Exploration of multiclass and one-class learning methods for prediction of phage-bacteria interaction at strain level

Diogo Manuel Carvalho Leite*†, Juan Fernando Lopez¶, Xavier Brochet*†, Miguel Barreto-Sanz*†, Yok-Ai Que‡, Grégory Resch§, Carlos Peña-Reyes*†
*School Of Business and Engineering Vaud (HEIG-VD), University of Applied Sciences Western of Switzerland (HES-SO) Route de Cheseaux 1, Switzerland, Yverdon-les-Bains 1400
† SIB - Swiss Institute of Bio-informatics
‡ Department of Intensive Care medicine, Bern University Hospital (Inselspital), Freiburgstrasse, 3010 Bern
§Department of Fundamental Microbiology University of Lausanne, 1015 Lausanne
¶Faculty of engineering, department of automatics and electronics Universidad Autónoma de Occidente, Cali, Colombia
Email: * carlos.pena@heig-vd.ch

***Abstract*—The emergence and rapid dissemination of antibiotic resistance threatens medical progress and calls for innovative approaches for the management of multidrug resistant infections. Phage-therapy, i.e., the use of viruses that specifically infect and kill bacteria during their life cycle, is a re-emerging and promising alternative to solve this problem. The success of phage therapy mainly relies on the exact matching between both the target pathogenic bacteria and the therapeutic phage. Several papers propose models for in-silico prediction of phage-bacteria interactions but at the species level. In clinical applications prediction of phage-bacteria interaction at species level is not enough to target a given pathogenic bacteria strain.**

One of the main challenges to train classification models able to predict phage-bacteria interactions is the need of both types of samples: interaction and non-interaction phage-bacteria couples. Non-interactions are rarely reported, making these data scarce. This problem is even more evident for non-interaction data at strain level. These factors make difficult the use of classic machine learning algorithms which need relatively-balanced classes to produce accurate predictions.

This problem calls for solutions to deal with such imbalanced data. In this paper are presented two approaches to tackle this problem. 1. To explore the use of One-Class learning methods 2. To generate putative non-interacting data and use single and ensemble-learning approaches, to predict phage-bacteria interaction at strain-level.

***Index Terms*—Phage-bacteria interaction, in-silico prediction of phage-bacteria interaction, phage therapy, one-class learning, ensemble learning**

I. Introduction

The rapid emergence of bacteria resistant to antibiotics is actually a worldwide problem calling for novel approaches to manage this global health issue [1]. This situation is a consequence of several factors. On the one hand, the abuse of antibiotics consumption clearly drives into an evolution of resistant bacteria [2]. On the other hand, the research and development of new molecules by the pharmaceutical industry is not following the same rate as the emergence of new bacteria resistant to antibiotics. [3].

Phage-therapy is one of the most promising re-emergent alternatives to treat bacterial infections. Phage-therapy use antimicrobial agents called bacteriophages which are viruses able to infect and kill pathogenic bacteria along its life cycle [4]. Bacteriophages (or phages) have cohabited and evolved with bacteria for billions of years, controlling bacterial epidemics, and population in a continuous genetic exchange [5]. Phages only attack bacteria and have a high specific range of infection, so they can only recognize a small range of bacterial strains [6]. This behaviour has several benefits, for instance it limits the damage in the commensal flora in opposite to the antibiotics [7].

In order to infect their bacterial hosts, a phage attaches to a bacterium and inserts its genetic material into the cell. After that, a phage usually follows one of two life cycles lytic (virulent) or lysogenic (temperate). On the one hand, virulent phages enter directly the lytic cycle, which yields the formation of new phages and their release by cell membrane cleavage, which therefore kills the bacteria. On the other hand, temperate phages instead undergo the lysogenic cycle, which integrates the injected genome into that of the host (called prophage), which therefore replicates at the pace of bacterial divisions, until conditions change and the phage eventually enters the lytic cycle (e.g. after cell damage) [8]. Conventional phage therapy relies on strictly lytic phages, which obligately kill their bacterial host. For treatment, lytic phages can be used individually or compiled into preparations called phage cocktails which consist of multiple phages proven to have in vitro efficacy against the target pathogen.

The mechanics of attack and defence of phages and bacteria are constantly evolving [9]. Some of the bacteria defence methods consist in rending phage receptors unrecognizable

This project is funded by the Swiss National Science Foundation (FNS).

for the phage trough a mutation of them. A variation of this type of defense is to hide the bacteria receptors binding regions with capsules as physical barriers [10]. Another mechanism rely on the ability of some bacteria to detect geneticaly-encoded sites that could be targeted by a restriction-modification system which cuts stranger DNA at specific recognition sites (e.g., the CRISPR/Cas system) [11].

The spectrum of strains of bacterial species that a given phage strain can infect is called "host range". Identifying the host range of a given phage is the first challenge in phage therapy. [12]. This allows identifying the phage or phages than can kill a given bacteria and use them in clinical applications. Currently the "host range" of a phage is identified by culture-based methods, for instance growing the host bacteria with phages on an agar plate and observing plaques, clear areas where the phages killed the host bacteria, and where the phage can be isolated [13]. Other techniques include PhageFISH, viral tagging, microfluidic PCR, single cell sequencing, and liquid assays. [14], [15].

Independently of the method used, there are two main drawbacks with these techniques: (i) Growing phages in a lab requires appropriate conditions such as chemical supplements, temperature, and specific grow media. (ii) The time to perform these processes is dependent on the number of bacteria and phages to be tested, usually they may take several days. In clinical applications as phage therapy the time to find the right phage is critical. Thus, new methods must be explored to find the right phage in the right time to kill a pathogenic bacteria.

