The GenPPI tool enhanced Protein Interaction Network Generation with Machine Learning-Based Protein Similarity Inference

Alisson Silva, Carlos Marquez, Iury Godoy, Lucas Silva, Matheus Prado, Murilo Beppler, Natanael Avila, and Anderson Santos E-mail: santosardr@ufu.br

***Abstract*—The study of protein interaction is still promising for a better understanding of diseases. At first glance, the term interaction among proteins could allude to, for instance, the interaction between the protein of a host and a pathogen. Such an interaction is being pursued mainly via machine learning algorithms since it is difficult to discriminate direct rules for it. However, the interaction among proteins on the same genome is also critical, for instance, to understand how a pathogen survives, starts, or maintains an infection. We can analyze interactions within a genome deterministically at the price of significant hardware employment. Our software GenPPI, in its first edition, allows us to explore interaction networks in genomes using mainly the known rules for neighborhood and phylogenetic profiles conserved. However, despite the speed, it suffered from underrepresentation from the core pangenome due to a simplistic algorithm to raise that, losing a pair of proteins possessing less than 90% of amino acid identity. The present work describes the new GenPPI software enhancements on determining homology between protein pairs, which is one of the principal bottlenecks in creating ab initio interaction networks from genomes and the primary step inferring neighborhood and phylogenetic profiles conserved for all genomes under analysis. This improvement was achieved using the Random Forest algorithm, working on biophysical features derived from ten amino acid propensity indexes used to calculate sixty features for each genome's proteins. We crafted a training data set of homolog and non-homolog proteins using nine full proteomes from critical bacteria. A significant number of expressive genomes as the training dataset allowed us to classify similar proteins with more than 65% amino acid identity via a machine learning test, an average result obtained from dozens of validations. Such a strategy resulted in more comprehensive and accurate protein interaction networks capable of analyzing genomes of different organisms. Our testing of the new GenPPI improvement using the bacterium Buchenera aphidicola yielded impressive results. We achieved an overlap of 62% with the interactions documented in the STRING, surpassing the previous GenPPI version, which was limited to less than 50% compared to STRING. More notably, we were able to achieve a full overlap using alternative GenPPI parameters, albeit at the cost of interactions absent on the STRING database. This significant achievement underscores the software's potential as a flexible tool for advancing research in various areas of biomedicine and other scientific fields, balancing precision, completeness, and a lower density of interaction networks. GenPPI is available for access at https://genppi.facom.ufu.br/**

***Index Terms*—Protein, Interaction, Networks, Random Forest, alignment-free**

A. Silva, C. Marquez, I. Godoy, L. Silva, M. Prado, M. Bepler, N. Avila, A. Santos are with Federal University of Uberlândia, Brazil

I. Introduction

The advancement of computational tools has played a revolutionary role in several areas of knowledge in recent decades. The relevance of the use of these tools in the studies of living systems is remarkable, as evidenced by the significant increase in biological research that uses computational strategies [1]. Computational biology has enabled genomic and regulatory analyses in short periods, modeling of proteins not yet observed experimentally, simulations of biological environments, prediction of crucial metabolic pathways, and discovery of new drugs and their mechanisms of action [2].

Protein-protein interactions represent a crucial step toward understanding biological systems. Examples of these interactions include enzyme-substrate interactions, regulation of enzyme activity, molecule transport, structure formation, and immune response. Protein interaction networks enable inferences about important biological processes that need to be better understood or easily studied experimentally. The determination of these networks can occur through experimental or computational approaches. Experimental methods include proteomic techniques, tandem affinity purification, yeast double-hybrid system, and protein and DNA assays [3].

Although experimental methods are valuable, they have drawbacks, such as high cost and extensive time demands [4]. The use of computational tools is desirable to mitigate these costs, explore networks of interactions that are not yet fully understood, and elucidate biological processes. In computational prediction, existing biological information is used to build interaction networks. Among the methods used are the analysis of amino acid sequence, physicochemical properties, three-dimensional structure, phylogenetic profile, conserved gene neighborhood, and gene expression [3].

Algorithms designed for the prediction of protein-protein interactions can make use of gene ontology [4] or combine it with other strategies [5]. However, it has been shown that more than gene ontology is needed to produce satisfactory results in the prediction process [6]. We did an extensive review of the literature considering recent years, and there is no state-of-the-art software that is able to predict interaction networks using exclusively genomic data efficiently.

Among the current approaches to the prediction of protein interaction networks, STRING leaves something to be desired, especially when it comes to complete, unpublished, little-known, or studied genomes. In this context, version 1.0 of

GenPPI presented a methodology that employs biochemical characteristics of the genome and comparative phylogenetic and genomic analyses, such as phylogenetic profile and conserved gene neighborhood. This approach allowed the prediction of interaction networks from any genome, not necessarily depending on data available in external databases, which may not cover the specific genome of interest to a researcher.

GenPPI is a software designed as a generator of protein interaction networks *ab initio*. The central proposal of GenPPI is to create networks of interactions from genomes based on three main characteristics: conserved gene neighborhoods, phylogenetic profiles, and gene fusions. The program was developed to analyze genomes represented in the form of multiple files of protein sequences in order to identify conserved neighborhoods, phylogenetic profiles, and gene fusions.

A relevant aspect is its ability to transfer to the end user the decision about how many and which genomes to use in the construction of the protein interaction network. All the code was written in Common Lisp and compiled using SBCL, allowing the creation of complex protein networks predicted from tens or hundreds of genomes. GenPPI stands out for its versatility, being designed to be compatible with a variety of operating systems and hardware configurations. Several versions of the program have been optimized to meet different needs and usage scenarios.

