Assessing and Improving Protein-Protein Interaction Prediction in *E. coli*

By

Eric Arezza

A thesis submitted to the Faculty of Graduate and Postdoctoral Affairs
in partial fulfillment of the requirements for the degree of

Masters of Applied Science

in Biomedical Engineering with

Specialization in Data Science

Ottawa-Carleton Institute for Biomedical Engineering

Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario, Canada
May 2022

Abstract

This thesis evaluates and extends the state-of-the-art in sequence-based binary protein-protein interaction (PPI) prediction for bacterial species. Accurately predicting PPIs for bacteria enables researchers to quickly identify targets for developing antimicrobial drugs and expand interactome knowledge for bacteria. *E. coli* is used here as a model organism for bacteria. A systematic and unbiased evaluation of four classifiers, SPRINT, DPPI, DEEPFE, and PIPR is conducted on new *E. coli* datasets. Classifier enhancement is accomplished using a stacked reciprocal perspective (RP) classifier, a technique recently developed by the cuBIC lab. Cross-validation results improve by 16.6% for the area under precision-recall (auPR) curve compared to the best base classifier, which increases to 262.5% when considering a 1:100 positive-to-negative sample imbalance. The results of this thesis also indicate the need for new benchmark datasets, more bacterial PPI data, and consistent evaluation protocols to be followed for new PPI predictions.

Acknowledgements

I would like to thank Dr. James Green for his exceptional guidance, patience, and support throughout this research effort. Additionally, I would like to thank my peers in the Carleton University Biomedical Informatics Collaboratory for their helpful insights, advisement, and feedback.

This study was supported by the Natural Sciences and Engineering Research Council of Canada.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	Definition
BioGRID	Biological General Repository for Interaction Datasets
BLAST	Basic Local Alignment Search Tool
CD-HIT	Cluster Database at High Identity with Tolerance
CME	Combination of Multiple Experts
cuBIC	Carleton University Biomedical Informatics Collaboratory
DEEPFE-PPI	Deep Feature-Embedded Protein-Protein Interaction
DIP	Database of Interacting Proteins
DNN	Deep Neural Network
DPPI	Deep-learning framework for Protein-Protein Interaction
GBM	Gradient-Boosting Machine
GOSS	Gradient-based One-Sided Sampling
NIP	Non-Interacting Pair
NN	Neural Network
PIPR	Protein Interaction Prediction based on Siamese Residual RCNN
PPI	Protein-Protein Interaction
RCNN	Recurrent Convolutional Neural Network
RP	Reciprocal Perspective
SPRINT	Scoring Protein Interactions
UniProt	Universal Protein Resource

1 Introduction

1.1 Background

The last two decades of biomedical research have seen a growing interest in discovering how humans and bacteria coexist [1]. This research in microbiomics investigates the pathogenic, symbiotic, metabolic, and therapeutic relationships between colonies of microbiota and animal cells, tissues, and organs. A popular historical example of a therapeutic application in this area showed how a strain of *E. coli* (Nissle) can be used to relieve digestive illnesses in humans [2]. More recently, there has been research into engineering bacteria as a probiotic solution to treat specific metabolic diseases [3]. In other cases, known diseases caused by *E. coli* and other bacterial infection can be remedied using antibiotics while some bacteria build resistance to such treatments [4]–[6]. Aside from the relationships with humans, bacteria are also known to affect plant life and agriculture [7]. Research regarding the fundamental mechanisms for these different biological activities may lead to advancements in such biomedical and horticultural applications in the future.

For that reason, understanding biochemical pathways in organisms is foundational to understanding medicine and microbiology. At the center of many dynamic processes are proteins which exhibit various functions inside and outside cellular environments. The behaviour of protein-protein interactions (PPIs) drives processes such as gene regulation, signal transduction, transport across cellular membranes and cytoplasm, mediating metabolism, and more [8], [9]. Thus, identifying the proteins involved and their interactive

roles provides a greater understanding for use in improving biotechnology and therapeutics.

1.2 Motivation

Traditionally, discovering or confirming the existence of PPIs is performed by laboratory experiments. Most commonly, high-throughput mass-spectrometry techniques and low-throughput yeast two-hybrid experiments have been the standard practice for obtaining confident data of direct protein interactions [10]. Performing these experiments may take months, be resource-intensive due to protein purification processes and equipment involved, suffer from noise, false-positive errors, and missing PPI detections [10]. This can make it unfeasible for using such methods to map entire interactomes of organisms in a reasonable time. Furthermore, there can be limited guidance for selecting which proteins to isolate and test for confirming interactions to avoid needlessly testing pairs that are unlikely to interact. Although efforts at complete interactome profiling have been researched, obtaining all known PPIs within an organism remains a challenge with similar resource and labour drawbacks [11].

As such, efforts to predict likely interactions between proteins using computational tools are pursued to accelerate the understanding of biological pathways by guiding researchers towards interactions that can be verified in the lab. PPI prediction has been attempted with many different models for estimating whether two proteins interact, the strength of interaction, and even sites (sub-sequences) of interactions [12]. Binary classification of protein pairs as interacting (positive) or non-interacting (negative) provides guidance to further investigate details of interactions. Thus, prediction models

used for binary classification can provide a starting point for further analysis of interactomes. These models have applied various machine learning approaches to represent and encode protein data and have become competitive for achieving reliable PPI predictions.

Most commonly, PPI prediction models have been trained and evaluated on outdated human and yeast benchmark datasets; few studies have focused on developing methods specifically for bacterial prediction. Several groups have reported weaker prediction performance for bacteria (often using *E. coli* as a model organism) than for other organisms in the past. Furthermore, results have often been presented without consistent metrics to compare predictors and may not allow for practical interpretations of their evaluations. For example, reporting only accuracy for predictions made on a set of known (positive) PPIs is not informative to discern how well the predictor can perform against non-interacting pairs or how each pair is scored relative to one another. Thus, establishing a comprehensive performance evaluation for bacteria using recently developed state-of-the-art PPI predictors is one goal of this thesis.

A technique developed by the Carleton University Biomedical Informatics Collaboratory (cuBIC) research group has been shown to enhance PPI predictions for several non-bacterial organisms [13]. This technique draws context from prediction scores for each protein in a query pair relative to all scores among all protein pairs to extract features for input into a meta-classifier. As such, this reciprocal perspective (RP) approach can be translated to any pairwise binary classifier. Exploring multiple state-of-the-art PPI predictors and combining RP features presents an opportunity to further

assess and improve PPI prediction performance for bacteria and is another goal of this thesis.

1.3 Problem Statement

Given the importance of understanding bacterial PPIs, and to address the lack of comprehensive reporting on bacterial PPI prediction, this thesis seeks to systematically evaluate the state-of-the-art in PPI prediction for *E. coli* and then improve their prediction performance. This problem requires the following:

1. Establish state-of-the-art PPI prediction performance for bacteria:
    a) Assemble datasets of known *E. coli* PPIs, maximizing dataset size and dataset quality.
    b) Identify state-of-the-art methods with high self-reported performance on *E. coli* and implement them locally.
    c) Evaluate those methods externally from their self-evaluations to ensure fair, rigorous, and systematic performance evaluation over the *E. coli* datasets.
2. Improve PPI prediction performance:
    a) Evaluate augmentation of each PPI prediction expert using RP, as RP has not yet been used within bacteria.
    b) Combine multiple experts using various approaches and RP.

1.4 Summary of Contributions

The work presented here contributes to bioinformatics research in three ways. First, it provides an algorithm for extracting updated and high-quality datasets and introduces new compiled *E. coli* PPI datasets. Secondly, it implements and evaluates four state-of-the-art PPI prediction algorithms using a consistent evaluation protocol to gain an objective comparison of bacterial PPI prediction performance. Finally, it proposes a method to improve PPI predictions through application of reciprocal perspective and a combination of multiple experts using a stacked classifier that leverages context. Overall, the results of this thesis show that the *E. coli* PPI prediction methods developed here provide the greatest reliability in predicting bacterial PPIs. Novel predicted bacterial PPIs can then be confirmed in the lab for accelerating our understanding of biological pathways in bacteria for use in biotechnology and therapeutic treatments. All datasets and source code used in this thesis has been made publicly available at https://github.com/GreenCUBIC/Bacterial-PPI-Prediction to encourage the research community to pursue further advancements.

1.5 Organization of Thesis

This thesis consists of four more chapters with multiple subsections. Chapter 2 provides a review of literature regarding the collection of PPI data, existing PPI prediction approaches with a description of four state-of-the-art methods used in this work, and a brief explanation of the reciprocal perspective technique. Chapter 3 presents an independent evaluation of four state-of-the-art methods on *E. coli* with comparisons drawn between methods and datasets. Chapter 4 develops two methods to improve PPI

prediction in *E. coli*: application of RP to each of the four baseline predictors and using RP to combine multiple experts (CME) to enhance predictive performance. Finally, Chapter 5 provides an overall summary of this thesis with recommendations for the future.

2 Literature Review

This chapter contains four subsections to provide an overview of the protein data landscape of *E. coli* and PPI prediction methodologies as a precursor to understanding the applications employed in this work. The first section reviews the currently known *E. coli* proteome and interactome data. Section 2.2 reviews methods for the curation of datasets. Section 2.3 briefly introduces general approaches to computational PPI prediction. In Section 2.4, four state-of-the-art methods used in this thesis are presented. Lastly, Section 2.5 describes the reciprocal perspective method.

2.1 Protein Data Landscape

The task of PPI prediction can be considered as a supervised classification problem. As such, the ability to predict protein interactions using a computational model depends largely on the quality and quantity of available data provided to the model. Therefore, this section investigates sources and tools used to collect and process PPI information for effective use in PPI prediction.

2.1.1 The Escherichia Coli Proteome

The Universal Protein Resource (UniProt) database [14] provides comprehensive protein information and cross-referencing to 180 external databases. It consists of the UniProt Knowledgebase (UniProtKB) which provides annotated protein information, UniProt Reference Clusters (UniRef) to search for similar proteins among sequence clusters, and the UniProt Archive (UniParc) which archives sequences and their identifiers. UniProtKB

is further separated into data that are automatically annotated (TrEMBL) and data that are also manually annotated and reviewed by human experts (SwissProt) [14].