In-silico methods to find the host range of a phage based on the prediction of phage-bacteria interactions is a promising approach that can help to tackle the aforementioned challenges. For instance, HostPhinder is a phage host prediction tool which can receive a phage genome as input, compare it with other phage genomes on its database and give as result a list of bacterial host species based on the phage genome similarity. [14]. Other computational approaches have been developed aiming at predicting phage-bacterium interactions. Coelho et al [16] created a method based on protein-protein interactions (PPIs), Edwards et al, propose a method based on similarity techniques like BLAST [15].

The aforementioned methods present two main limitations:

1) The theoretical background of these methods rely on genomic similarities of groups of phages. Thus, phages with similar genomic sequences are predicted as more likely to interact with related bacterial hosts. The main limitation of these approaches is that they are based on a measure of similarity. It posses some limitations to prediction of interactions. for instance when a new type of phage with a very different genomic sequence to those in the database will be introduce, it will be barely assign to cluster of phage since there are not phages similar to it. A better approach to predict phage-bacteria interactions is to learn about the mechanisms of defence and attack from the couples phage-bacteria (when they interact and do not) and use algorithms to extract rules and patterns to predict when a phage will attack or not a bacteria.
2) Another disadvantage to use these methods is that they predict interactions at species level. In clinical applications prediction of phage-bacteria interaction at a species-level is not enough to target a given pathogenic bacteria strain.

In this paper is presented an approach that is more suitable for clinical use of phages. We developed a pipeline (i.e. data gathering, data preparation, feature engineering, models building and models testing) based on multiclass and one-class techniques to predict phage-bacterium interactions at strain level.

The first step consisted on gathered genome sequences of phages and bacteria from GenBank [17] and PhageDB [18] based on several criteria as the confirmation of the phage-bacteria interaction. In the feature engineering we extracted the distribution of predicted protein-protein interaction scores, as well as the amino acid frequency, the chemical composition, and the molecular weight of such proteins. In modelling building we used multiclass and one-class learning approaches. To train multiclass classification models is necessary to have two kind of samples, namely: interaction and non-interaction phage-bacteria couples. Usually databases as NCBI and phageDB, do not report non-interaction. This data is scarce, and it must be gathered, directly from scientific papers or performing own experiments in the lab. This problem is even more evident for interactions at strain level. Two approaches are presented to tackle this problem. 1. To explore the use of five one-class learning techniques. 2. To generate putative non-interacting data and use well known multiclass techniques to predict interactions. In addition, to combine the multiclass techniques in odd groups to create ensemble-learning models.

II. Methods

A. Data gathering

1) Organisms selection: We selected from public annotated databases (i.e. GenBank [17] and PhageDB [18], accessed in November 2017) phages and bacteria with reported interactions and complete genomes. As only positive interactions are available in both databases, and negative interactions are needed to train multiclass classification algorithms, we generated putative non-interactions based on explicit theory. The decision in grounded on the fact that phages are strain specific, except for some rare known exceptions that infect and kill a wide range of bacteria. For example the bacteriophage Mu is able to infect species of Escherichia coli, Citrobacter freundii, Shigella sonnei,

Enterobacter, and Erwinia [19].

In this context, we selected a phage from the database only if:

- Its complete genome is available.
- Its host information is known.
- The complete genome of its host is available.

As a result, we created a database composed of 1715 bacteria (all from GenBank) and 3 747 Phages (2 005 from PhageDb and 1 742 from GenBank). Thus, with a total of 5 462 organisms and 20 287 positive interactions (2 297 at strain level and 17 990 at species level).

2) Collecting organisms' genomes: For each organism selected in the previous step, we collected its complete genome. In the case of phages, their genomes come from PhageDB [18]. To obtain supplementary information as coding-DNA and protein sequence, we performed gene prediction using GenMarkS [20]. We collected from GenBank the genomes of the 1 742 selected phages querying the Entrez Nucleotide web-service [21] with: *phage [title] and complete genome*, *coding-DNA* and, *protein sequence*. We obtain the bacterial host range parsing the annotation in phageDB [18] and GenBank [17] extracting *Isolation host* and *host* fields from databases.

To retrieve the bacteria genomes we also used the Entrez service [21] of NCBI using the following query: *name of bacteria [ORGN] AND complete genome*.

3) Positive interactions: We collected 20 287 positive interactions at species level. To train the models we only used interactions at strain level, specifically: 2 297 interactions from 98 different bacteria. The bacteria used regroup 33 families, 44 genus, and 60 species, as is described in Table I.

TABLE I
TAXONOMY DISTRIBUTION

Family name	Genus	Species	Strains	Interactions
Bacillaceae	1	1	2	2
Brucellaceae	1	3	3	3
Burkholderiaceae	1	1	1	1
Campylobacteraceae	1	1	1	1
Can. Puniceispirillum	1	1	1	1
Caulobacteraceae	1	1	1	1
Clostridiaceae	1	1	4	8
Comamonadaceae	1	1	1	1
Enterobacteriaceae	6	6	18	28
Enterococcaceae	1	1	1	1
Flavobacteriaceae	3	3	6	17
Gordoniaceae	1	1	2	260
Lactobacillaceae	1	3	3	4
Leuconostocaceae	1	2	4	9
Micrococcaceae	1	1	1	166
Moraxellaceae	1	1	3	3
Mycobacteriaceae	1	3	3	1633
Paenibacillaceae	1	1	1	1
Pasteurellaceae	1	1	1	1
Pectobacteriaceae	1	1	1	1
Pelagibacteraceae	1	1	1	4
Prochloraceae	1	1	3	5
Propionibacteriaceae	1	1	1	52
Pseudoalteromonadaceae	1	2	2	2
Pseudomonadaceae	1	2	9	33
Rhizobiaceae	1	1	1	1
Rhodobacteraceae	4	4	4	7
Staphylococcaceae	1	1	2	2
Streptococcaceae	2	4	7	7
Streptomycetaceae	1	6	7	36
Synechococcaceae	1	1	1	4
Thermaceae	1	1	1	1
Vibrionaceae	1	1	1	1
Total	44	60	98	2 297