It is easy to access and use, requiring no specific expertise in computing or biology. The software is designed to facilitate the discovery of novel relationships between proteins and their associated biological processes and is available for download at the following address: https://genppi.facom.ufu.br/

A. State of the Art

In the context of predictions of protein interactions, the STRING (Search Tool for the Retrieval of Interacting Genes/Proteins) stands out as one of the main state-of-the-art tools. STRING is a database that brings together, in an integrated way, known and predicted protein interactions, covering multiple sources of evidence, such as biochemical experiments, genetic associations, and gene coexpression, among others. Its applicability extends to more than 2000 organisms, allowing comparative analyses between different species. The platform offers an intuitive interface for the visualization of interaction networks, facilitating the identification of functional modules and the interpretation of complex biological contexts. The constant updating and inclusion of new functionalities make STRING a valuable tool for researchers in the field of bioinformatics and systems biology, contributing significantly to the advancement of knowledge about protein interactions and their implications in health and disease [7].

There are other tools available for the prediction of protein-protein interaction networks, but not all of them are applicable or comparable to GenPPI in all contexts. For example, some of these tools are restricted to specific organisms, do not allow data to be uploaded for analysis, or are limited to data from multicellular organisms (eukaryotes). These limitations make it impractical to directly compare these tools with GenPPI, especially when it comes to unpublished, little-known, or studied complete genomes. GenPPI, on the other hand, allows the prediction of interaction networks from any genome of bacteria or archaea. The flexibility in creating interaction networks distinguishes GenPPI from other methodologies and makes it particularly useful for researchers working with genomes that other tools may not cover. Therefore, while it is helpful to be aware of the other tools and methodologies available, it is crucial to understand the limitations that prevent a direct comparison with GenPPI.

For example, in [5], which uses information from known complexes, gene expression profiles, Gene Ontology terms, and subcellular localization information for the discovery of essential proteins, is a theoretical algorithm and does not have an associated software for direct implementation in the code of the GenPPI.

Regarding [8], which proposes a new method to predict unknown protein-protein interactions, it is important to clarify that, despite its theoretically interesting proposal, the associated software presented functionality problems. Attempts to execute the code made available by the authors were unsuccessful, and our requests for assistance still need to be answered. Therefore, it was not possible to compare our results with theirs.

In addition, during the development of GenPPI, it was demonstrated that the interaction network accuracy produced using Gene Ontology [4] was inferior to an *ab initio* generation [6].

The topological analysis of protein-protein interaction networks (PPI) is a relevant aspect of bioinformatics and systems biology, serving as one of the main motivations for the development of version 1.5 of GenPPI. A study entitled *Topological scoring of protein interaction networks* [9] reinforces the need for tools such as GenPPI, which facilitate topological analysis in PPI networks, contributing to a better understanding of protein interactions and their functional implications.

II. Materials and Methods

A. Features

The source code named *Features.lisp* is a recent addition to GenPPI in version 1.5, being an in-house program that uses a file called *propensity.dat* as a basis for mapping specific features of proteins. This file contains specific values for each amino acid, allowing the program to create a propensity index for each characteristic of each amino acid present in all proteins in a multi-fast file. The data in the *propensity.dat* file represent different physicochemical characteristics of the amino acids, such as acidity/basicity (BASIC, ACID), polarity (POLAR, NONPOLAR), mass (MASS, MASSMR) and other specific indices (PARJ860101, JOND750101, EISD840101, JURD980101), in a total of ten characteristics available in the database of the AAindex website.

Each row of the *propensity.dat* file represents a characteristic, and each column is an amino acid. The numerical values represent the propensity of each amino acid for this characteristic. These indices were selected based on their effectiveness in classifying proteins according to the most common subcellular sites in prokaryotic organisms, such as the membrane, cytoplasm, cell wall surface, and secreted proteins.

The program reads the FASTA file, which contains several amino acid sequences, and the propensity file, which contains information about the physicochemical properties of amino acids. For each sequence in the FASTA file, the *Features* program calculates a histogram that counts the occurrence of each amino acid in the sequence. In addition, it calculates the weighted sum of the physicochemical properties for the entire sequence and the start, end, and middle regions of a sequence.

For each trait, the *Features* program calculates the value along the entire protein. Also, it segments the protein into the beginning, middle, and end, resulting in four values for each trait. This results in a subtotal of forty values per protein. In addition, simply counting the amount of each amino acid adds twenty more numerical characteristics to each protein, totaling sixty values that define the propensity index of a protein. A greater number of discriminative features allows for a more granular analysis of protein properties, as different regions of a protein can have different amino acid compositions and, therefore, different physicochemical properties.

B. Random Forest

The insertion of Random Forest as a machine learning algorithm in GenPPI emerges as a solution to improve the ability to predict protein-protein interaction networks (PPI) in bacterial genomes, a limitation faced by GenPPI 1.0 that caused a non-optimal sampling of similar proteins. The challenge faced was to identify interactions between proteins with levels below 90%. Exact alignment or heuristic algorithms, such as blast, become computationally costly, especially in complex pangenomes, which is why we have sought alternative ways to identify protein similarity.

The choice of the Random Forest algorithm as a solution to the problem is justified by its ability to efficiently handle complex and high-dimensional datasets, as is the case with the genomic and proteomic information involved in the GenPPI. Random Forest builds a collection of decision trees, each of which is trained on a random subset of the available data and variables. This approach promotes diversity among trees, preventing dominant traits from exerting an excessive influence on the model, which contributes to better generalization. Each tree forms a structure similar to a flowchart, contributing to joint decision-making [10].

The choice of the *CL-random-forest* library for incorporation into GenPPI was based on the need for robust and efficient implementation of Random Forest in Common Lisp [11]. Cl-random-forest is an implementation of the Random Forest algorithm for SBCL. This library supports multiclass classification and univariate regression and is notable for its efficiency and accuracy, outperforming other popular implementations such as scikit-learn (Python/Cython) and ranger (R/C++). In addition, *CL-random-forest* includes a global refinement of Random Forest, as described in [12], making it even faster and more accurate than the standard implementation.

The code used to train the Random Forest model used in GenPPI 1.5. is available in the GitHub repository https://github.com/santosardr/non-CSPs/tree/main/src/lisp. This file is dependent on the *CL-random-forest* and *CL-store* libraries, which are installed via Quicklisp. Different training and testing sets were used with this interface to generate the final model for predicting similarities between protein pairs used in this new version of GenPPI.