A complete genomic sequencing of *E. coli's* single chromosome has been accomplished as early as 1997 [15]. This sequencing led to an established reference proteome made of 4438 proteins – of which 4390 have been manually reviewed – and is publicly available through UniProt as Proteome ID UP000000625. This K-12 strain (MG1655 sub strain) proteome has been assembled and annotated from the European Molecular Biology Laboratory (EMBL) [16] in collaboration with EcoCyc [17]. More generally, UniProt contains 8174 reference proteomes for bacteria at the time of this thesis, each with varying completeness. Bacterial proteins also make up 59% of the SwissProt database out of a total 14,132 species represented. *E. coli* (strain K12) is also the 8th most represented species and highest of all bacteria.

Besides UniProt as a source for information from a network of databases, earlier efforts to make *E. coli* protein information available includes EchoBase [18] and EcoProDB [19] and may also be cross-referenced by UniProt. Additionally, PATRIC [20] provides a database and analysis tools for bioinformatics researchers investigating genomic, proteomic, and transcriptomic information for bacteria and related diseases. As for the data collected for use in this thesis, UniProt will be used as the source for protein references as it is accessible, widely referenced, and regularly maintained.

2.1.2 Escherichia Coli Interactome

More recent proteomic studies of *E. coli* have investigated how the expressed proteome changes under various cellular and environmental conditions [21]–[23]. These conditions

can affect which proteins are available to interact with each other, limiting the potential size of an interactome. Studying an interactome under different conditions may provide better understanding of relevant interactions but adds complexity to an already complicated system. Moreover, the functional information of all proteins can be incomplete and further limit the known interactome [24], [25]. Thus, this thesis will consider the entire genetic proteome as it allows for a more comprehensive examination for any potential interactions that may exist under any condition.

A proteome size of 4,438 proteins would result in 9,850,141 possible interactions for *E. coli*. However, many of these interactions are not likely to occur due to various reasons that limit their ability to make contact. This can be due to proteins not being expressed at the same time (co-expression), not being expressed in the same regions of the cell (co-localization), and lacking binding affinities or docking sites between them [26]. Therefore, elucidating proteins that do interact can be a difficult task without some guidance. To add, determining cross-species interactomes whereby proteins from different organisms interact becomes a greater challenge when combining both species' proteomes. Thus, prioritizing protein pairs for experimental investigation has often relied on heuristic filters, such as testing proteins found in similar subcellular locations or those thought to be related to similar functions and pathways.

As mentioned, many lab techniques are used to verify the presence of interacting protein pairs. These lab techniques can be classified as biochemical (co-immunoprecipitation, western blotting, pulldown assays, etc.), genetics-based (yeast-2-hybrid, gene co-expression, etc.), biophysical (tandem affinity purification-mass

spectrometry, co-localization imaging, etc.) or some combination of them and vary in accuracy, throughput, noise, and level of ability to identify interaction type [27].

As wet-lab researchers contribute to identifying proteins and interactions, growing databases organize and store this information. Dozens of interaction databases such as DIP [28], BioGRID [29], MINT [30], IntAct [31], HPRD [32], STRING [33], and more [34], [35] host these experimentally determined PPIs. Each database's information is curated differently and can focus on certain species, aspects of biological significance, reliability of entries, and may contain overlapping data. Meta-databases also attempt to consolidate external databases and apply filters to build more comprehensive and confident PPI information [36], [37]. Regarding sources containing *E. coli* PPIs, Table 1 presents a survey of this PPI data as of March 2022. Note that these numbers may change over time, may contain redundant interactions determined by different experimental methods or publications, contain different sub strains of *E. coli*, and sources could also contain smaller "high-quality" sets of PPIs depending on filters for recorded interactions.

Table 1: Databases containing *E. coli* PPIs.

Source	Number of PPIs Listed	Number of Proteins
BioGRID [29]	186,296	4,026
DIP [28]	13,379	2,994
IntAct [31]	24,779	3,072
MINT [30]	6,857	3,556
STRING [33]	1,083,186	4,126
HitPredict [38]	18,244	3,290
PATRIC [20]	4,945	3,113
APID [37]	37,985	3,844
EcID [39]	1,847,729	4,150
Bacteriome.org [40]	9,860	2,131

Although this thesis focuses on *E. coli*, it is worth mentioning that there are also efforts to collect, record, and organize bacterial PPI information in general. For example, there is the Microbial Protein Interaction Database (MPIDB) that contains interactions from 191 different species [41] and MPI-LIT which attempted to build a standard dataset of microbial PPIs to increase prokaryote representation in PPI databases [42]. There is also MorCVD [43] that focuses on host-pathogen interactions in relation to microbe-induced cardiovascular diseases as well as HPIDB [44], [45] that contains host-pathogen PPIs.

Researchers attempting to computationally predict PPIs have also constructed standard “benchmark” datasets to evaluate their models. One such *E. coli* dataset collected from DIP in 2011 that has been used in PPI model evaluations consists of 6,954 known interactions [46]. The next section will present ways that datasets are built for use

in training and evaluating PPI predictors. A compilation of datasets used in this thesis are also presented in Section 3.2. In the subsequent sections, BioGRID will be referenced as the source for compiling datasets as it is regularly updated, contains enough information to assess PPIs published, provides data for many organisms, and has a simple curation workflow.

2.2 Curation of Datasets for PPI Prediction

In this section, methods for building PPI datasets are discussed with a proposed algorithm for building reliable datasets. The first subsection presents a filtering approach for selecting known interactions. The following subsections describe a process to improve confidence in chosen PPIs, remove redundancies in sequences, and methods for acquiring non-interacting PPIs.

2.2.1 Selecting Reliable Interactions

As mentioned above, interactions can be detected through various experimental procedures with different levels of accuracy and noise. This may result in false positives being detected by some wet-lab methods more often than others. Considering this, selecting reliable sources for PPI data is crucial.

One major difference in PPI experimental data is whether the interaction is detected by a genetic interaction or physical interaction. The detection of a PPI as a genetic interaction provides inferred evidence that two proteins interact based on their related gene changes. For example, in a yeast-2-hybrid experiment a bait protein is bound to a DNA site which is expressed if a prey protein binds to the bait [27]. This implies that the prey protein is the cause for gene expression; however, false positives can result from

this assumption. In contrast, detection by physical interaction provides stronger and more direct evidence that two proteins directly engage in an interaction. For example, in a pull-down assay prey proteins are bound to immobilized bait proteins and the remaining cellular contents are washed away [27]. The bait-prey protein complexes can then be analyzed to confirm PPIs. For these reasons, selecting PPIs based on experiments using physical detection methods is preferred to reduce possible false positives in PPI data.

Among PPIs detected by physical interaction, the type of detection method used in the experiment can also affect the quality of data collected and used for predictions. An investigation of this was performed using BioGRID data filtered by the POSITOME [47] algorithm which was developed to retain high quality PPIs. It showed that restricting available interactions to those discovered through conservative detection methods can improve PPI prediction performance of PIPE [48] (a sequence-based computational PPI prediction method) for various species. Thus, the quality of a dataset can overcome the drawback of reduced number of PPIs available for training the model.

This approach of selecting reliable and more conservative interactions from available PPI databases can be applied here for *E. coli*. As shown in Table 1, 186,296 intra-species interactions are listed in BioGRID datasets pertaining to *E. coli* (Release 4.4.203). However, most entries have been identified through genetic interaction detections. Only 15,071 of those PPIs are supported by physical interaction detection methods. Of note, most of those physical interactions were reported by two large-scale wet lab studies: [49] and [50] publishing 12,798 and 2,183 interactions, respectively.

2.2.2 Strengthening Confidence in Interacting Pairs

Once physically detected interactions have been selected among the experimental data, additional filters can be applied to further increase confidence in the data. For instance, one can retain only those PPI that have been reported in multiple studies and use supporting evidence from expert knowledge and external databases to strengthen confidence in selected PPIs. Applying these ideas to a set of known interactions will further improve the quality of data collected to remove possible false positives, while reducing the number of PPI remaining in the dataset, leading to a quality versus quantity trade-off.

For example, using the 15,071 *E. coli* PPIs mentioned above, 173 of those are duplicated which leaves 14,898 in total. Out of these, only 129 have multiple sources of evidence. In the most restrictive case, only 2 PPIs have multiple sources of evidence by different authors. Therefore, a categorization for the strength of confidence in PPIs listed by BioGRID can be sorted by levels 0, 1, or 2 for each subset of 14,898, 129, and 2 PPIs, respectively. It is obvious that the highest confidence set of 2 PPIs is impractical for training a prediction model, so this quality versus quantity trade-off should continue to be taken into consideration.

Additionally, references to the protein sequences contained in each pair should also be reliable. In this case, queries to the SwissProt database can ensure that protein information has been reviewed both automatically and manually by experts. This also ensures that each protein in the pair has rich annotations (e.g., regarding function) that enables further investigations into the details of interacting pairs. In the following sets of *E. coli* PPIs discussed, any proteins that do not exist in the SwissProt database or those

that have no sequence information can be removed from the datasets. This way, the resulting protein pairs can be more confidently accepted due to higher confidence in their protein constituents. From BioGRID, mapping the Entrez gene IDs in PPIs to their SwissProt Accession protein IDs can be accomplished using the programmatic access application programming interface (API) provided by UniProt. Including only proteins for which this mapping is available reduces the level-0 dataset to 2,039 PPIs composed of 1,236 proteins.

2.2.3 Reducing Data Redundancies

When providing training datasets to prediction models, the existence of multiple protein homologs (i.e., proteins that have evolved from a common ancestral protein and therefore share significant sequence similarity) can introduce a redundancy in detected patterns and thus bias the model's training. This causes a positive feedback bias whereby the model is inclined to predict PPIs containing homologs more heavily than other possible PPIs. This reduces the model's ability to effectively generalize for different datasets. Therefore, ensuring the uniqueness in the proteins of a PPI dataset is typically done to avoid problems with training predictive models.