4) Negative interactions: As previously mentioned the public databases consulted (i.e. GenBank [17] and PhageDB [18]) only contain information about positive interactions. Thus, based on the assumption that phages are strain specific, we created our negative dataset based on two rules: (1) if a phage interact with a specie, no negative-interaction of this specific phage with this bacterium will be generated (2) a phage only attacks one bacterial species. Using this approach we created 20 586 negative pairs. The datasets created have the same number of positive and negative pairs of interactions. In addition, to maintain an equilibrium of the quantity of data per bacterium, we generated the same amount of non interactions per species.

B. Feature engineering

As in all organisms, protein-protein interactions (PPIs) play a fundamental role in phage biology. PPIs are important in all virus-related processes, including host infection, transcriptional regulation, DNA replication, virion assembly, and lysis [22]. Hence, it is expected that the information from protein-protein interactions contained in proteins sequences can help to predict phage-bacteria interactions. Based on this hypothesis, we used phage and bacteria proteins' sequences to extract informative features, and then, to train machine-learning algorithms able to predict interactions. We created two sets of features based on primary protein structure information and domain-domain interaction scores. [16], [23].

1) Primary structure sequence: We extracted 27 features from all the proteins based on the primary structure sequences from phage and bacteria proteins [24], namely:

- The percentages of each amino acid. We created a total of 21 features. They are composed by the 20 amino acids plus one to indicate unknown amino acids.
- The abundance of each chemical component in the protein sequence, namely: Carbon, Hydrogen, Nitrogen, Oxygen, and Sulfur [25].
- Its molecular weight in Daltons.

Thus, for each PPI we have 54 features (i.e. 27 from bacterium and 27 from phage). The bacteria in our database encode in average 3575 proteins and for the phages 86 proteins. It is in average $3575*86 \approx 3.07 \times 10^5$ PPIs per phage-bacterium pair. It represents a very high dimensionality. In order to reduce it, we calculated, for each phage-bacterium pair, the mean and the standard deviation of the 54 features across all its PPIs. In summary, each phage-bacterium pair is

represented by 108 features, 54 values for the mean and 54 for the standard-deviation.

2) Domain-Domain interaction (DDI) scores: A protein domain is a structural or functional subunit of a protein that can evolve, function and exist independently of the rest of the protein chain. [26]. An interaction of two proteins can involve one or more binding pairs of domains. Since the majority of the proteins are multi-domain proteins, an interaction between two proteins often involves binding of two or more domains. Thus, understanding protein interactions at the domain level seems to be a logical step towards the prediction of phage-bacteria interactions [27]. For the extraction of DDI features we used DOMINE [27], a database containing known and predicted protein domain interactions compiled from a variety of sources. DOMINE combines 15 sources including experimental methods as PDB crystal structures and others from different predictive models as DIMA, DIPD and RDFF. Experimental information sources of DOMINE include the databases 3did [28] and iPfam [29]. Unfortunately, the last time these sources were integrated to DOMINE was in 2007. In order to obtain the most recent data we extracted the updated version of both databases, (accessed in May 2018). This includes 11 383 DDIs.

To calculate the DDI scores we used the 15 sources from DOMINE in a pondered way. We gave the higher height to 3did and iPfam which predictions since they come from experimental methods (both based on experimentally determined high-resolution three-dimensional structures). The 13 remaining methods based on computational approaches will have the lower height. The rules that govern the calculation of the DDI score are described bellow:

- DDI scores are integer numbers from 0 to 9;
- If the DDI is predicted by 3did or iPfam, the DDI score is 9 (only one is taking into account).
- For any method different to 3did or iPfam the score is 1;
- If a DDI is predicted by 3did or iPfam, the DDI score is 9, even is it is predicted for other method;
- If the DDI is not predicted by 3did or iPfam but is predicted for one or several of the others methods, the score is the sum of the methods.

DOMINE obtain the DDIs from proteins domain definitions using Pfam Hidden Markov Models Profiles [32]. We used the HMMER API to extract the domains of each protein in our database [30]. HMMER is an online service used to search sequence databases for homologs of protein sequences and perform protein sequence alignments. HMMER methods use Profile Hidden Markov Models. HMMER receive as input a protein sequence and return the list of Pfam existing domains. For the 98 bacterium in our database we collected 424 058 domains and, in the case of phages 60 349 domains.

For each phage-bacterium interaction we created a vector of DDIs following the steps bellow:

- Arrange the proteins of both organisms in one vector.
- Verify the existence of the proteins' DDIs according the information in our database.
- Calculate the DDI score from 0 to 9 based on the rules presented previously.
- Create a vector of scores (one score per DDI).

The cumulated interaction score of a PPI is then calculated as the sum of all its DDIs. Using the scoring procedure described before, we obtained a vector of PPIs scores for each phage-bacterium pair. Phage-bacterium pairs does not have the same number of proteins, consequently, the vectors of PPIs scores have different lengths. In order to have vectors of the same size, we transform the "PPIs' vectors" into a "PPIs' vectors of frequency score" (it can be seen as a histogram of PPI-scores).