C. Database

We compared the results obtained by the GenPPI software with the STRING data. STRING is recognized as a gold standard in protein interaction analysis, providing information on known interactions and predictions of interactions between proteins. For this purpose, the PostgreSQL relational database management system was used.

The database used for this analysis is available in the GenPPI repository on GitHub, under the name *compareba.dump*, since the study organism for comparison is *Buchnera aphidicola*. This database was specifically designed for genome-related analyses *Ba_ak*, used as an example in this study. It consists of two main tables: *protein* e *related*. In the *protein* table, the identifiers of the proteins and from which genome they originate are stored, obtained from both the STRING interaction network and the GenPPI interaction network. The *related* table stores the interactions between proteins of the same genome, allowing the comparison of interactions between proteins.

For an accurate analysis of the interactions between proteins, the *count_intersect_distinct* function was developed. This function returns a table with the name of the genome and the number of unique interactions performed, considering the interactions with and without repetitions. In the context of STRING, where the interaction between A and B is considered different from the interaction between B and A, the count of interactions without repetitions is equal to half of the total interaction count. On the other hand, in the GenPPI, repeated interactions are removed through the GEPHI program, commonly used in this work to visualize the interaction networks generated by the GenPPI [13].

D. Experiments

1) Genomes Training and Testing: We turned to the National Center for Biotechnology Information (NCBI) due to its vast database and the availability of genomes of pathogenic bacterial species that have been widely studied. We used the genomes of several species, including *Aeromonas hydrophila, Bacillus anthracis, Bacteroides fragilis, Borrelia burgdorferi, Clostridium botulinum, Clostridium tetani, Corynebacterium Jeikeium, Enterobacter huaxiensis, Enterococcus faecalis, Escherichia coli, Francisella tularensis, Helicobacter pylori, Moraxella Catarrhalis, Morganella morganii, Mycobacterium avium, Mycobacterium tuberculosis, Providencia rettgeri, Pseudomonas aeruginosa, Salmonella enterica, Staphylococcus aureus, Streptococcus pyogenes, Vibrio cholerae e Yersinia pestis*. The selection of these species is based on their relevance as pathogenic agents and on their extensive study in the scientific literature.

Data from model organisms, such as those found in the NCBI, can provide a solid basis for future investigations, allowing the confirmation of hypotheses and conclusions

reached in our work. In addition, the availability of these genomes facilitates the creation of representative datasets and the performance of comparative analyses, contributing to the robustness and validity of the results obtained in our experiments.

2) Generating the Machine Learning Model: In the creation of the training file, proteins from nine medically important genomes were selected: *Bacillus anthracis, Clostridium tetani, Enterococcus faecalis, Escherichia coli, Francisella tularensis, Mycobacterium tuberculosis, Salmonella enterica, Staphylococcus aureus e Vibrio cholerae*. The *makearff* script performs a series of manipulations on text files containing organic information, applying filters and converting them to the ARFF format, which is essential for training the algorithm. We used *makearff* to compare the nine genomes with each other. This process generated a file with proteins identified with their characteristics and the class to which they belong (positive or negative).

The original dataset showed a significant imbalance between positive (8,858) and negative (318,309) proteins, which made it difficult to train classification models. The initial application of the Random Forest algorithm did not obtain satisfactory results due to this discrepancy. A series of tests were carried out to determine the optimal ratio between positive and negative proteins. Several strategies were tested, such as random selection, in order of start, middle, or end of the data set and combinations between them. Based on analyses and experiments, the ratio of 2.6:1 (positive:negative) was identified as the most adequate.

The program call is performed with the following command: *./randomforest similar_test.arff xxxx similar_training.arff*, where the first parameter is the name of the program, the second parameter *(similar_test.arff)* is the test file containing the genome proteins to be evaluated, the third parameter *(xxxx)* indicates the number of negative proteins present in the test file, in digit format. It is important to note that the ARFF file is carefully structured in such a way that all negative proteins precede positive ones. This ordering is critical, as true labels for each data instance are used in training. The Random Forest program, implemented in Lisp, uses this information to calculate the accuracy of the model under construction, accounting for successes and errors. The last parameter represents the training file. Similar to the test file, the training file also follows the pattern of negative instances preceding positive ones. This compliance allows training and test files to be interchangeable, providing flexibility in machine learning model evaluation and validation.

3) Using the GenPPI-trained model: The GenPPI initially employed the simple counting of the occurrences of the twenty-six amino acids (including six non-standardized) as a method of analysis. However, in the enhanced version, incorporated by the sixty features generated by the *Features* program, in the default mode of execution, called *Features* mode, two proteins are considered similar if they present at least twenty-five of these characteristics with a maximum difference of one or two units. This last number is a value obtained empirically. Changing this number affects the ability to generate inferences of correct similarities. If this number were increased too much, for example, close to the sixty features that GenPPI 1.5 uses, GenPPI 1.5 in *Features* mode would pass like GenPPI 1.0 when it used twenty-six amino acid possibilities.

The same sixty features used by the *Features* algorithm are also employed in the *CL-random-forest* program, which performs machine learning to create more complex networks of proteins. The GenPPI machine learning mode is triggered through the *-ml* parameter, which is optional. To use the machine learning mode, the user must download the Random Forest model previously trained to recognize similar protein pairs with more than 65% identity and size alignments as determined by BLASTP.

In the new version of GenPPI 1.5, some parameters have been standardized to simplify the user experience. The *-ppcomplete* parameter, which was used to perform PPI predictions by the conserved phylogenetic profiling method without applying filters, is now enabled by default; the program will make the predictions without reducing the number of predicted interactions.

In addition, the type of neighborhood expansion of a recurrent gene, which was controlled by the *-expt* parameter, is now *fixed*. The size of the main window *(-w1)* and the required amount of genes conserved within window 1 *(-cw1)*, which were set to ten and four, respectively, are now also default values. Therefore, in the absence of extra parameters, GenPPI 1.5 will run with these default settings, simplifying the process and allowing users to focus more on interpreting the generated results.