Ensuring sequence uniqueness is normally accomplished by clustering similar proteins and keeping only a single representative protein sequence from each cluster. There are many algorithms and software implementations for grouping similar proteins together. Most commonly, clustering by sequence similarity is performed by sequence alignment tools. A frequently used sequence clustering tool used in PPI prediction

literature is the Cluster Database at High Identity with Tolerance (CD-HIT) program [51]. The algorithm for CD-HIT can be described as follows:

1. All sequences are sorted longest to shortest by number of amino acids.
2. Using the longest sequence as a representative of the current cluster:
   I. Find all other sequences that have greater or equal to *X*% shared sequence identity to the representative (where *X* is a threshold defined by user input).
   II. Group all sequences sufficiently similar to the representative into a cluster.
   III. Use the next largest sequence that had less than *X*% sequence identity from the representative as a new representative sequence for a new cluster.
3. Repeat

Thus, if the sequence identity threshold is set to 1.0, only true 100% duplicate protein sequences will be clustered, and then could be removed from the dataset. As the threshold is lowered, more proteins are clustered and only representative proteins may be retained in the dataset. A frequent choice for this CD-HIT threshold is 0.6. Again, applying this to the *E. coli* dataset results in 2,005 PPIs and 1,221 proteins (down from 2,039 PPIs and 1,236 proteins).

The final dataset of positive PPIs comprises those pairs for which both proteins can be mapped to gene IDs and where both proteins are retained following homology filtering with CD-HIT. For level-0 and level-1 confidence, the number of positive PPIs is 1998 interactions involving 1,221 proteins and 125 interactions involving 194 proteins, respectively. Overall, applying these filters to raw datasets provided by databases can be performed to curate reliable and confident positive PPIs. Adjusting the stringency of the

filters results in an opportunity to explore a quantity versus quality trade-off for use in computational prediction tasks.

2.2.4 Generating Non-Interacting Pairs

Once a confident set of known positive PPIs has been collected, a set of non-interacting protein pairs (NIPs, or negative PPIs) is often required for training and validation of PPI classifiers. Unfortunately, as most researchers aim to find positive PPIs, there is less data available from which to identify high-confidence NIPs. To compound this, some PPIs may require specific conditions to exist and can be masked be more abundant proteins and their interactions making them difficult to detect resulting in a false negative. Additionally, as mentioned above most proteins are presumed to not interact. This leads to a class imbalance whereby more NIPs are needed than positive PPIs to build datasets that will accurately reflect this imbalance. For these reasons, there have been a few approaches to obtain NIPs through literature and database mining and by randomly generating NIPs.

An effort to collect protein pairs that do not interact is the Negatome database [52], [53]. Negatome hosts 9 datasets that contain between 1,234 to 6,532 pairs (depending on its compilation of literature and database sources) that are unlikely to interact. Despite this, there may not be sufficient species-specific interactions from which to build an *E. coli* dataset for classification purposes (i.e., a dataset where there are an equal number of positive and negative examples). Due to a lack of known non-interacting protein pairs, Moscatelli [54] describes other approaches that have been devised to obtain negative PPIs and some of their drawbacks.

One simple approach is to take a list of proteins and randomly pair them such that there are no known positive PPIs in the set of pairs. This random sampling is widely used and considered to be reliable for predictions and evaluations of classifiers. However, investigations by Yu *et al*. [55] and Park *et al*. [56] showed that generating NIPs by random sampling this way can affect the apparent performance of a PPI predictor. In essence, a balanced random sampling of NIPs to positive PPIs is preferred when training a model; however, evaluations should use imbalanced sets to accurately reflect the higher NIP to PPI ratio presumed to exist among all possible protein pairs.

An additional approach to generating NIPs goes a step further and ensures that the proteins in each pair are found in different subcellular locations. This assumes that the two proteins only exist in isolated environments from each other within cells, so they would likely never interact. An obvious inherent drawback of applying this approach is that it reduces the protein space from which possible NIPs can be generated. More importantly, studies [57], [58] showed that the performance of classifiers is affected by biases in the tolerance of protein pair co-localization and classifiers trained this way are more likely to produce false positive predictions. Thus, considering different subcellular locations for generating NIPs should be avoided.

Besides random sampling and selecting co-localized protein pairs, compiling a high-quality set of negative PPIs has undergone further investigations with the following criticisms. Most recently, the Neglog method [59] was developed to supplement the available data in Negatome by inferring that gene paralogs (genes within the same organism derived from the same ancestral gene) and orthologs (genes in different organisms derived from a common ancestor) of NIPs found in Negatome are also likely

to be non-interacting. Training PPI predictors using Neglog NIP data showed improvement over using purely random sampling, but Neglog still applies random sampling to reduce biases mentioned above. Besides Neglog, authors suggested that random sampling may be the next best method to obtain NIPs and may be required when creating large datasets. Therefore, to simplify dataset creation and ensure enough species-specific proteins are available, Neglog is not used in this thesis.

For the reasons discussed above, generating negative pairs by random sampling of proteins may provide the most reliable approach for developing and evaluating computational classifiers. To address the imbalance in positive and negative pairs in the real world, one can choose to sample more negatives than positives when evaluating performance on test sets. In this thesis, a balanced dataset is compiled for both training and testing PPI predictors, while a prevalence-corrected calculation is performed to address a hypothetical imbalance during evaluation of PPI predictors. This simplifies training because selecting an imbalance ratio would be arbitrary and difficult to correctly assume. It also allows evaluations to be easily simulated for any imbalance. This is further described in Section 3.1.

2.3 Computational Approaches to PPI Prediction

In silico PPI classification methodologies have employed many approaches to predict PPIs using computation techniques. The supplied data can include protein domain physiochemical information, sequences, evolutionary analysis, structural information, topological networks, or combinations thereof. Each encompasses contextual biological data as features for PPI prediction and may have strengths over one another depending

on the type of interaction being predicted and inferring functional associations or direct interactions [60]. These features can then be implemented in various classifier methods that use statistical scoring approaches, support vector machines (SVM), decision trees, and most recently neural networks (NN). The following subsections will only briefly describe each approach as they have been reviewed in great detail elsewhere [10], [12], [35], [60]–[62] from which these subsections take reference. A focus on sequence-based approaches is presented as these have had the most applications within *E. coli* [61] and are therefore most relevant to this thesis. Subsequently, a summary of four sequence-based state-of-the-art PPI prediction methods that are used in this thesis are discussed in Section 2.4.

2.3.1 Gene-based Approaches

Genetics-based predictions apply concepts of gene fusion, gene neighborhoods, and domain co-expression to infer possible protein interactions by considering their gene-level characteristics. In gene fusion, genes in one organism may be found as a fused concatenation in another organism suggesting that the separate genes could be functionally related. A clear drawback to this is that gene fusions do not occur frequently. Gene neighborhoods consider that genes within a small distance of one another may present functional relationships due to co-expression of the translated proteins. A disadvantage of using gene clusters is that it is difficult to predict PPIs for proteins that are encoded by genes farther from each other. Finally, domain co-expression can be used by considering that genetic domains sharing similar expression levels implies that their

proteins may be related and interact. Overall, gene-level relationships may only indirectly infer protein-level interactions.

2.3.2 Evolution-based Approaches

Evolutionary information used in predicting PPIs includes phylogenetic and coevolved similarities between organisms. A phylogenetic profile for orthologous proteins among organisms can be used to suggest that if two proteins share similar profiles then they may provide the same biological function and thus may interact with each other. Phylogenetic trees also infer interactions by suggesting that proteins must be functionally related if they have co-evolved and are conserved between different species. Drawbacks to these approaches include an inability to predict PPIs for unique proteins with no orthologs and a sensitivity in selecting organisms and genomes used for detecting orthologs and producing phylogenetic profiles and trees.

2.3.3 Network-based Approaches

Interaction networks are used to make PPI predictions by studying topological relationships between proteins known to interact. In these networks, nodes represent proteins and edges that connect nodes represent interactions between proteins. A densely connected region of nodes in a network can suggest that those proteins may be functionally related and could form protein complexes. Thus, proteins within these hubs that are not directly connected to each other by an edge can be predicted to interact with proteins within its hub and with proteins in connected hubs. Again, as with the above approaches an interaction network alone may imply associations indirectly and can

benefit from additional information such as structural and sequence data when building network edges and making predictions.

2.3.4 Structure-based Approaches

Structure-based approaches use the 3D shapes of proteins and associated domain physiochemical properties to predict PPIs. Identifying orientations for docking of two proteins and interface sites with high binding affinity enables PPIs to be predicted. Unlike the above approaches, structure-based methods allow interactions to be directly predicted using physical protein information. Currently, the availability of protein sequence information greatly exceeds that of protein structures making this approach more difficult to implement. However, growing capabilities in identifying protein structures such as by AlphaFold [63] and RoseTTAFold [64] may soon enable structure-based approaches to be broadly applied and such methods may provide greater characteristic insights of interactions.

2.3.5 Sequence-based Approaches

Sequence information is often fundamental to enable the approaches described above but can also be used directly for predicting PPIs. Contextually, the physical interaction between two proteins occurs through direct contact between their amino acids. Several methods hypothesize that pairs of short sequence motifs are reused in multiple PPIs to enable such interactions [48], [65]. It is also more likely that similar sequences share similar traits, so known interacting pairs can be used as reference to predict interactions for homologous proteins. Protein sequences also gain the advantages of having the most extensive coverage of protein data available, being simple to comprehend, and being

compatible with established bioinformatics tools for processing and understanding relationships such as BLAST [66], [67], for example. To add, recent advancements and popularity in natural language processing (NLP) and deep neural networks (DNN) have made sequence-based approaches an obvious candidate for use in computational PPI prediction.

Some older sequence-based studies have examined PPI predictions for *E. coli* and other bacteria (most often *H. pylori*) reporting weaker performance than for other organisms. It should also be noted that these methods most often reported only a few metrics using a single decision threshold for classification which does not reveal the overall performance of PPI predictions; this is discussed in Section 3.1.2. Sarkar and Saha [61] presented a survey of sequence-based methods using different classifiers. The number of those studies investigating different organisms is shown in Table 2 below.