We tested several configurations based on the variation of two parameters:

- Variation of the histogram bins in two ways: (i) varying the size or bins and (ii) changing the number bins.
- Taking (or not) into account the PPIs with score zero.

The description of the configurations used and its corresponding detasets are described in Table II.

TABLE II
DDIs BASED DATASETS

Bin config.	Use PPI with score zero	Parameter value	Dataset name	Number of features
Size	No	1	SB1	108
	Yes	1	SB_ZEROS_1	108
	No	54	NB54	54
		108	NB108	108
Number	Yes	54	NB_ZEROS_54	54
		108	NB_ZEROS_108	108

3) Datasets for modeling building and modeling validation: In total, we created nine datasets. One based on the primary structure information, named Chemical composition (CH), and eight using DDIs, (as described in Table II). Each dataset is composed by 4 594 pairs phage-bacterium interactions with a half-half distribution of positive and negative interactions.

We split each dataset into 70% for training and validation and the 30% remained for test.

In the case of one-class learning must be take into account that for training 10-Fold cross validation is implemented using only positive interactions in the respective training process but using positive and negative interactions for the validation process.

C. Exploration of Machine-learning approaches

We explored two machine-learning approaches :

- The traditional multiclass classification approach, where the algorithm tries to distinguish between two or more classes with the training set containing objects from all the classes.
- One-class learning methods where the algorithm try to identify objects of a specific class among all objects, by

learning from a training set containing only the objects of that class.

We implement our modeling pipeline in Python 3.6 using Scikit-Learn library on 0.19.1 version [31].

D. Building models and hyperparameters selection

1) Multiclass learning: We tested multiclass classification techniques in single way, and also combined as ensemble-learners. We used Six multiclass machine learning techniques individually, namely : Artificial Neural Networks (ANN) [32], Support Vector machines (SVM) [33], Random Forest (RF) [34], K-Nearest-Neighbors (K-NN) [35], Naive Bayes (NBAY) [36], and Logistic Regression (LR) [37]. We performed a grid search (for the techniques using multiple parameters) using the nine datasets described previously in Table II. We found the best hyperparameters and the datasets that more consistently achieved the highest predictions. The parameters tested for each technique are described in Table III. All along the process we used 10-fold cross-validation in order to prevent overfitting. We calculated the predictive performance using: accuracy, F-score, recall, and precision.

We used heatmaps to visualize the techniques' performances after grid search exploration. We generated a heatmap per each couple (technique, performance measurement). We measured F-score, accuracy, precision and recall. Hence, 44 heatmaps where generated (i.e. 5 one-class techniques * 6 multiclass tecniques * 4 performance measurements).

The Figure 1 shows the heatmap for the couple (Random forrest, F1 score), where rows represent datasets and columns a specific combination of parameters. The heatmaps generated for the rest of techniques and performances' measurements are available on our repository (https://drive.switch.ch/index.php/s/SfPYofpU01BsVQ1).

After analysis of the results we selected for each technique the best parameters found on the grid search. These parameters are presented in Table III highlighted in bold. The techniques with the selected configurations will be used in the modeling test phase.

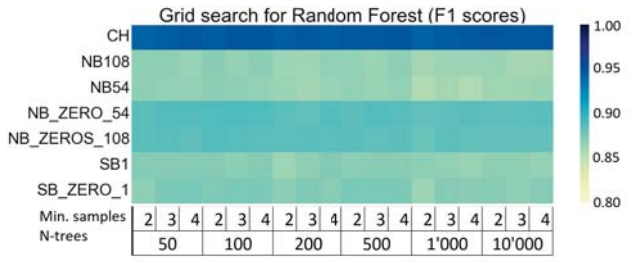

Fig. 1. RF results - vertical lines represent all the data sets tested and horizontal the number of each configuration according to the table III

TABLE III
MULTICLASS TECHNIQUES CONFIGURATIONS

Name	Parameter	Exploration
K-NN	K	2, **3**, 4 ,5 ,6 ,7 ,8 ,9 ,10 ,15 , 20
	Distance	**2**
	Metric	**Minkowski**
RF	N-trees	50, 100, 200, 500, 1 000, **10 000**
	Min samples	**2**, 3, 4
SVM	Kernel	radial basis function
	Penalty	1, 100, **10 000**
	Gamma	**'auto'**, 0.0001
ANN	N-Neurones	2, 5, **10**
	Epochs	50, 100, 300, 500, **1 000**
	Learning rate	**0.1**, 0.01
	Momentum	0.1, 0.5, **0.9**
NBAY	—	No configuration options
LR	Penalty	**12**
	Inv. of Reg.	**1**, 2, 5
	N of iterations	**50**, 100, 200, 300, 500

2) One-class learning: This type of algorithms uses only one labelled class to be trained. They are used in problems where objective is detect outliers [38] or detect samples which are different from a group. The models are trained with only the positive class allowing them to predict if the interactions are positive or different of positive [39]. We trained models based only on the phage-bacteria positive interaction class and treat the negative case as outliers.