Once one clones our example repository, it is possible to perform tests on the *Buchnera Aphidicola* genome. The program can be run with the following command (in the Linux operating system): *./genppi -dir test1*. The first parameter represents the version of the GenPPI program selected for the machine, with multiple builds available for different memory usage limits. The second parameter *('-dir')* indicates the name of the directory where the genomes are stored in the form of multifasta files (one file per genome), and the last parameter *('test1')* is the name of the directory in question.

The program's machine learning mode (in Linux) is fired, for instance, with the command: *./genppi -dir test1 -ml*. The parameters are the same, with the exception of the last one *('-ml')*, which instructs the program to use machine learning to identify similar proteins.

E. Split

When employing machine learning, a significant increase in the number of interactions was identified compared to the approach without the use of machine learning. Such an increase represented an additional challenge for the topological analysis of the resulting network. To mitigate this increase, an algorithm called *split* was developed that is incorporated into the GenPPI 1.5 source code. (Algorithm 1).

To understand the need to use the split algorithm, let us exemplify a common situation in the generation of interaction networks. Assuming that a phylogenetic profile of five conserved genes was found in several genomes, all genomes in

which this profile was found will have edges created between all five homologs of this genome, i.e., six edges created.

$$f(x) = \frac{(x-1)(x-2)}{2} = 6 \text{ edges} \quad (1)$$

Algorithm 1 Pseudocode for the split Algorithm

```
REPLACE-ITEM(target, item, replacements)
   pos ← position of item in target
if pos then
      target ← concatenation of (subsequence of target up to

      pos), replacements, (subsequence of target from pos+1)
end if

SUBSTITUTE-NODE(target, number, node)
   header, replacements ← empty
if target not empty, node not empty and number > 0 then
   if number < length of target then
       header ← 5 element of the number-th element of target

      for each subnode in node do
Add (header, subnode, length of subnode) to the replacements list
      end for
   else
replace-item(target, number-th element of target, replacements)
   end if
end if

SHUFFLE-LIST(target)
   copy ← copy of target
for i from length of copy down to 2 do
Swap (random(i)th element of copy) with ((i-1)th element of copy)
end for
   return copy

SPLIT-LIST(protein-list, dividend, redundants)
   protein-list-shuffled ← shuffle-list(protein-list)
   n ← length of protein-list-shuffled
   sublist-size, sublists, pivotal-sublist, non-pivotal-sublists ←
empty
if dividend < n and redundants < n then
      sublist-size ← round(n / dividend)
      sublists ← subdivision of protein-list-shuffled into
                 sublists of size sublist-size
      dividend ← length of sublists
end if
for each pivotal index from 0 to dividend-1 do
      pivotal-sublist ← pivotal-th element of sublists
      non-pivotal-sublists ← sublists without pivotal-sublist
   for each redundant do
      if there are elements to choose from non-pivotal-sublists
      then
         source sublist index                          ←
         random index of non-pivotal-sublists
         source sublist ← source sublist index
         element index ← random index of source sublist
         element ← element index
         if the element is not in the pivotal-sublist then
               add element to pivotal-sublist
               remove element from source sublist
         end if
      end if
   end for
    update the pivotal-th element of sublists to pivotal-sublist
end for
   return sublists
```

When we analyze two or more genomes whose organisms are evolutionarily close, the tendency is for the number of genes in conserved phylogenetic profiles to increase significantly. For example, it is very common to have phylogenetic profiles conserved with more than five hundred genes in organisms with more than two thousand genes. Therefore, the number of edges that would initially be created is 124,251 edges, with each gene having 499 edges connected to the others. For another phylogenetic profile with 700 conserved genes, we would have 243,951 edges, with each gene having 699 edges. With few large phylogenetic profiles conserved in a genome, it is very common that at the end of the generation of the interaction network, we have edges that exceed half a million or even a million. The final amount of edges is proportional to the level of similarity between the genomes.

To try to slow down this rapid growth in the number of edges and average degrees of the vertices of the graph, we thought of a process that could make a meaningful sampling of the interactions existing in phylogenetic profiles with several hundred conserved genes. The idea of sampling is to create as many interactions as possible between the genes of a phylogenetic profile with many genes without necessarily including all the edges. In this way, we were able to show that gene *A* interacts with gene *B* without necessarily including all the interactions of genes *A* and *B*. We call this sampling approach split, alluding to how the approach works by separating the large set of genes of a profile into many smaller portions. The split generates a number of smaller profiles equal to 90% of the size of the phylogenetic profile but places in each derived profile a smaller number of randomly selected genes.

Initially, the *split* promotes a disorder of the list of similar proteins known as *shuffle*. Later, this list is subdivided into a number of sublists equal to 90% of the number of elements in the original list. Each sublist is randomly assigned about half minus one element of the undivided unordered list. The program is then run separately on each of these sub-lists, resulting in a reduction in the number of possible combinations and, consequently, in the total number of edges in the network. This process of splitting and executing separately into subgroups allows for an effective reduction in the number of interactions, facilitating topological analysis of the resulting network.

A relevant question that arises when considering this strategy is the fate of proteins that do not form edges. Although some interactions are not represented, the tests carried out showed that after the application of the *split*, the result of the network of interactions practically did not change in terms of analyses of the topological importance of nodes, that is, in addition to reducing the number of edges in the network, which facilitates the analysis of the network of protein-protein interactions (PPI), The quality of the underlying data was not compromised, maintaining node differentiation in centrality measures. The explanation lies in the fact that the decrease in the degree of the nodes was done homogeneously due to randomness, preserving the ranking of the most connected nodes. This approach provides an effective solution to address the increased number of interactions resulting from the use of machine learning while maintaining the quality and integrity of the data for subsequent analysis by the PPI network.

III. Results

A. Comparative Analysis of the Results of the Classification Model and Performance Evaluation of the Random Forest Algorithm in Different Genomes

In the table I are presented all the genomes or pairs of genomes used to evaluate the results of the classification model elaborated with the *CL-random-forest* algorithm. Using the BLASTP, sets of results considered similar (positive) or non-similar (negative) were selected based on criteria of protein identity and size of alignments greater than 65%, a value empirically selected as a cut-off to define homology. These datasets constituted test ARFF files for the classifier implemented with the Random Forest algorithm.