Table 2: Organisms investigated by sequence-based predictors presented in [61]**.**

Organism	*E. coli*	*H. pylori*	Human	Yeast	Other
Number of Studies	4	5	7	14	2

As early as 2001, SVM classifiers were used by [68] and [69] for PPI prediction and reported weaker results for *E. coli* than for other organisms. Across subsequent *E. coli* PPI prediction studies, data were frequently sourced from the DIP database. A benchmark *E. coli* positives-only dataset was compiled in 2011 [46] (described in Section 3.2.1) and was used in many subsequent prediction methods. The SVM by Zhou *et al*. [46] also showed weaker results for *E. coli* compared to other non-bacterial species. More work in improving SVM classification using bacterial PPIs was also conducted in 2015

[70], [71]. More recently, state-of-the-art DNNs have been used for PPI predictions and tested on *E. coli* data presenting high accuracy (>96%) results [72]–[74]. The investigation herein aims to independently and comprehensively assess the predictive performance of more recent state-of-the-art PPI predictors for use in *E. coli* PPI prediction and improve upon them.

2.4 State-of-the-Art Sequence-Based Methods

Sequence-based methods attempt to identify patterns in sequences among interacting proteins that indicates a strong binding affinity for those amino acid patterns. Recognizing these patterns has been a goal for state-of-the-art methods which have recently leveraged NLP and deep learning to build strong predictors. The sequence-based methods used in this thesis were selected based on accessibility of source code, ease of implementation, recency in publication, and reported prediction performance claimed.

2.4.1 SPRINT

Li and Ilie [65] presented a fast PPI prediction method named SPRINT to predict the human interactome. It is based on cumulative scoring of subsequence similarities between proteins known to interact and the query protein pair and is conceptually similar to PIPE [48], [75], [76] . In an example illustrated by Li and Ilie shown in Figure 1 below, an interaction between protein pairs (P_2-Q_2) and (P_3-Q_3) can be inferred based on pairs of subsequences that also exist in a known interacting pair (P_1-Q_1). In this way, pairs containing more subsequence similarities to known PPIs are thought to have higher likelihood of interacting and will therefore be given higher interaction scores.

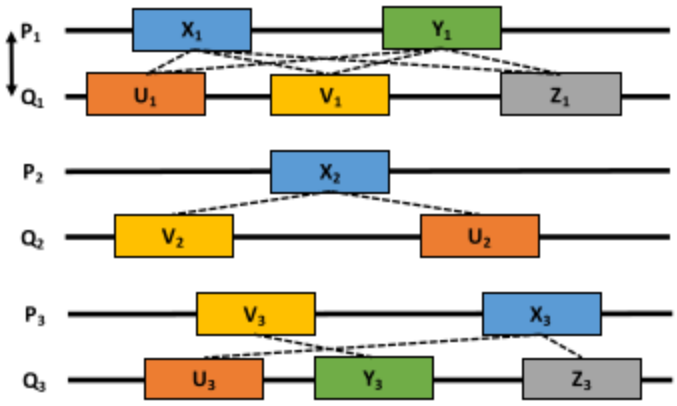

Fig. 3 Interaction inference. The proteins P_1 and Q_1 are known to interact; blocks of the same colour represent occurrences of similar subsequences. Dashed lines indicate potential contributions to interactions: there are six between P_1 and Q_1 and they imply two between P_2 and Q_2 and three between P_3 and Q_3

Figure 1: SPRINT inference of interacting pairs using subsequence similarities as illustrated in [65].

To elaborate, SPRINT begins by sliding 4 spaced seeds (11****11***1, 1**1*1***1*1, 11**1***1**1, 1*1******111) along each of the given sequences and storing the resulting spaced-mers into 4 respective hash tables, as depicted in Figure 2 (e.g., 5 spaced-mers).

```
MVLSPADKTNVKAAWG
MV____DK___K
 VL____KT___A
  LS____TN___A
   SP____NV___W
    PA____VK___G
```

Figure 2: Example of sliding the first spaced seed along a sequence to obtain five spaced-mers.

Then similarities are calculated for each alignment between spaced-mers. For each alignment of spaced-mers, they are extended in both directions while their similarity remains above a given threshold to capture a high scoring subsequence similarity pair (HSP). These HSPs are stored and used to infer interacting pairs as illustrated above in Figure 1. Thus, scoring interacting pairs involves identifying HSPs between protein pairs.

SPRINT was presented by comparing results of predicting the human interactome using human PPI data from several database sources and employed an evaluation

scheme described by Park and Marcotte [77] which is discussed in Section 3.1. Shown in [65], an average over all test sets from all database sources resulted in an area under the precision-recall (auPR) curve of 0.8522 and an area under the receiver-operating characteristic curve (auROC) of 0.8308.

2.4.2 DPPI

Hashemifar *et al*. [72] developed DPPI, a deep learning model, for PPI classification using protein sequence profiles as a position-specific scoring matrix (PSSM) for input to convolutional neural networks. Shown in Figure 3, protein profiles generated from sequences using PSI-BLAST [67] are passed through five convolutional layers before a random projection module maps the protein pair's final representations for prediction.

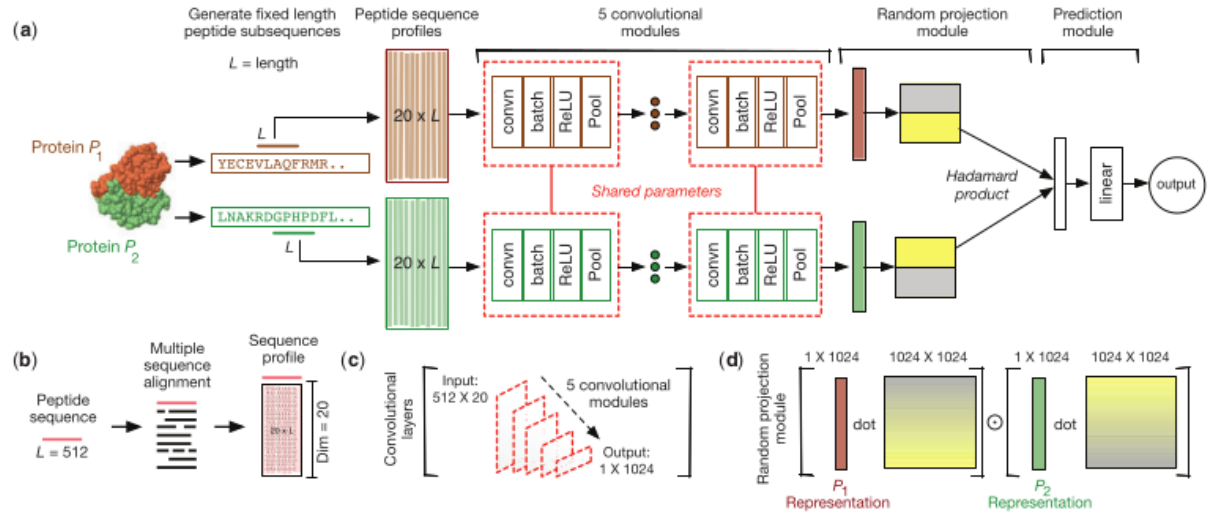

Fig. 1. Illustration of the DPPI model for predicting binary protein–protein interactions. (**a**) DPPI takes as input a pair of protein sequences and learns to predict an interaction score. (**b**) Each input protein is represented as a probabilistic sequence profile generated by PASI-BLAST. (**c**) The convolutional module consists of several convolutional layers that learn a set of filters, each of which is responsible for detecting various patterns in a sequence. (**d**) The random projection module maps each sequence to a representation useful for modelling paired sequences

Figure 3: DPPI model architecture as illustrated in [72]**.**

Using PSSMs as input was selected with the intention to capture relevant biological and evolutionary information from individual amino acid sequences. A PSSM is created by performing an alignment of a query sequence against a database of protein sequences. For each of the 20 amino acids, a score is given for each position in the query sequence. These scores indicate the probability of replacing the observed amino acid with each of the 19 other amino acids through random mutations (estimated from multiple sequence alignment). Subsequently, the PSSM provides information about distant protein relationships and conservation of amino acids among sequences within a database which can include proteins from different organisms.

In PSI-BLAST [67], an E-value is a threshold parameter calculated as a statistical p-value multiplied by the database size to determine which database sequences are significantly similar to the query protein. Those proteins are added to the alignment and the resulting PSSM. The resulting PSSM can then be used to iterate a PSI-BLAST search again to find any more significant sequences and generate an updated PSSM.

For DPPI, PSI-BLAST is used to generate protein profiles by running 3 iterations using an E-value of 0.001 to search through UniProt/SwissProt database. The original model searched through the 2016 UniProt/SwissProt database, however, the work presented here used the updated UniProt/SwissProt sequence database from July 2020 consisting of 562,755 sequences. The resulting profiles are n x 20 arrays where n is the length of the input sequence. To handle sequences of different lengths during training, DPPI crops these n-length profiles to 512 amino acids with subsequence overlapping at multiples of 256 until the full sequence length is covered.

Profiles are convolved with filters of width 512, 256, 128, 64, and 20 until the final 1 x 1,024 representation is achieved. The network uses five batch normalization stages, rectified linear unit nonlinearities, and max-pooling layers. Then, a random projection module is applied to each protein representation to improve the predictive power of DPPI and enable it to be insensitive to the order of the two input proteins. Finally, a prediction probability is calculated using a linear combination of a bias term with the dot-product of each protein's representation multiplied by a weight factor.

2.4.3 DeepFE-PPI

Yao *et al.* created DeepFE-PPI [73], herein synonymously referred to as DEEPFE, which employed Res2vec, the protein equivalent of the NLP encoding method Word2vec [78], to create numerical vector representations of each proteins' amino acid sequence. These protein sequence representations are then used in a deep neural network framework to predict interactions as illustrated in Figure 4 below.