We used the techniques described below:

- Replicator Neural Networks (RNN) [40]: is a type of neural network that replicates the input data in order to learn a representation of it. The structure is very similar to an multilayer perceptron (MLP) but using samples from one-class. In this configuration the target is the same input. This structure allows the RNN to learn how to reconstruct the input.
- Local Outlier Factor (LOF) [41]: is very similar to KNN but the difference is that it takes into account the local density deviation of each sample compared to its neighbors. This means that an outlier is detected when it present much lower density deviation than its neighbors (K). Conversely, inliers have much more density as they are thought to have almost similar characteristics.
- Isolation Forest (ISO): is an ensemble of randomly selected decision trees that are only trained with one of the classes trying to isolate (i.e. the outlier samples). The isolation process occurs when the length path or the decision is made very near the root, meaning that the random partitioning of the trees produces shorter paths for anomalies. This early partitioning is due to the high difference between the features of inliers and outliers. Usually, decision trees converges faster when the feature-value is distinguishable from normal values. [42]
- One-Class Support Vector Machines (OC-SVM): is very similar to the regular SVM as it tries to maximize a hyperplane in order to separate or classify the data in an optimal way, but as it only uses data points from one-class

the process of training is different. It generates a boundary from the origin of the data points, and maximize the distance between the boundary (hyperplane) and the origin creating a separation between inlier and outliers. [43]

- Elliptic Envelope (ELL): is based on the minimum covariance determinant (MCD) method which is a robust estimator of multivariate data. It uses the Mahanalobis distance as a discriminant factor for high-dimensional data as well as the minimum volume ellipsoid (MVE). The foundation of this process is fitting the known data into a Gaussian density and then being able to calculate the Mahanalobis distance for each data point in order to optimize the minimum covariance determinant.

To find the optimal hyper-parameters for each technique, we explored several parameters' configurations using grid search (see Table IV).

TABLE IV
ONE-CLASS MACHINE-LEARNING CONFIGURATIONS

Name	Parameter	Exploration
RNN	Layer division	2, 3, **4**, 5
	Threshold	**0.1**, 0.3, 0.5
	Epochs	**50**, 100, 150, 200
LOF	N of neighbors	1, 5, 10, 20, 40, 50, 80, **100**, 150
	Algorithm	ball_tree (BT), kd_tree (KT), **brute (BR)**
ISO	N of estimators	**5**, 10, 20, 40, 50, 80, 100
	Max samples	5, **10**, 50
	max features	0.1, 0.2, 0.4, 0.6, **0.8**
OC-SVM	Kernel	**poly**, rbf
	Degree	1, 2 , **3**, 4
	Nu	0.1, 0.3, **0.5**, 0.7, 0.9
	Gamma	0.1, 0.3, **0.5**, 0.7, 0.9
ELL	Support fraction	0.1, 0.3, 0.5, **0.7**, 0.9
	Center	True, **False**

III. TEST MOLDELS AND SELECTION

1) Multiclass approaches: After analyze the results of the grid search using multiclass techniques we conclude that:

- Independently of the algorithm and configuration used, we obtain the best performances using the CH dataset (where features come from amino acid frequency, the chemical composition, and the molecular weight of proteins).
- We obtained the lowest performances using Naïve Bayes and logistic regression.
- Ramdon Forrest obtained the best results among the other techniques.

Based on these results, we decided to test the models on the "CH test dataset" (i.e. 30% of the original CH dataset put aside to test). The results of these tests are shown in Table V.

In the case of ensemble-learning we use Bagging with hard voting [44]. We tested all possible combinations of the aforementioned techniques arranged into odd groups. So, we tested 6 ensemble-learners composed of 5 techniques, and 20 ensemble-learners formed of 3 techniques. For the creation

TABLE V
MULTICLASS PERFORMANCES

Method	Acc	F1	Precision	Recall
Bagging (RF, ANN, KNN)	**95.7**	**95.9**	**93.6**	98.4
Bagging (RF, SVM, KNN)	95.5	96.2	93.7	**98.7**
RF	95.5	95.7	93.2	98.4
ANN	92.3	95.7	90.1	95.5
KNN	94.3	94.6	91.6	97.9
SVM	94.9	95.2	93.4	97.0
NBAY	75.5	78.8	71.0	88.6
LG	76.7	78.6	74.5	83.2

of the ensemble-learners we used for each technique the best parameters found on the grid search. For the training and validation of the ensemble learners we used the "CH dataset-training" and to test the "CH dataset-test". The only performances surpassing the result of the best single technique (i.e. Random Forest) where found when using SVM, ANN, K-NN, and RF. Random Forrest surpassed the performance of the ensemble learners using 5 techniques.

Once the configuration chosen, we combined the four models in groups of three and to observe if the bagging systems could overpass single methods. The table V shows the performance obtained for each model and the two best hard vote systems. We can observe that Bagging has a highest score in three of the performance measurements.

After analyze the results we can see the high influence of RF in the Bagging models. The two methods that accompany RF seem to help classify some few samples more, increasing in some tenths ACC, F1 and precision.

2) One-class approaches: After performing grid search with the configurations listed in Table IV we analyzed the results and drawing the following conclusions:

- Contrary to multiclass methods where the best performances for all the techniques where obtaining using the CH dataset, by using one-class approaches several datasets shown good performances with diverse techniques.
- With LOF the datasets with the best performances were NB108, SB1, NB54 (interestingly all of them non zero datasets) with a maximum F-score of 67.83% using the dataset NB108 and the parameters highlighted in Table IV).
- By using the technique OC-SVM we obtained the best performance using the dataset NB_ZERO_54. The configuration used is presented in Table IV). We obtained a F1-score obtained of 71.40%.
- With ISO all datasets had at least one configuration with a good performance. Nevertheless, the higher F-score is 66.73% (obtained with the CH dataset).
- The higher scores though all the datasets and techniques were obtained with the CH dataset, namely using ELL and RNN with an F-score of 83.40% and 81.41% respectively. The configurations used to obtain these results are highlighted in bold on Table IV).