TABLE I
Genomes employed and classification results of the Random Forest

Genomes	True Negative	False Negative	True Positive	False Positive	Total's Positive	Total's Negative	Total
Aeromonas Hydrophila	7279	1086	65	23	88	8365	8453
Aeromonas Hydrophila - Bacteroides Fragilis	19379	6103	206	62	268	25482	25750
Aeromonas Hydrophila - Clostridium Botulinum	26387	3631	163	58	221	30018	30239
Aeromonas Hydrophila - Enterobacter Huaxiensis	34136	3631	163	58	944	40248	41192
Bacteroides Fragilis	5988	3417	128	35	163	9405	9568
Bacteroides Fragilis - Clostridium Botulinum	20065	6101	212	58	270	26166	26436
Clostridium Botulinum	6403	1987	79	21	100	8390	8490
Enterobacter Huaxiensis	10736	2160	68	9	77	12896	12973
Enterobacter Huaxiensis - Clostridium Botulinum	24692	6336	213	47	260	31028	31288
Enterobacter Huaxiensis - Bacteroides Fragilis	29796	7908	170	31	201	37704	37905
Morganella Morganii	3834	991	125	3	128	4825	4953
Providencia Rettgeri	4652	1394	51	7	58	6046	6104
Yersinia Pestis	6831	879	150	26	176	7710	7886
Streptococcus Pyogenes	1562	156	17	2	19	1718	1737
Mycobacterium Avium	4631	10027	170	7	177	14658	14835
Mycobacterium Avium - Corynebacterium Jeikeium	9577	13181	465	37	502	22758	23260
Mycobacterium Avium - Moraxella Catarrhalis	7605	10552	208	36	244	18157	18401
Corynebacterium Jeikeium	1201	585	45	8	53	1786	1839
Corynebacterium Jeikeium - Moraxella Catarrhalis	3422	1121	83	37	120	4543	4663
Moraxella Catarrhalis	521	92	35	29	64	613	677
Helicobacter Pylori	724	128	9	12	21	852	873
Helicobacter Pylori - Borrelia Burgdorferi	1522	854	400	69	469	2376	2845
Helicobacter Pylori - Pseudomonas Aeruginosa	15727	6307	134	35	169	22034	22203
Pseudomonas Aeruginosa	12646	5346	113	23	136	17992	18128
Pseudomonas Aeruginosa - Borrelia Burgdorferi	14911	6035	503	80	583	20946	21529
Borrelia Burgdorferi	241	594	389	56	445	835	1280
Média	10556	3869	168	33	229	1452	

After applying the *CL-random-forest* algorithm to the collected data, confounding matrices were generated to evaluate the effectiveness of the classification model in relation to each genome, represented by the count of errors and successes in relation to the *negative* and *positive* classes, essential to calculate the evaluation measures of the trained Random Forest algorithm.

Evaluation measures were calculated, including accuracy, sensitivity, and specificity for each genome evaluated. The results of these metrics are shown in Table II. The last row of table II shows the average of these values for all genomes, providing an overall measure of the performance of the classification model. The analysis of the data reveals a mean sensitivity of 0.81, specificity of 0.72, and accuracy of 0.73 for the twenty-six genomes evaluated, indicating great sensitivity and being able to identify a high proportion of true similarities correctly. False positive similarities in greater quantity are not necessarily a problem because GenPPI will still analyze the conserved gene neighborhood traits and phylogenetic profile to define an interaction. It is unlikely that two false positive proteins for similarities will have these types of preservation.

When looking at the individual data, we note that some genomes, such as *Morganella Morganii* and *Mycobacterium Avium*, have exceptionally high sensitivity (0.98 and 0.96,

respectively). However, the specificity and accuracy are relatively lower. Genomes such as *Yersinia Pestis* and *Streptococcus Pyogenes* have high accuracy (0.89 and 0.91, respectively), indicating a balance between sensitivity and specificity. These variations highlight the importance of considering the specific genomic context when interpreting the results of predictions of protein-protein similarities. This systematic approach to evaluation allows a comprehensive understanding of the performance of the classification model in relation to the different genomes analyzed, contributing to a more accurate interpretation of the results obtained.

TABLE II
PERFORMANCE EVALUATION OF THE CLASSIFICATION MODEL ON VARIOUS GENOMES

Genomes	Sensitivity	Specificity	Accuracy
Aeromonas hydrophila	0,74	0,87	0,87
Aeromonas hydrophila - Bacteroides Fragilis	0,77	0,76	0,76
Aeromonas hydrophila - Clostridium Botulinum	0,74	0,88	0,88
Aeromonas hydrophila - Enterobacter Huaxiensis	0,88	0,85	0,85
Bacteroides Fragilis	0,79	0,64	0,64
Bacteroides fragilis - Clostridium Botulinum	0,79	0,77	0,77
Clostridium Botulinum	0,79	0,76	0,76
Enterobacter Huaxiensis	0,88	0,83	0,83
Enterobacter Huaxiensis - Clostridium Botulinum	0,82	0,80	0,80
Enterobacter Huaxiensis - Bacteroides Fragilis	0,85	0,79	0,79
Morganella Morganii	0,98	0,79	0,80
Providencia Rettgeri	0,88	0,77	0,77
Yersinia Pestis	0,85	0,89	0,89
Streptococcus pyogenes	0,89	0,91	0,91
Mycobacterium Avium	0,96	0,32	0,32
Mycobacterium Avium - Corynebacterium Jeikeium	0,93	0,42	0,43
Mycobacterium Avium - Moraxella Catarrhalis	0,85	0,42	0,42
Corynebacterium Jeikeium	0,85	0,67	0,68
Corynebacterium Jeikeium - Moraxella Catarrhalis	0,69	0,75	0,75
Moraxella Catarrhalis	0,55	0,85	0,82
Helicobacter Pylori	0,43	0,85	0,84
Helicobacter Pylori - Borrelia Burgdorferi	0,85	0,64	0,68
Helicobacter Pylori - Pseudomonas Aeruginosa	0,79	0,71	0,71
Pseudomonas Aeruginosa	0,83	0,70	0,70
Pseudomonas Aeruginosa - Borrelia Burgdorferi	0,86	0,71	0,72
Borrelia Burgdorferi	0,87	0,29	0,49
Média	0,81	0,72	0,73

B. Comparison of Results: GenPPI v1.0 vs. Use of Features and Machine Learning Features

The analysis of the results obtained by GenPPI in version 1.0 compared to the incorporation of the *Features* and *Machine Learning* functionalities in GenPPI 1.5 reveals a significant improvement in the prediction of protein interaction networks. We observed a consistent trend of increasing the number of nodes and edges, indicating a greater complexity and scope of the networks generated after implementations.

