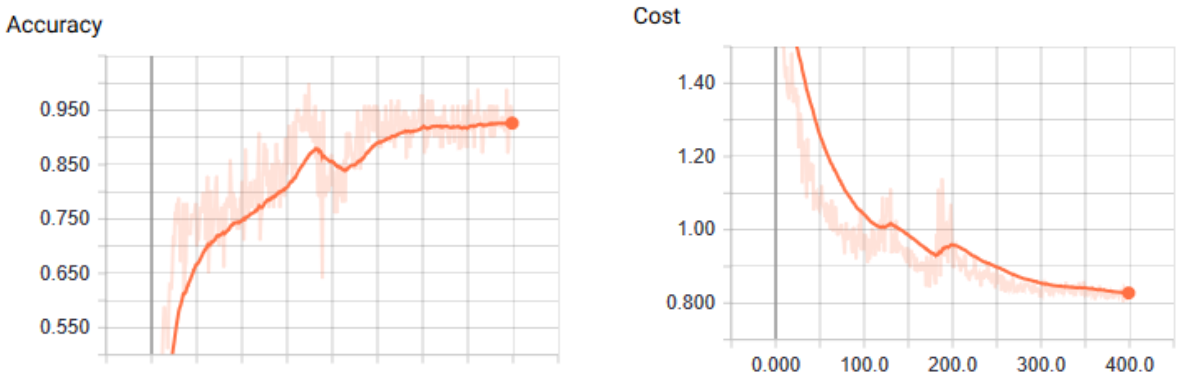


Figure 8 and 9. Values of accuracy and cost using tanh function over GPU.

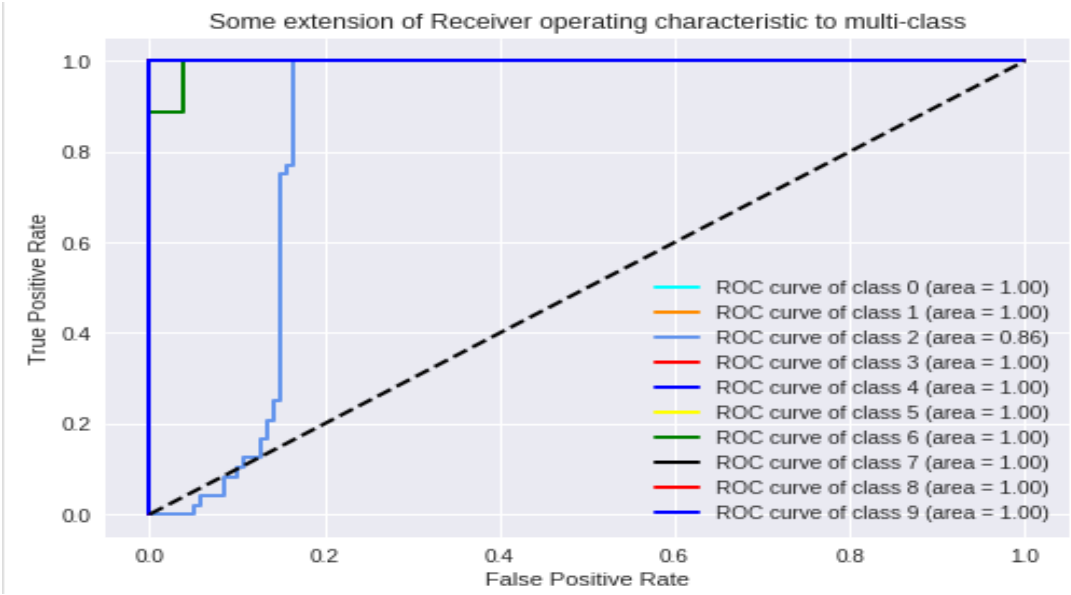


Figure 10. ROC analysis for the tanh activation function over GPU.

4. Discussion

4.1. Conclusions for the tanh activation function

We found that ANN had the most optimal behavior under the tanh activation function for the training stage. The reason is that the reference value for the precision test with the variation of the Training epoch and Hidden size parameters was 0,879. Precision and cost behaviors were the expected taking into account that the cost decreased and the precision increased for all the evaluations proposed with different parameters. Another relevant conclusion is that according to the ROC analysis the class that is least likely to be identified under these ANN parameters is class 2 and class 6.