TABLE VI
ONE-CLASS PERFORMANCES

Method	Acc	F1	Precision	Recall	dataset
RNN	76.5	72.7	87.1	62.4	CH
ELL	75.1	77.7	70.3	86.7	CH
ISO	50.8	64.0	50.5	87.4	CH
OC-SVM	60.5	53.4	65.0	45.4	NB_ZERO_54
LOF	50.0	64.3	50.0	90.0	NB108

To test each technique we used the best hyper-parameters found in the previous step and the datasets with the highest performance (i.e. the 30% remaining for test). The results are shown in Table VI. The most balanced performances through the different measurements are obtained with RNN and ELL. ISO and LOF show the highest recall indicating a good detection of true positives. Nevertheless, precision and accuracy are low indicating a low detection of true negatives.

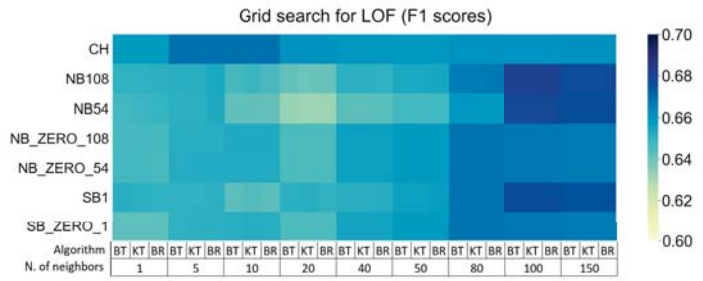

Fig. 2. RNN results - vertical lines represent all the datasets tested and horizontal the number of each configuration according to the table IV

IV. CONCLUSION AND FUTURE WORK

The interaction phage-bacteria is a complex process that involve many variables and biological mechanisms. In this work we do not pretend to explain the biological processes behind phage-bacteria interaction but give some clues about how phage-bacteria interactions can be predicted at strain level using machine learning and which features can lead this path. The approach presented herein might be extended to some more-general contexts as for example: (1) Predicting interaction from pure genomic information: host-pathogen interactions, metagenomics analysis, PPI prediction. (2) Using one-class learning in other biological/clinical data and contexts where positive or negative results are usually over-represented

The novelty of this paper is the application of machine-learning techniques to predict phage-bacterium interactions based solely on genomic/proteomic sequence information at strain level. In this work we presented multiclass and one-class approaches to predict the interaction phage-bacteria at strain level. The best score using a multiclass approach was a F1-score of 95.9%. using a ensemble learner (bagging with hard voting) composed by random forest, artificial neural networks and Nearest-Neighbors models.

The best score using one-class learning was a F1-score of 77.7% with the Elliptic Envelope technique.

Even if multiclass and one-class are different approaches, we obtained in both cases the best performances by using the data sets named Chemical composition (CH). This fact give us a direction to explore new features based on primary structure sequences.

In reference to the new data gathered, if we compare with our previous work [23], we increased our collection of interactions with 20 586 new positive interactions, of which 2 297 are at strain level. This new data allowed us to train models at a lower taxonomic level (i.e. we has passed from species to strain). We completed the data from DOMINE (which last version dated from 2007) with 11 338 pairs of DDI, 5 707 from the last versions of iPfam (2017, release 31.0) and 5 631 from 3did (2017, release 30.0).

As future work several ideas that could improve our results are presented as follows.

We used in this paper F1-score as performance metric to select the best hyperpameters after gird search. New ways to select the best models can be explored using several performance metrics. One option is use the concept of Pareto efficiency.

A modeling approach to try in the future will be use homogeneous representation of interactions at strain level per specie. In this paper we used a dataset with over representation of interactions for the family Mycobacteriaceae.

Although the voting approaches only increase the performances for a few decimals, we think that this shows that voting remain a good methodology to continue testing for instance with one-class methods. It will be interesting to try others methods as boosting or Bayesian model combination. Deep learning techniques are considered for future exploration, actually we can not be used it due to the few amount of available data in our database. We are working on increasing our database in order to apply this type of approach.

Explaining and interpreting the models (e.g. ranking of the most important variables or rules governing the models) remains an interesting subject to explore in order to give biological answers.

REFERENCES

[1] C. L. Ventola, "The antibiotic resistance crisis: part 1: causes and threats," *P T*, vol. 40, no. 4, pp. 277–283, Apr 2015.

[2] A. F. Read and R. J. Woods, "Antibiotic resistance management," *Evolution, Medicine, and Public Health*, vol. 2014, no. 1, pp. 147–147, oct 2014. [Online]. Available: https://doi.org/10.1093/emph/eou024

[3] J. G. Bartlett, D. N. Gilbert, and B. Spellberg, "Seven ways to preserve the miracle of antibiotics," *Clinical Infectious Diseases*, vol. 56, no. 10, pp. 1445–1450, feb 2013. [Online]. Available: https://doi.org/10.1093/cid/cit070

[4] Z. Golkar, O. Bagasra, and D. G. Pace, "Bacteriophage therapy: a potential solution for the antibiotic resistance crisis," *The Journal of Infection in Developing Countries*, vol. 8, no. 02, feb 2014. [Online]. Available: https://doi.org/10.3855/jidc.3573