Initially, in version 1.0 of the GenPPI, the genomes of *Buchnera aphidicola* resulted in a relatively low number of nodes and edges, as evidenced in the data in table III. However, after the introduction of the *Features* and *Machine Learning* features, there was a notable increase in both the number of nodes and edges in all the analyzed samples. This expansion of protein interaction networks suggests a greater ability of the improved software to identify and predict relationships between proteins, providing a more comprehensive and detailed view of the underlying biological processes.

The results also indicate an upward trend in the complexity of the networks generated, reflected in the significant increase in the total number of nodes and edges. In summary, the comparative results between the different versions of GenPPI demonstrate the benefits of using additional resources in predicting protein interaction networks.

TABLE III
COMPARISON OF GENPPI PERFORMANCE WITH FEATURES AND MACHINE LEARNING ON DIFFERENT DATASETS

File	Number of Proteins by Genomes	GenPPI 1		GenPPI 1.5			
				Features		Machine Learning	
		Nodes	*Edges*	*Nodes*	*Edges*	*Nodes*	*Edges*
Ba_Bp.dot	553	46	91	217	1746	539	30220
Ba_Ua.dot	591	111	314	450	6115	581	40049
Ba_G002.dot	621	120	395	529	9166	617	41969
Ba_Sg.dot	620	88	203	430	5583	614	41483
Ba_Ak.dot	559	119	339	448	7917	559	44005
Summary	2944	484	1342	2074	30527	2910	197726

C. Comparison with STRING

The data presented below refer to the *Ba_ak* genome of the bacterium *Buchenera aphidicola* and were obtained through a SQL query to the *compareba.dump* database. The SQL file *count_intersect_distinct.sql* contains the definition of a function called *count_intersect*. This function returns a table with information about the intersection of different genomic datasets. Function code involves identifying unique elements and intersections between those elements and calculating metrics such as the proportion of unique intersections and the proportion of interactions between elements. After the function is defined, there is a SELECT statement that calls the *count_intersect* function and returns the results of the query.

The conserved phylogenetic profile infers interactions for pairs of proteins that have identical phylogenetic profiles. It is reasonable to consider that pairs of proteins with similar profiles can also interact. To incorporate this notion of similarity rather than exact identity, the *-ppdifftolerated* parameter in GenPPI establishes the tolerated difference between phylogenetic profiles. The flexibility provided by the *-ppdifftolerated* parameter allows for a more comprehensive analysis of protein-protein interactions, taking into account the similarity of phylogenetic profiles [14].

Table IV presents the results of different implementations of GenPPI, varying parameters such as the number of proteins in a phylogenetic profile and the tolerated difference between the phylogenetic profiles of a pair of proteins. Each row in the table represents a specific implementation. The table features various metrics, including *query_nounique, query_unique, string_intersect_unique, unique_intersect_ratio e string_proportion* for different parameters. For example, the term *tol* is synonymous with the *-ppdifftolerated parameter*. The N number that accompanies the *split* represents the attempts made to determine whether the *split* algorithm would be triggered when a phylogenetic profile had more than N proteins. The standard that has been established (*mlsplit*) is N=100.

In the table IV, we have results for the *Features* and the *Machine Learning* modes. Only the sixty traits described in section **??** were used with a cut-off factor of twenty-five identical features between a pair of proteins, considering them identical for eventual interaction conservation analyses. The term *ml* is synonymous with the *Machine Learning* mode of GenPPI 1.5. The *ml* mode is triggered when the *Features* mode concludes that, based on the lack of at least twenty-five

identical features among a pair of proteins, a pair of proteins is not identical.

TABLE IV
RESULTS OF THE QUERY EXECUTION ON THE COMPAREBA.DUMP DATABASE WITH DIFFERENT IMPLEMENTATIONS AND PARAMETERS.

Genome	query_ nounique	query_ unique	string_ intersect_ unique	unique_ intersect_ ratio	string_ proportion
string	54082	27041	54082	0.50	2.00
Features	7918	7917	2773	2.86	0.10
mlsplit	92137	49077	16681	2.94	0.62
Featurestol1	19247	19244	5380	3.58	0.20
mlsplittol1	150789	103743	26924	3.85	1.00
Featurestol2	33582	33579	8021	4.19	0.30
mlsplittol2	170849	128579	29846	4.31	1.10
Featurestol3	40465	40461	9232	4.38	0.34
mlsplittol3	187547	144468	32278	4.48	1.19
mlsplit25	47854	46634	9509	4.90	0.35
mlsplit10	45430	44216	8844	5.00	0.33
mlsplit35	48196	47103	9335	5.05	0.35
mlsplit35tol1	81182	80055	14573	5.49	0.54
mlsplit35tol3	136538	135617	23840	5.69	0.88
mlsplit35tol2	116101	114998	20034	5.74	0.74

The *query_nounique* and *query_unique* columns represent the number of implementation-related interactions, which are repeated or not. In comparison with STRING, column *string_intersect_unique* indicates the number of unique interactions that match those of STRING, while column *unique_intersect_ratio* represents the ratio of the number of unique interactions in GenPPI to STRING. The *string_proportion* column indicates the proportion of interactions in GenPPI that are also present in STRING.