4.2. Conclusions about the dataset

Regarding the dataset we can conclude that for future work it is advisable to consider more *CTX-M* contigs. For this work the 10 most representative groups were considered, some of the groups were not representative enough to be able to carry out a stratified cross validation, especially for the experimentation in the validation stage, in which 20% of initial dataset is used for this validation. Regarding the data set, we can conclude that more *CTX-M* contigs can be considered for future studies.

4.3. Perspective

We propose to validate in a future study a more significant number of meta-genomes corresponding to the geographical area of influence to support the design of public policies related to the prevention and detection of infectious diseases. Other types of metrics especially histograms would be considered, taking advantage of the fact that they can be generated by the TensorBoard tool, to corroborate the final results more accurately. Finally, it is recommended to continue with the training process so that this software can identify a higher number of infectious diseases with the same characteristics as *TEM, SHV, metalloenzymes, carbapenemase*s type *KPC* or *OXA-48*.

Author Contributions: Conceptualization, Diego Hernando Ceballos López, Diana Carolina López Álvarez, Gustavo Isaza Mejía and Carlos Diego Ferrín Bolaños. Formal analysis, Diana Carolina López Álvarez, Gustavo Isaza Mejía, Simón Orozco Arias and Carlos Diego Ferrín Bolaños. Investigation, Diego Hernando Ceballos López. Methodology, Diego Hernando Ceballos López; Software, Simón Orozco Arias, Reinel Tabares Soto and Carlos Diego Ferrín Bolaños. Supervision, Diana Carolina López Álvarez and Gustavo Isaza Mejía. Writing – original draft, Diego Hernando Ceballos López. Writing – review & editing, Diana Carolina López Álvarez, Gustavo Isaza Mejía, Simón Orozco Arias and Carlos Diego Ferrín Bolaños.

References

1. Torsten Thomas, Jack Gilbert and Folker Meyer (2012). Metagenomics - a guide from sampling to data analysis
2. Katharina J Hoff, Maike Tech, Thomas Lingner, Rolf Daniel, Burkhard Morgenstern and Peter Meinicke (2008). Gene prediction in metagenomic fragments: A large scale machine learning approach
3. Zeehasham Rasheed and Huzefa Rangwala (2012). METAGENOMIC TAXONOMIC CLASSIFICATION USING EXTREME LEARNING MACHINES
4. Cantón, R.;, González, J.M.;, & Galán, J.C. (2012). CTX-M enzymes: origin and diffusion Frontiers in microbiology 3(110), 1-19.
5. Haysaam Soueidan, Nacha Nikolski. (2016). Machine learning for metagenomics: methods and tools.
6. J. Johnson, Kunal Jain and D. Madamwar, 2 - Functional Metagenomics: Exploring Nature's Gold Mine, In Current Developments in Biotechnology and Bioengineering, Elsevier, 2017, Pages 27-43, ISBN 9780444636676, http://dx.doi.org/10.1016/B978-0-444-63667-6.00002-X. (http://www.sciencedirect.com/science/article/pii/B978044463667600002X)
7. Chuang Ma, Hao Helen Zhang, Xiangfeng Wang, Machine learning for Big Data analytics in plants, Trends in Plant Science, Volume 19, Issue 12, December 2014, Pages 798-808, ISSN 1360-1385, http://dx.doi.org/10.1016/j.tplants.2014.08.004
8. Tom M. Mitchell. July 2006. The Discipline of Machine Learning. CMU-ML-06-108. School of Computer Science. Carnegie Mellon University. Pittsburgh, PA.
9. Kevin Vervier, Pierre Mahe, Maud Tournoud, Jean-Baptiste Veyrieras, Jean-Philippe Vert. Large-scale Machine Learning for Metagenomics Sequence Classication. 2015. <hal- 01151453>
10. Pinyi Lu, Vida Abedi, Yongguo Mei, Raquel Hontecillas, Casandra Philipson, Stefan Hoops, Adria Carbo and Josep Bassaganya-Riera, ISBN 9780128025086.
11. Ana Nuñez (2016). Anábioimutendifetide blaCTX-M



© 2019 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).