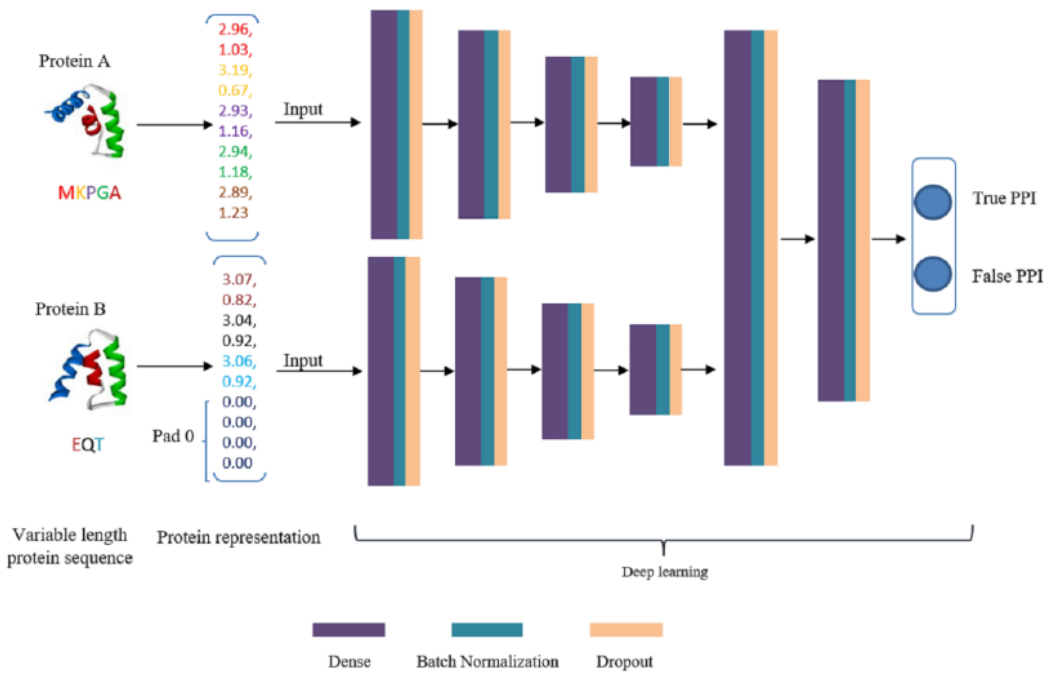

Figure 4: DEEPFE framework as illustrated in [73].

The use of Res2vec for representation learning of protein sequences in DEEPFE is a feature embedding approach to encapsulate the effects of surrounding amino acids in a sequence. In this context, a residue (amino acid) can be considered a word, the entire sequence is a sentence, and the full list of sequences in the dataset is the total corpus used by Res2vec to learn representations. Depicted in Figure 5, each residue is represented by eigenvectors which become concatenated for the entire sequence. DEEPFE used a Skip-gram model and the SwissProt database from 2018 containing 558,590 proteins to train and build the Res2vec model through representation learning. In the local implementation of DEEPFE used in this thesis, an updated SwissProt database from July 2020 consisting of 562,755 sequences was used instead.

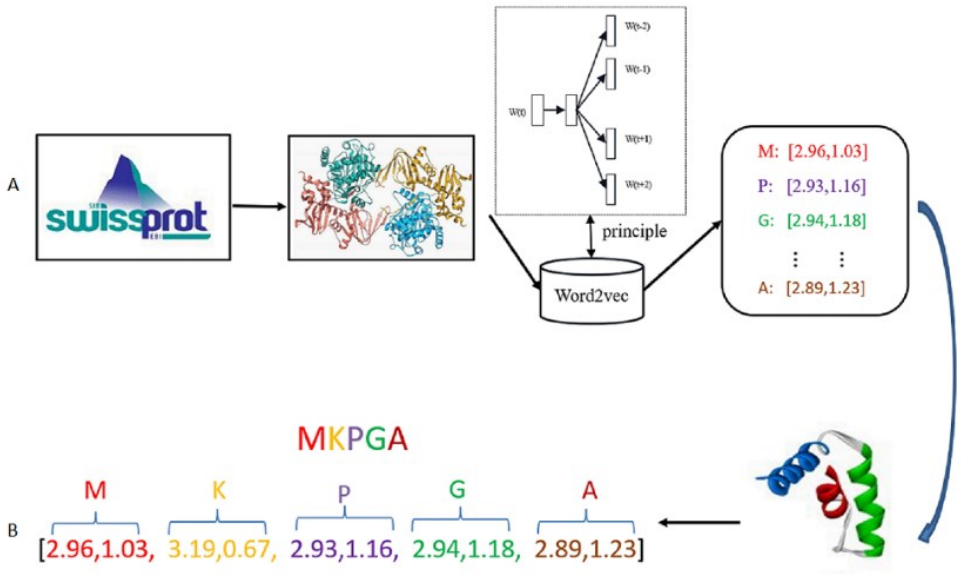

Figure 5: Implementation of Res2vec used in DEEPFE as illustrated in [73]**.**

DEEPFE was originally evaluated on the yeast benchmark dataset and a human dataset from Huang *et al*. [79] using 5-fold cross-validation. The performance metrics reported were accuracy, precision, recall, and Mathew's Correlation Coefficient (MCC) while only considering a threshold score of 0.5 for binary classification. A Park and Marcotte (P&M) evaluation scheme was also used to evaluate DEEPFE on the yeast dataset and reported accuracy, precision, recall, specificity, MCC, F1-score, area under ROC curve, and area under precision-recall curve. Finally, in [73] an accuracy (equivalently recall) of 100% was reported for DEEPFE trained using yeast and tested on *E. coli* and *H. Pylori* positive-only datasets.

2.4.4 PIPR

As with the previous two methods, Chen *et al*. [74] also used a Siamese type deep learning architecture for PPI classification with a residual recurrent convolutional neural network (RCNN) that they named PIPR (Protein–Protein Interaction Prediction Based on

Siamese Residual RCNN). Shown in Figure 6, sequences are transformed using a pre-trained embedding before passing through the residual RCNN network to form embedded vectors for each protein in a pair. The pair of vectors are then combined and passed through a simple multi-layer perceptron (MLP) network to make final predictions. Beyond binary classification of PPIs, PIPR can also predict the strength of interactions and the interaction type.

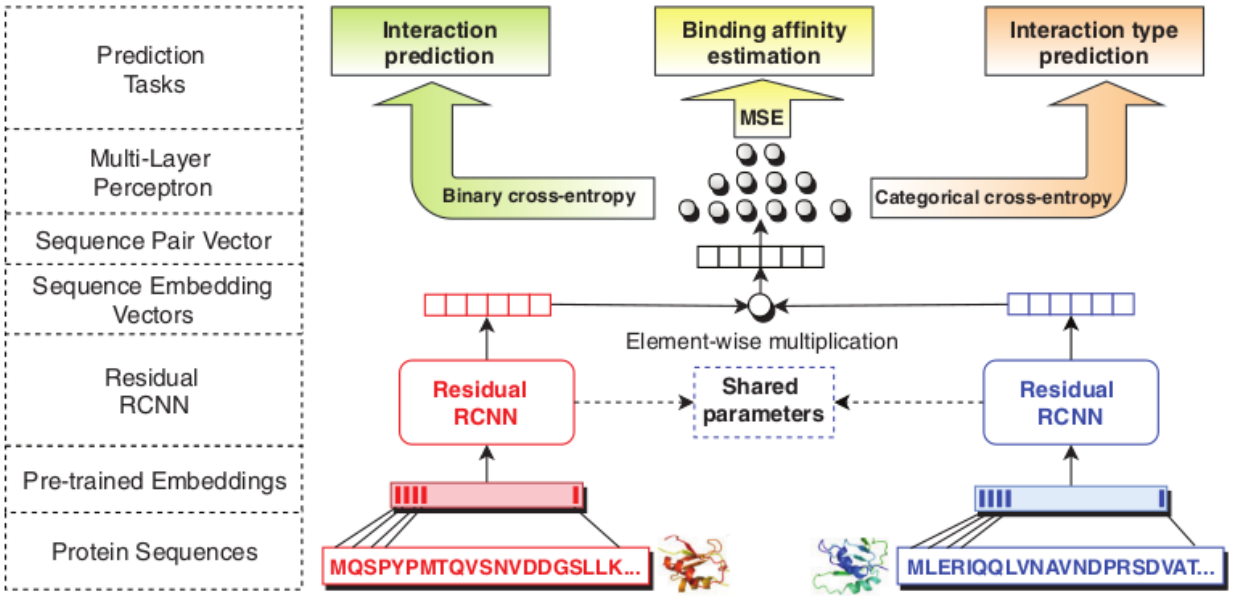

Figure 6: PIPR framework as illustrated in [74].

Similar to DEEPFE, the pre-trained embedding had been built using a Word2vec Skip-gram model to provide sequence context of amino acids. In PIPR, an 8000 human protein sequence dataset from STRING was used to train this embedding. Additionally, electrostatic and hydrophobic properties of amino acids were one-hot encoded in the embedding based on a 7-class clustering of 20 amino acids defined by Shen *et al*. [80]. The pre-trained embedding converts input protein sequence characters to numerical vectors that are then put through the RCNN seen in Figure 7 to encode each protein

sequence. The use of the residual RCNN allows variable length sequences to be used and obtains both local and sequential protein features from sequence pairs. The final sequence embedded vectors are combined through element-wise multiplication which is passed to a MLP to make predictions.