Note that the values referring to the row corresponding to the STRING are duplicated because the data source in the STRING considers the interaction between A and B different from B and A. In the other data, this duplicity is removed. Therefore, the actual value of *unique_intersect_ratio* and *string_proportion* in the line corresponding to the STRING is 1 and 1, instead of 0.5 and 2, respectively. The reason is that the *count_intersect* function is intended to obtain metrics for implementations made in GenPPI and not for STRING.

Unique_intersect_ratio is a crucial metric for understanding the effectiveness of different GenPPI implementations. This ratio represents the number of predicted interactions divided by the number of interactions that intersect with the STRING. Therefore, for every interaction that matches the STRING, there are a certain number of other interactions that do not belong to the STRING. The existence of interactions absent in STRING can be interpreted as an inverted distance measure of the STRING, where a *unique_intersect_ratio* of 1 would indicate a perfect match with the STRING. The higher the *unique_intersect_ratio* value, the greater the difference between the phylogenetic profiles of a pair of proteins is allowed to classify them as interacting, a situation that may result in more interactions being predicted by GenPPI. However, some of these additional interactions may not be present in STRING. Interactions predicted by GenPPI but in STRING can be useful for discovering potential new interactions, but it can also lead to a higher number of false positives. Therefore, it is important to strike a balance when adjusting the *unique_intersect_ratio* value.

It is important to consider that a high average degree of edges, determined by the value of *unique_intersect_ratio*, can hinder a topological analysis, as it generates a very large number of interactions. Therefore, when evaluating the effectiveness of different GenPPI implementations, it is crucial to consider not only the STRING match but also the structure and topology of the resulting network.

As the value of *tol* increases (from *tol1* to *tol3*), the *unique_intersect_ratio* also increases. Greater values to *tol* suggest that allowing a greater difference between phylogenetic profiles may result in greater agreement with the STRING. Similarly, increasing the number of proteins in a phylogenetic profile (from *mlsplit10* to *mlsplit35*) also appears to increase *unique_intersect_ratio*. However, it is important to note that *unique_intersect_ratio* higher does not necessarily indicate a better prediction of protein-protein interactions. While greater agreement with the STRING may suggest greater accuracy, it is also possible that some valid interactions may be lost. Therefore, it is crucial to consider the balance between sensitivity (the ability to identify interactions correctly) and specificity (the ability to exclude non-interactions correctly) when evaluating the effectiveness of different GenPPI implementations.

In addition, the *string_proportion* represents the result of dividing the number of single-intersect interactions of the specific GenPPI implementation by the number of STRING single-intersecting interactions (i.e., 27041), a measure of the proportion of interactions in GenPPI that are also present in STRING. For the implementation with *Machine Learning* and the use of the *split* algorithm - *mlsplit*, it obtained a percentage of 62%. For *mlsplit35tol3* and *mlsplit35tol2*, a percentage of 88% and 74% respectively. For *mlsplittol1*, *mlsplittol2* and *mlsplittol3*, values higher than 100%, being respectively 100%, 110% and 119%. Values higher than 100% occur in certain cases where the adopted implementation recognizes the STRING values of interactions that are repeated in different scenarios.

Among the various implementations, the most significant is *mlsplit*, which uses *Machine Learning* and *split* algorithm and generates a total of 49,077 unique interactions, representing a proportion of 1.81 times more interactions than the 27,041 unique interactions present in STRING. Of this total, 16,681 interactions coincide with STRING, representing 62% of the interactions predicted in STRING. The ratio between the number of unique interactions of the implementation divided by the number of interactions that intersect with the STRING was 2.94, indicating that for each coincident interaction, there is 1.94 interaction that does not match the STRING. These quantitative results demonstrate the ability of the GenPPI program equipped with the SPLIT algorithm to generate predictions of protein interactions with a significant proportion of overlap with the interactions documented in the STRING while presenting a network that is not extremely dense.

IV. DISCUSSION

The protein interactions predicted by GenPPI that are not present in the STRING database do not necessarily imply the non-existence of these interactions or the incorrectness of the GenPPI software. The correctly implemented and conserved phylogenetic profile and gene neighborhood traits conferred between the GenPPI versions speak in favor of a high probability that these interactions actually occur. However, an overly

dense protein network can make it difficult to identify significant differences in the network topology, hindering deeper analyses. On the other hand, an overly sparse network is also not ideal. Therefore, the search for an intermediate-density network becomes a relevant objective.

It is crucial that the predicted protein networks significantly overlap with the protein interactions documented in the STRING database. String overlapping validates the accuracy of the forecasts and strengthens the reliability of the results obtained.

When using the *Features* method without the use of *Machine Learning*, a reduced occurrence of duplicate edges was found. When adopting machine learning techniques, a significant distinction is observed between the *query_unique* and *query_nounique* datasets. When comparing these results with the data from the STRING database, an approximate intersection of sixty percent was verified. In certain data sets, this intersection reached one hundred percent relative to the STRING data; however, significantly higher proportions of non-existent interactions were generated, reaching almost five times more interactions absent in the STRING. In addition, such interaction networks presented a high density of edges, which makes it difficult to identify topological differences in the structure of the networks, compromising the performance of robust analyses.

Generating a smaller number of interactions can lead to a loss of information present in the STRING, but better conditions exist to infer the network topology. On the other hand, when generating a high volume of interactions, it is possible to achieve intersections of up to one hundred percent with the STRING data. However, the density of the network increases, making it more challenging for subsequent analysis.

An interaction can also have a high probability of being true or indicate a lower probability of being a real interaction. To workaround the problem of the high number of interactions, some parameters can be adjusted to increase the confidence of the probability of an interaction. Both GenPPI and STRING provide information on known interactions and predictions of interactions between proteins. STRING, for example, assigns a confidence value or a confidence score to indicate the strength or probability of the interaction. This value is calculated based on a variety of data sources, including experimental evidence, gene coexpression, sequence similarity, and more. The confidence value assigned by STRING and GenPPI is displayed as a number between 0 and 1.