[5] S. Sharma, S. Chatterjee, S. Datta, R. Prasad, D. Dubey, R. K. Prasad, and M. G. Vairale, "Bacteriophages and its applications: an overview," *Folia Microbiologica*, vol. 62, no. 1, pp. 17–55, oct 2016. [Online]. Available: https://doi.org/10.1007/s12223-016-0471-x

[6] F. L. Nobrega, A. R. Costa, L. D. Kluskens, and J. Azeredo, "Revisiting phage therapy: new applications for old resources," *Trends in Microbiology*, vol. 23, no. 4, pp. 185–191, apr 2015. [Online]. Available: https://doi.org/10.1016/j.tim.2015.01.006

[7] A. Wernicki, A. Nowaczek, and R. Urban-Chmiel, "Bacteriophage therapy to combat bacterial infections in poultry," *Virology Journal*, vol. 14, no. 1, sep 2017. [Online]. Available: https://doi.org/10.1186/s12985-017-0849-7

[8] M. Dalmasso, C. Hill, and R. P. Ross, "Exploiting gut bacteriophages for human health," *Trends in Microbiology*, vol. 22, no. 7, pp. 399–405, jul 2014. [Online]. Available: https://doi.org/10.1016/j.tim.2014.02.010

[9] J. E. Samson, A. H. Magadán, M. Sabri, and S. Moineau, "Revenge of the phages: defeating bacterial defences," *Nature Reviews Microbiology*, vol. 11, no. 10, pp. 675–687, aug 2013. [Online]. Available: https://doi.org/10.1038/nrmicro3096

[10] K. D. Seed, "Battling phages: How bacteria defend against viral attack," *PLOS Pathogens*, vol. 11, no. 6, p. e1004847, jun 2015. [Online]. Available: https://doi.org/10.1371/journal.ppat.1004847

[11] A. C. Greene, "CRISPR-based antibacterials: Transforming bacterial defense into offense," *Trends in Biotechnology*, vol. 36, no. 2, pp. 127–130, feb 2018. [Online]. Available: https://doi.org/10.1016/j.tibtech.2017.10.021

[12] C. O. Flores, J. R. Meyer, S. Valverde, L. Farr, and J. S. Weitz, "Statistical structure of host-phage interactions," *Proceedings of the National Academy of Sciences*, vol. 108, no. 28, pp. E288–E297, jun 2011. [Online]. Available: https://doi.org/10.1073/pnas.1101595108

[13] J. S. Weitz, T. Poisot, J. R. Meyer, C. O. Flores, S. Valverde, M. B. Sullivan, and M. E. Hochberg, "Phage–bacteria infection networks," *Trends in Microbiology*, vol. 21, no. 2, pp. 82–91, feb 2013. [Online]. Available: https://doi.org/10.1016/j.tim.2012.11.003

[14] J. Villarroel, K. Kleinheinz, V. Jurtz, H. Zschach, O. Lund, M. Nielsen, and M. Larsen, "HostPhinder: A phage host prediction tool," *Viruses*, vol. 8, no. 5, p. 116, may 2016. [Online]. Available: https://doi.org/10.3390/v8050116

[15] R. A. Edwards, K. McNair, K. Faust, J. Raes, and B. E. Dutilh, "Computational approaches to predict bacteriophage–host relationships," *FEMS Microbiology Reviews*, vol. 40, no. 2, pp. 258–272, dec 2015. [Online]. Available: https://doi.org/10.1093/femsre/fuv048

[16] E. D. Coelho, J. P. Arrais, S. Matos, C. Pereira, N. Rosa, M. Correia, M. Barros, and J. Oliveira, "Computational prediction of the human-microbial oral interactome," *BMC Systems Biology*, vol. 8, no. 1, p. 24, 2014. [Online]. Available: https://doi.org/10.1186/1752-0509-8-24

[17] NCBI. Genbank - ncbi. [Online]. Available: https://www.ncbi.nlm.nih.gov/genbank/

[18] PhageDB. The actinobacteriophage database at phagesdb. [Online]. Available: http://phagesdb.org/

[19] A. Ross, S. Ward, and P. Hyman, "More is better: Selecting for broad host range bacteriophages," *Frontiers in Microbiology*, vol. 7, sep 2016. [Online]. Available: https://doi.org/10.3389/fmicb.2016.01352

[20] J. Besemer, "GeneMarkS: a self-training method for prediction of gene starts in microbial genomes. implications for finding sequence motifs in regulatory regions," *Nucleic Acids Research*, vol. 29, no. 12, pp. 2607–2618, jun 2001. [Online]. Available: https://doi.org/10.1093/nar/29.12.2607

[21] NCBI. Entrez programming utilities. [Online]. Available: https://www.ncbi.nlm.nih.gov/books/NBK25501/

[22] R. Häuser, S. Blasche, T. Dokland, E. Haggård-Ljungquist, A. von Brunn, M. Salas, S. Casjens, I. Molineux, and P. Uetz, "Bacteriophage protein–protein interactions," in *Advances in Virus Research*. Elsevier, 2012, pp. 219–298. [Online]. Available: https://doi.org/10.1016/b978-0-12-394438-2.00006-2

[23] D. M. C. Leite, X. Brochet, G. Resch, Y.-A. Que, A. Neves, and C. Peña-Reyes, "Computational prediction of host-pathogen interactions through omics data analysis and machine learning," *Bioinformatics and Biomedical Engineering*, vol. 10209, pp. 360–371, apr 2017. [Online]. Available: https://doi.org/10.1007/978-3-319-56154-7_33