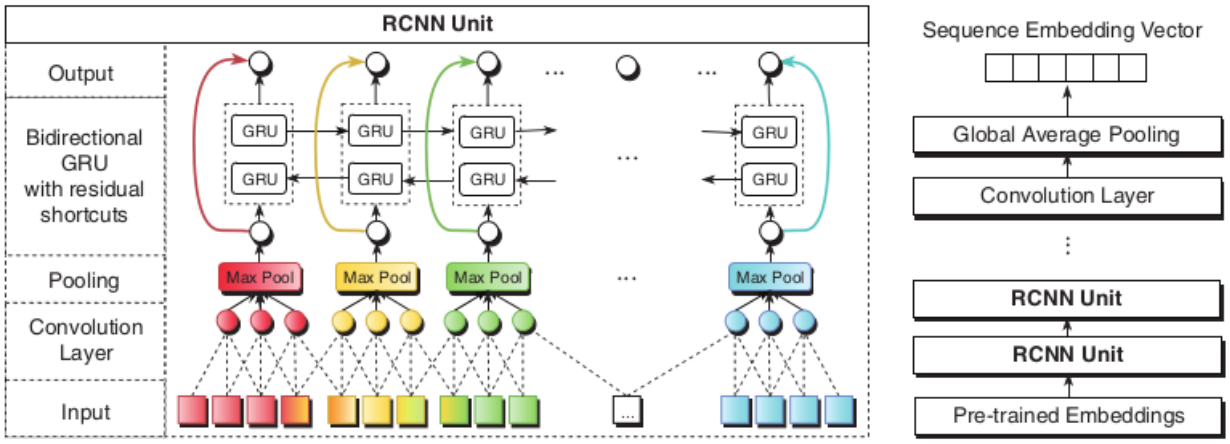

Figure 7: PIPR RCNN sequence vector encoding structure as illustrated in [74]**.**

2.4.5 Notable Considerations

For the purposes of this thesis, investigating PPI prediction for bacteria using older published methods would be resource exhaustive, counter-intuitive to the progression of PPI classification, and provide inconsequential relevancy to the applications proposed in this thesis. With that mentioned, it is understood that there is potential for an older predictor to achieve superior classification abilities than the four methods presented here; however, they would still likely benefit from the enhancements discussed in Section 4.

Of the state-of-the-art methods investigated, SPRINT can leverage multi-core processing to speed up computations. The other three model use GPUs to train their NNs. Since DPPI uses PSSMs as feature representations, some proteins which do not produce a valid PSSM from PSI-BLAST would need to be removed from the datasets.

2.5 Reciprocal Perspective

Reciprocal Perspective (RP) was recently developed by the cuBIC group and was presented in [13]. RP is a cascaded classifier approach that enhances PPI predictions by examining the score of a predicted protein pair within the context of all scores from all pairs involving either protein from the pair. This is conceptually related to the Reciprocal Best Hit method for finding orthologous genes using BLAST [81]. The general idea behind using RP is that the likelihood of a protein pair (e.g., A-B) interacting can be refined by examining the position of this pair's score (i.e., the predicted score of A-B, S_{AB}) within each of the protein's one-to-all (O2A) curves, where $O2A_B$ plots the rank-ordered prediction scores of all protein pairs involving protein B across the entire proteome (i.e., the set of all scores S_{iB} for $i \in$ all proteins). Features can be extracted from each of the two O2A curves that describe the relative position of the query pair's score, S_{AB}, relative to the curve. Additionally, since the "baseline" score for each protein differs, local thresholds can be established and used to identify high-scoring pairs instead of using a globally defined threshold. Thus, predictions are enhanced by accounting for local characteristics of each protein pair's score-ranking. This is further demonstrated below.

2.5.1 Description of RP Features

A score-ranking is obtained by making one-to-all interaction predictions for a given protein. For example, for a proteome with *n* proteins, each O2A curve will have *n* scores, sorted by score. The highest scoring interactor is given the first rank position followed by the next highest scorer ranked in the next position and so on for all predicted interactions. This results in a O2A curve that can be plotted as shown in Figure 8. In [13], a method

using locally weighted regression (LOESS) method is used to model the O2A curve. Then, the rank where a sharp change in slope occurs in the curve is used to define the "knee" of the curve. The score at the knee represents a local threshold (or baseline score) from which to derive RP features. The RP features defined in [13] are listed in Table 3 and were used to enhance PPI predictions in humans, mice, yeast, *A. thaliana*, and *C. elegans* using SPRINT [65] and PIPE [48]. The features are fed into a meta-classifier that produces the refined PPI prediction scores. Previous studies have used random forests or gradient-boosted decision trees for the meta-classifier [13], [75].

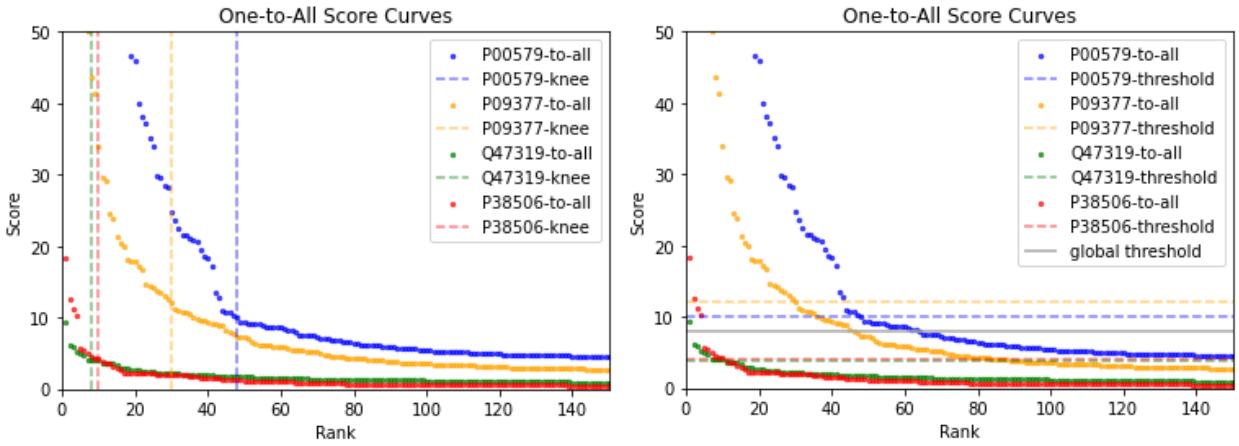

Figure 8: Examples of one-to-all curves with knees (left) and their respective score thresholds (right).

Table 3: RP Features Presented in [13]**.**

Feature Number	Description	Datatype, [Range of Values]
1	Rank of A in B's curve ($rankA_b$)	int, [1, |proteome|]
2	Rank of B in A's curve ($rankB_a$)	int, [1, |proteome|]
5	Naïve Reciprocal Rank Order (NaRRO) $1/(rankA_b \times rankB_a)$	float, [$1/(\|proteome_A\| \times \|proteome_B\|)$, 1]
6	Adjusted Reciprocal Rank Order (ARRO) $1/[(rankA_b/\|proteome_A\|) \times (rankB_a/\|proteome_B\|)]$; for interspecies case, $\|proteome_A\| != \|proteome_B\|$	float, [1, $\|proteome_A\| \times \|proteome_B\|$]
7	Normalized Reciprocal Rank Order (NoRRO) $1/[(rankA_b/\|proteome\|) \times (rankB_a/\|proteome\|)]$ for intraspecies case, $\|proteome_A\| == \|proteome_B\|$	float, [1, |proteome|]
8	$NoRRO_A$ $1/(rankA_b/\|proteome_A\|)$	float, [1, $\|proteome_A\|$]
9	$NoRRO_B$ $1/(rankB_a/\|proteome_B\|)$	float, [1, $\|proteome_B\|$]
10	Rank of knee in A's curve	int, [1, |proteome|]
11	Rank of knee in B's curve	int, [1, |proteome|]
12	Score of knee in A's curve	float, [0, maximum score]
13	Score of knee in B's curve	float, [0, maximum score]
18	Rank of B on A's curve is above (left of) knee	bool, [0 or 1]
19	Rank of A on B's curve is above (left of) knee	bool, [0 or 1]
24	Fold Difference A knee (score of B in A's curve from knee normalized)	float, [-1, 1]
25	Fold Difference B knee (score of A in B's curve from knee normalized)	float, [-1, 1]

3 Evaluation of State-of-the-Art

This section explains the methods used in this thesis to achieve the goal of independently assessing performance of state-of-the-art predictors for bacterial PPI prediction. Section 3.1 describes the evaluation protocols used for assessment. Section 3.2 provides details about datasets used in evaluations. Section 3.3 to Section 3.6 presents evaluation results of the four predictors. Finally, Section 3.7 contains a discussion to conclude these results.

3.1 Evaluation Procedures

This section describes the protocols used to perform a rigorous and fair evaluation of each predictor. The following subsections illustrate schemes used to evaluate classifiers, introduce comprehensive metrics defined to quantify performance, and describe the statistical methods used to compare results.

3.1.1 Experiment Design Protocols

The experimental design used to estimate the performance of classification models varies widely among PPI prediction studies. Most commonly, a k-fold cross-validation (k-CV) has been used to estimate how well a PPI model performs. Cross-validation can be considered an appropriate approach in this application because it provides a balance between resource consumption for running tests and obtaining an adequate estimation of a trained model's performance. For example, a leave-one-out design can provide a more accurate performance estimation of a model trained using nearly the entire dataset but would require tens of thousands of runs for each model due to the size of the *E. coli* proteome. On the opposite end, a single hold-out train-test split would require the least

number of runs but provides insufficient information to assess performance since the trained model would only be evaluated on a single set of test samples. Therefore, splitting the dataset into k-fold subsets for use in k-CV is a practical approach for approximating an accurate evaluation of a model's performance and ability to generalize to unseen data. A nested cross-validation scheme with bootstrapping of samples for each fold of k-CV would provide even more robust evaluation but would increase computational runtime requirements. In this thesis, a single 10-fold cross-validation as seen in Figure 9 is performed, whereby the same train and test subset definitions are used for evaluating all models to ensure fair comparisons.