It is also worth noting that when only *Features* without *Machine Learning* is used, the number of edges that are repeated (*query_nounique* less *query_unique*) is greatly reduced.

The results suggest that the interactions generated by GenPPI have a significant overlap with the interactions predicted by STRING, which is considered a gold standard. The interpretation of these data indicates that GenPPI is capable of capturing a substantial amount of interactions similar to those of STRING, thus validating the effectiveness and relevance of the software in the generation of protein-protein interaction networks. This quantitative comparison provides important insights into the quality and accuracy of the interactions produced by GenPPI relative to the standard set by STRING.

The results obtained through the use of GenPPI demonstrate its usefulness in improving the quality and efficiency of investigations. The average accuracy of about 72.63% in selecting similar proteins between genomes without the use of traditional or heuristic methods for aligning protein sequences was a key step contributing to this achievement.

The use of open-source code in all versions of GenPPI presents numerous advantages for scientific research. Access to open-source code provides benefits such as customization, control, cost reduction, collaboration, transparency, and reliability. However, it is important to note that scientific sources such as STRING are not open to the public, which limits access to the source code and can compromise the transparency and reliability of the results.

V. Conclusion

Based on the results obtained with GenPPI, it is possible to infer that the software offers functionalities that can be valuable for researchers in molecular biology and bioinformatics. GenPPI's ability to generate protein interaction networks from specific, unpublished genomic data without depending on external databases and with high accuracy suggests significant versatility. This capability allows researchers to explore protein interactions in lesser-studied genomes or organisms with unique characteristics, thus broadening the scope of scientific investigations.

The results of the latest version of GenPPI demonstrate a significant advance in the area of protein interaction studies.

Acknowledgments

We want to express our gratitude to the Federal University of Uberlândia for the support and resources provided during the completion of this work.

REFERENCES

[1] E. Lopez-Lopez, J. Bajorath, and J. L. Medina-Franco, "Informatics for Chemistry, Biology, and Biomedical Sciences," *Journal of Chemical Information and Modeling*, vol. 61, pp. 26–35, Jan. 2021. [Online]. Available: https://pubs.acs.org/doi/abs/10.1021/acs.jcim.0c01301

[2] E. Gayathiri, P. Prakash, P. Kumaravel, J. Jayaprakash, M. G. Ragunathan, S. Sankar, S. Pandiaraj, N. Thirumalaivasan, M. Thiruvengadam, and R. Govindasamy, "Computational approaches for modeling and structural design of biological systems: A comprehensive review," *Progress in Biophysics and Molecular Biology*, vol. 185, pp. 17–32, Dec. 2023.

[3] T. Tang, X. Zhang, Y. Liu, H. Peng, B. Zheng, Y. Yin, and X. Zeng, "Machine learning on protein-protein interaction prediction: models, challenges and trends," *Briefings in Bioinformatics*, vol. 24, pp. 1–11, Mar. 2023. [Online]. Available: https://dx.doi.org/10.1093/bib/bbad076

[4] L. Hu, X. Wang, Y. A. Huang, P. Hu, and Z. H. You, "A Novel Network-Based Algorithm for Predicting Protein-Protein Interactions Using Gene Ontology," *Frontiers in Microbiology*, vol. 12, Aug. 2021. [Online]. Available: http://www.frontiersin.org

[5] W. Zhang, X. Xue, C. Xie, Y. Li, J. Liu, H. Chen, and G. Li, "CEGSO: Boosting Essential Proteins Prediction by Integrating Protein Complex, Gene Expression, Gene Ontology, Subcellular Localization and Orthology Information," *Interdisciplinary sciences, computational life sciences*, vol. 13, pp. 349–361, Sep. 2021. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/33772722/

[6] G. S. Oliveira and A. R. Santos, "Using the Gene Ontology tool to produce de novo protein-protein interaction networks with IS_A relationship," *Genetics and molecular research : GMR*, vol. 15, Dec. 2016. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28002604/

[7] D. Szklarczyk, A. Franceschini, S. Wyder, K. Forslund, D. Heller, J. Huerta-Cepas, M. Simonovic, A. Roth, A. Santos, K. P. Tsafou, M. Kuhn, P. Bork, L. J. Jensen, and C. von Mering, "STRING v10: protein-protein interaction networks, integrated over the tree of life." *Nucleic acids research*, vol. 43, pp. 447–452, Jan. 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/25352553http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=PMC4383874

[8] Y. Kang, X. Wang, C. Xie, H. Zhang, and W. Xie, "BBLN: A bilateral-branch learning network for unknown protein-protein interaction prediction," *Computers in Biology and Medicine*, vol. 167, p. 107588, Dec. 2023.

[9] M. E. Sardiu, J. M. Gilmore, B. D. Groppe, A. Dutta, L. Florens, and M. P. Washburn, "Topological scoring of protein interaction networks," *Nature Communications 2019 10:1*, vol. 10, pp. 1–14, Mar. 2019. [Online]. Available: https://www.nature.com/articles/s41467-019-09123-y

[10] J. L. Speiser, M. E. Miller, J. Tooze, and E. Ip, "A comparison of random forest variable selection methods for classification prediction modeling," *Expert Systems with Applications*, vol. 134, pp. 93–101, Nov. 2019.

[11] S. Imai, "CL-random-forest: Random forest in Common Lisp," GitHub, Aug. 2024. [Online]. Available: https://github.com/masatoi/cl-random-forest

[12] S. Ren, X. Cao, Y. Wei, and J. Sun, "Global refinement of random forest," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June-2015, pp. 723–730, Oct. 2015.

[13] M. Bastian and S. Heymann, "Gephi : An Open Source Software for Exploring and Manipulating Networks," *ICWSM*, vol. 8, pp. 361–362, 2009. [Online]. Available: http://www.aaai.org

[14] W. F. Anjos, G. C. Lanes, V. A. Azevedo, and A. R. Santos, "GENPPI: standalone software for creating protein interaction networks from genomes," *BMC Bioinformatics*, vol. 22, pp. 1–26, Dec. 2021. [Online]. Available: https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-021-04501-0