[24] Z.-H. You, L. Zhu, C.-H. Zheng, H.-J. Yu, S.-P. Deng, and Z. Ji, "Prediction of protein-protein interactions from amino acid sequences using a novel multi-scale continuous and discontinuous feature set," *BMC Bioinformatics*, vol. 15, no. Suppl 15, p. S9, 2014. [Online]. Available: https://doi.org/10.1186/1471-2105-15-s15-s9

[25] L. Wade and J. Simek, *Organic Chemistry*. Pearson Education, 2016. [Online]. Available: https://books.google.ch/books?id=mbN5CwAAQBAJ

[26] S. Chong, C. Dugast-Darzacq, Z. Liu, P. Dong, G. M. Dailey, C. Cattoglio, A. Heckert, S. Banala, L. Lavis, X. Darzacq, and R. Tjian, "Imaging dynamic and selective low-complexity domain interactions that control gene transcription," *Science*, vol. 361, no. 6400, p. eaar2555, jun 2018. [Online]. Available: https://doi.org/10.1126/science.aar2555

[27] S. Yellaboina, A. Tasneem, D. V. Zaykin, B. Raghavachari, and R. Jothi, "DOMINE: a comprehensive collection of known and predicted domain-domain interactions," *Nucleic Acids Research*, vol. 39, no. suppl_1, pp. D730–D735, nov 2010. [Online]. Available: https://doi.org/10.1093/nar/gkq1229

[28] R. Mosca, A. Céol, A. Stein, R. Olivella, and P. Aloy, "3did: a catalog of domain-based interactions of known three-dimensional structure," *Nucleic Acids Research*, vol. 42, no. D1, pp. D374–D379, sep 2013. [Online]. Available: https://doi.org/10.1093/nar/gkt887

[29] R. D. Finn, B. L. Miller, J. Clements, and A. Bateman, "iPfam: a database of protein family and domain interactions found in the protein data bank," *Nucleic Acids Research*, vol. 42, no. D1, pp. D364–D373, dec 2013. [Online]. Available: https://doi.org/10.1093/nar/gkt1210

[30] S. R. Eddy, "Profile hidden markov models," *Bioinformatics*, vol. 14, no. 9, pp. 755–763, oct 1998. [Online]. Available: https://doi.org/10.1093/bioinformatics/14.9.755

[31] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, and G. Varoquaux, "API design for machine learning software: experiences from the scikit-learn project," in *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, 2013, pp. 108–122.

[32] E. F. Ian H. Witten and M. A. Hall, *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, 2011, implementation of Sklearn. [Online]. Available: https://doi.org/10.1016/c2009-0-19715-5

[33] C.-C. Chang and C.-J. Lin, "LIBSVM," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, pp. 1–27, apr 2011. [Online]. Available: https://doi.org/10.1145/1961189.1961199

[34] A. Cutler, D. R. Cutler, and J. R. Stevens, "Random forests," in *Ensemble Machine Learning*. Springer US, 2012, pp. 157–175. [Online]. Available: https://doi.org/10.1007/978-1-4419-9326-7_5

[35] T. Cover and P. Hart, "Nearest neighbor pattern classification," *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, jan 1967. [Online]. Available: https://doi.org/10.1109/tit.1967.1053964

[36] H. Zhang, "The optimality of naive bayes," *AA*, vol. 1, no. 2, p. 3, 2004. [Online]. Available: http://www.aaai.org/Papers/FLAIRS/2004/Flairs04-097.pdf

[37] H.-F. Yu, F.-L. Huang, and C.-J. Lin, "Dual coordinate descent methods for logistic regression and maximum entropy models," *Machine Learning*, vol. 85, no. 1-2, pp. 41–75, nov 2010. [Online]. Available: https://doi.org/10.1007/s10994-010-5221-8

[38] J. Mourão-Miranda, D. R. Hardoon, T. Hahn, A. F. Marquand, S. C. Williams, J. Shawe-Taylor, and M. Brammer, "Patient classification as an outlier detection problem: An application of the one-class support vector machine," *NeuroImage*, vol. 58, no. 3, pp. 793–804, oct 2011. [Online]. Available: https://doi.org/10.1016/j.neuroimage.2011.06.042

[39] A. M. Bartkowiak, "Anomaly, novelty, one-class classification: A short introduction," in *2010 International Conference on Computer Information Systems and Industrial Management Applications (CISIM)*. IEEE, oct 2010. [Online]. Available: https://doi.org/10.1109/cisim.2010.5643699

[40] R. Hecht-Nielsen, "Replicator neural networks for universal optimal source coding," *Science*, vol. 269, no. 5232, pp. 1860–1863, sep 1995. [Online]. Available: https://doi.org/10.1126/science.269.5232.1860

[41] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "LOF," in *Proceedings of the 2000 ACM SIGMOD international conference on Management of data - SIGMOD '00*. ACM Press, 2000. [Online]. Available: https://doi.org/10.1145/342009.335388

[42] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *2008 Eighth IEEE International Conference on Data Mining*. IEEE, dec 2008. [Online]. Available: https://doi.org/10.1109/icdm.2008.17

[43] P. J. Rousseeuw and K. V. Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212–223, aug 1999. [Online]. Available: https://doi.org/10.1080/00401706.1999.10485670

[44] M. Kirk, *Thoughtful Machine Learning with Python: A Test-Driven Approach*. O'Reilly Media, 2017. [Online]. Available: https://www.amazon.com/Thoughtful-Machine-Learning-Python-Test-Driven/dp/1491924136?SubscriptionId=AKIAIOBINVZYXZQZ2U3A\&tag=chimbori05-20\&linkCode=xm2\&camp=2025\&creative=165953\&creativeASIN=1491924136