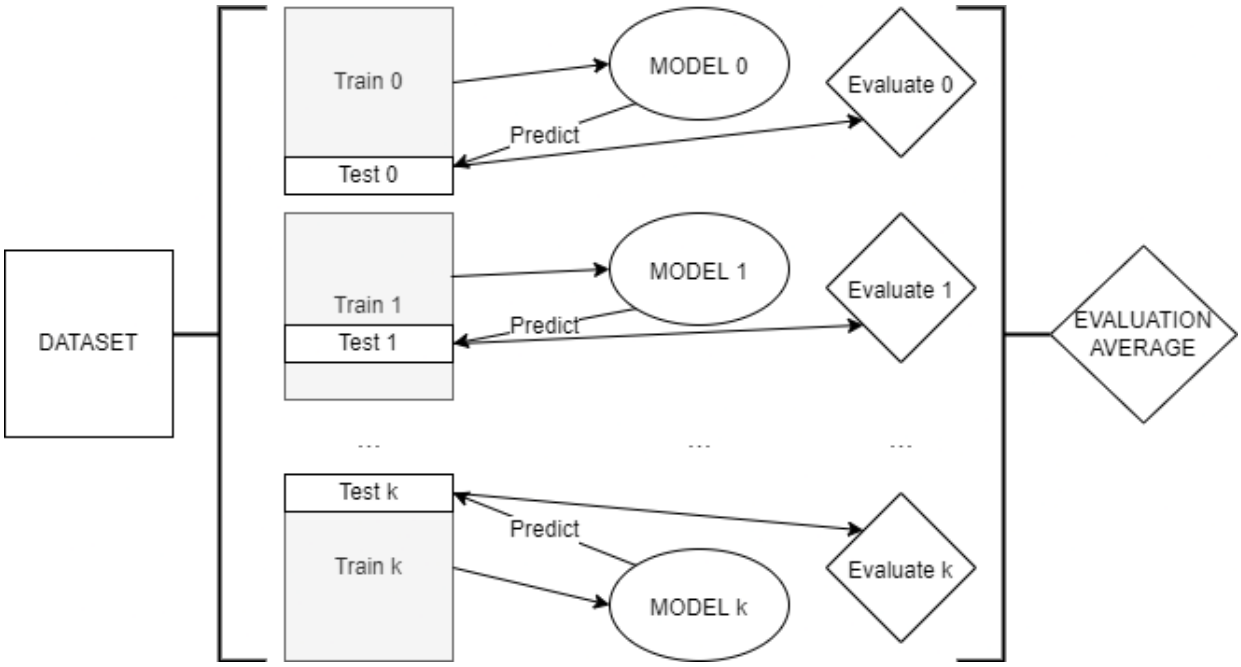

Figure 9: Cross-validation scheme for evaluation.

While it is clearly important that the same PPI not appear in both train and test subsets, the question remains of whether individual protein sequences should be present in both subsets. Predicting whether A interacts with B may be easier when the training set includes examples of PPI involving proteins A and B (separately). Park and Marcotte

(P&M) have proposed [77] a rigorous evaluation scheme to explore this issue. P&M discussed that traditional cross-validation may overestimate how well a model would generalize to predicting interactions containing proteins not found in the training data. They present an approach whereby the data are first split into training and testing sets (this would happen *k* times for a k-CV test) seen in Figure 10. Furthermore, the test data are split into three test subsets defined as C1, C2, and C3. Subset C1 contains pairs where both proteins in each pair are found in other PPI with the training set; C2 contains pairs where only one protein in each pair is found in the training set; and C3 contains pairs where neither protein is found in the training set. Therefore, it follows that the number of available pairs in each test subset decreases (|C1| > |C2| > |C3|). This approach provides an additional means to estimate the generalizability of a model using the same dataset besides providing a model with separate training and test datasets. In this thesis, P&M sets are constructed by randomly selecting a balanced 70 percent of the dataset for training and extracting the 3 test sets from the remaining data, also keeping the test sets balanced by generating NIPs if needed. This is performed 10 times to obtain 10 P&M sets (training, test C1, test C2, test C3) containing different pairs of proteins.

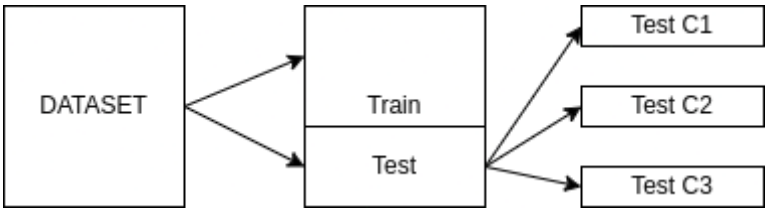

Figure 10: Park and Marcotte subset partitioning to evaluate the effects of prevalence of proteins in test sets.

3.1.2 Performance Evaluation Protocols

As for quantifying performance of PPI predictions, many publications do not adhere to a consistent set of metrics to evaluate their models in the same way. In certain contexts, it

can be sensible to evaluate a model's results by different means; however, the variability in reported metrics can make fair and direct comparison of models across reported studies difficult or even nonsensical. For example, using the accuracy of a model's predictions on one dataset while examining a precision-recall curve for predictions of another dataset does not enable any comparison to be made between the effect of datasets on a model's performance. Additionally, accuracy alone does not provide enough information to assess resulting predictions. For example, two models may produce the same value of accuracy, but one model could have correctly predicted all positive pairs and no negative pairs, while the other model could have correctly predicted all negative pairs and no positive pairs while identifying positive PPIs is more valuable in PPI prediction.

Consequently, providing a comprehensive set of metrics to evaluate predictions is necessary to gain greater perspective of a model's capabilities. Below lists six metrics and their calculations for binary classification which have been used inconsistently throughout PPI prediction publications. The true positive (TP), true negative (TN), false positive (FP), and false negative (FN) classifications are derived from classifying the PPI predictions of labelled pairs. It should be noted that a decision threshold must first be defined to classify predictions and obtain these metric values. Frequently, classifiers that output prediction probabilities between 0 and 1 (e.g., DPPI, DEEPFE, PIPR) use a default decision threshold value of 0.5. However, other models (e.g., SPRINT) that output a prediction score which is unbounded requires a more sophisticated choice of threshold value. Thus, the resulting value of each metric depends on the choice of decision threshold. For this reason, this thesis makes use of the precision-recall curve (PR curve)

and the receiver-operating characteristic curve (ROC curve) described below for evaluating prediction performance across all possible decision thresholds.

$$Recall = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$F_1 = \frac{2TP}{2TP + FP + FN}$$

$$MCC = \frac{(TP \times TN) - (FP \times FN)}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

An additional consideration that is required when evaluating PPI prediction is the practical implications for making a prediction. While a model may present good performance on a balanced dataset, confirming PPIs in the lab may not follow that same expectation. The ratio of interacting to non-interacting pairs among a set of proteins is greatly imbalanced as mentioned in Section 2.2.4 above. Evaluating a PPI predictor on a balanced test set ignores this reality and may dramatically over-estimate the proportion of positive predictions that will be true (i.e., the ratio of positive predictions that will be successfully validated in the wet lab).

Even though the train and test datasets may be balanced, one can estimate the performance on an imbalanced dataset. In this case, metrics can be recalculated to account for a theoretical imbalance. The calculation for the prevalence-corrected metrics is shown below and obtained from [82].

$$\delta = 2\left(\frac{NumberofPositives}{NumberofPositives + NumberofNegatives}\right) - 1$$

$$Recall = \frac{TP}{TP + FN} = \lambda_{PP}$$

$$Specificity = \frac{TN}{TN + FP} = \lambda_{NN}$$

$$Precision = \frac{\lambda_{PP}(1+\delta)}{\lambda_{PP}(1+\delta) + (1-\lambda_{NN})(1-\delta)}$$

$$Accuracy = \frac{1}{2}[\lambda_{PP}(1+\delta) + \lambda_{NN}(1-\delta)]$$

$$F_1 = \frac{2\lambda_{PP}(1+\delta)}{(1+\lambda_{PP})(1+\delta) + (1-\lambda_{NN})(1-\delta)}$$

$$MCC = \frac{\lambda_{PP} + \lambda_{NN} - 1}{\sqrt{\left[\lambda_{PP} + (1-\lambda_{NN})\frac{1-\delta}{1+\delta}\right]\left[\lambda_{NN} + (1-\lambda_{PP})\frac{1+\delta}{1-\delta}\right]}}$$

Mentioned above, PPI prediction models output a continuous value (often in the range [0,1]). Classifying a PPI score as interacting or not requires a decision threshold in making a binary decision. By calculating the above metrics using incremental decision thresholds between 0 and the maximum score, a more complete assessment of the behaviour for a model's predictions can be made. An ROC curve and PR curve are two common ways to provide this information. A ROC curve plots the true positive rate (recall) against the false positive rate (1 – specificity) at each decision threshold while the PR curve is self explanatory. The area under the curves (AUC) establishes a single value that can be used to interpret the overall classification performance. Examples in Figure 11 show these curves in the case of a perfect classifier versus a classifier with no predictive power. For a perfect classifier, the average precision, or auPR and the auROC

is equal to 1 whereas in a random classifier these values are 0.5, for balanced data. For imbalanced data, the auPR of a random classifier becomes the ratio of the imbalance in binary classification while the auROC is insensitive to class imbalance. Further details on evaluation metric calculations can be found in Appendix B.

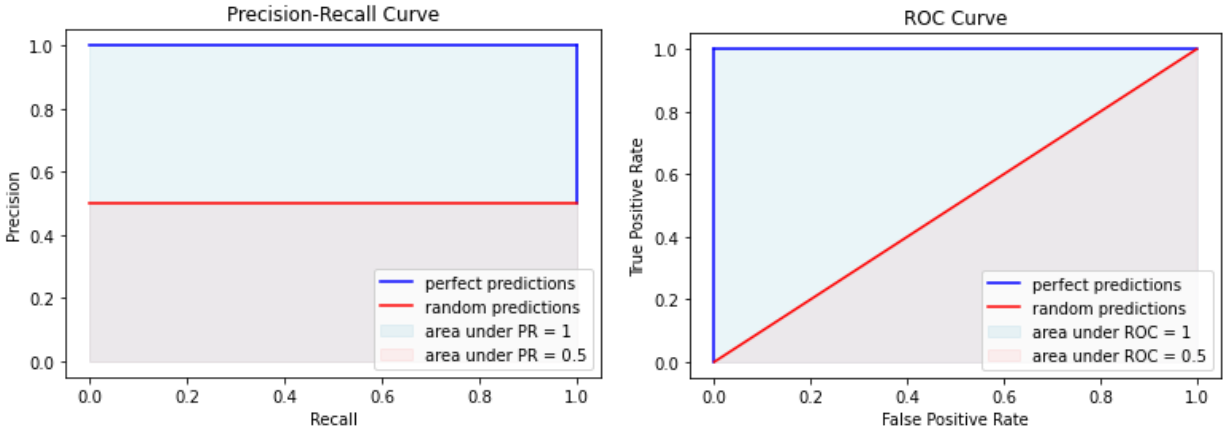

Figure 11: Illustration of PR and ROC curves best and worst-case classification performance.

In addressing the desire to correctly predict positive PPIs in the imbalanced space of possible protein interactions, PR curves can be more appropriate to use for performance evaluation. While an ROC curve provides similar information, it is often considered more optimistic when evaluating imbalanced data. As precision and recall focus on true positives which are the minority class in PPIs, PR curves are preferred to evaluate performance. In this thesis, both the auROC and auPR are presented for balanced and prevalence-corrected imbalanced (auPR only) prediction performance evaluation. The average metrics are computed from each k-fold predictions in cross-validations.

In addition to evaluating PPI prediction performance, overall computational performance can be investigated. Although deeper analysis can be performed on models, this thesis will simply consider the time and resources required to run the methods from

start to finish. This provides simple estimates for practical use in running these models. These resource utilization results can be found in Appendix A (Table 15).

3.1.3 Comparisons of Methods Protocols

After obtaining performance results from each method and using different datasets, quantitative comparisons must be made to determine which models outperform others. Since a mean and variance of metrics can be computed from the 10-CV evaluation experiments, significance testing using statistics can be applied. In this thesis, a one-way analysis of variance (ANOVA) and two-tailed t-test (paired when comparing methods, independent when comparing datasets) are used with alpha values of 0.05 and 0.01 to compare the auPR curves. The ANOVA test allows for many methods or datasets to be compared altogether and is used first to reduce the possibility of repeated pairwise t-tests producing false results. It determines if their performance (auPR) means vary sufficiently to be attributable to at least one method or dataset standing out significantly from the rest. The t-tests can then determine which method or dataset produces significantly different results from the others.

3.2 Datasets and Preprocessing

The following section combines ideas described in Section 2.2 to build high-quality datasets and investigates issues discovered in common benchmark datasets.

3.2.1 Algorithm for Building Datasets

Producing a high-quality dataset is the first step to allow computational models to be more accurately evaluated and provide more reliable predictions. By combining the merits of

the PPI dataset filtering approaches described in Section 2.2, an algorithm is presented here for creating datasets and is shown in Figure 12. This algorithm begins with raw data sourced from BioGRID to include the most recent interaction information available that has been curated and includes adequate PPI details for selection. Conservative filters used in POSITOME [47] are then applied to remove less reliably detected interactions. Then for both intra-species and inter-species interactions, only those with multiple sources of evidence are retained. Proteins involved in PPIs are then cross-referenced with UniProt to ensure sufficient protein information is available. PPIs containing proteins with no cross-referencing are removed. Redundancy in the remaining positive PPIs is removed by deleting pairs containing homologous proteins in the set of sequences using CD-HIT with a default identity threshold of 0.6. Finally, negative pairs are generated by random sampling of the available proteins left to finalize a balanced PPI dataset.

This algorithm was implemented in Python on Linux using a command-line interface and is available at the CUBIC GitHub https://github.com/GreenCUBIC/Bacterial-PPI-Prediction/tree/main/PREPROCESS. The program allows for optional preferences to tailor dataset creation based on the user's desires. This includes the ability to choose if conservative filters should be applied, choice to produce files of intra- and/or inter-species interactions, required level of confidence in PPIs, choice of including reviewed or unreviewed proteins, level of sequence homology reduction via CD-HIT, whether to consider subcellular locations when sampling for negative pairs, and formatting for use in prediction models. Table 4 lists the datasets used in this thesis of which the ECOLI, ECOLI_FULL, ECOLI_LARGE, ECOLI_SMALL, YEAST, and HUMAN datasets have been constructed using this preprocessing algorithm. Execution details are presented in

Table 5 summarizing the options used for each of these datasets. Runtime of the program is also shown for a typical laptop with an Intel i7-8550U processor and 12GB DDR4 memory.

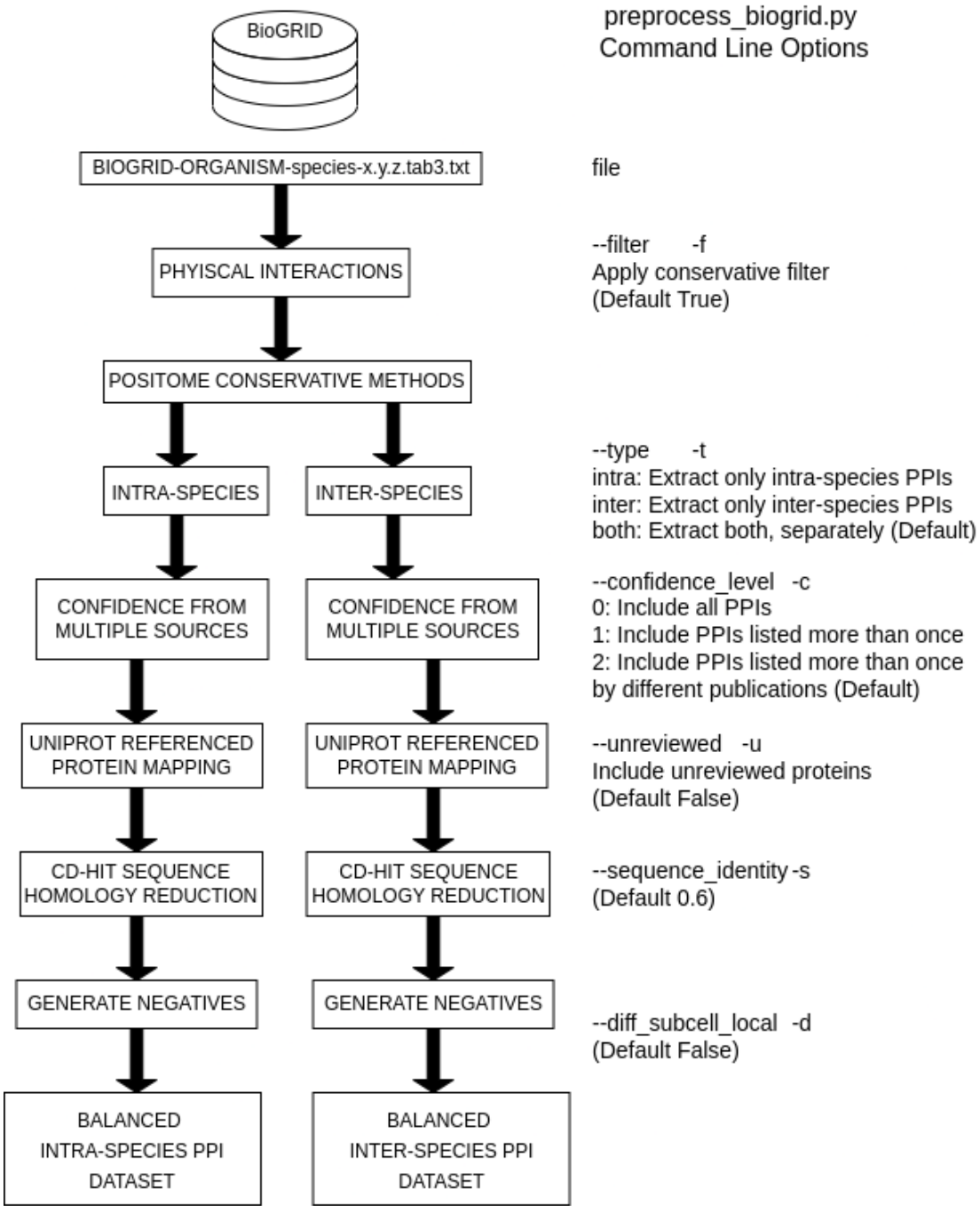

Figure 12: Algorithm for building PPI datasets.

Table 4: Datasets used in this thesis.

Dataset	Number of Positive PPIs	Number of Negative PPIs	Total PPIs	Number of Proteins
ECOLI	125	125	250	194
ECOLI_FULL	1,998	1,998	3,996	1,221
ECOLI_LARGE	1,875	1,875	3,750	1,219
ECOLI_SMALL	123	123	246	282
ECOLI_PROTEOME	n/a	n/a	9,850,141	4,438
DEEPFE_ECOLI	6,952	6,952	13,904	1,832
YEAST	14,944	14,944	29,888	3,722
YEAST_REDUCED	1,998	1,998	3,996	2,117
PIPR_YEAST	5,594	5,594	11,188	2,497
DEEPFE_YEAST	5,594	5,594	11,188	2,529
HUMAN	37,938	37,938	75,876	9,613
HUMAN_REDUCED	1,998	1,998	3,996	3,142

Each dataset produced from BioGRID here was collected from BIOGRID-ORGANISM Release 4.4.198, compiled on May 25, 2021. Newer releases were found to provide no significant additional PPIs for *E. coli*. For example, a newer version, Release 4.4.206 compiled January 25, 2022, would only produce 1 more PPI for the ECOLI dataset. Thus, this preprocessing algorithm may benefit from future BioGRID releases which may contain more PPI information; however, new data for *E. coli* has remained stagnant recently.

Note that for the ECOLI_FULL dataset, four BioGRID files for *E. coli* sub strains were first manually combined before running the automated script, unlike ECOLI which only used the K12-MG1655 strain file. This was performed to increase the quantity of PPIs available for use as few PPIs were available under level-2 and level-1 confidence conditions. Most PPIs retained were derived from the K12-MG1655 strain which were also retained in the ECOLI dataset. The two datasets ECOLI and ECOLI_FULL represent the level-2 and level-1 confidence of PPIs extracted from BioGRID respectively and serve as the main datasets from which ECOLI_LARGE and ECOLI_SMALL are built. The relationships between PPIs in these datasets are illustrated in Figure 13.

The ECOLI_LARGE and ECOLI_SMALL datasets were created to investigate the effect of quantity versus quality of datasets on predictor performance. ECOLI_SMALL contains all positive PPIs found in ECOLI with balanced NIPs found in ECOLI_FULL. ECOLI_LARGE contains all positive PPIs in ECOLI_FULL minus those in ECOLI and balanced NIPs from ECOLI_FULL, excluding those used in ECOLI_SMALL. Thus, ECOLI_SMALL and ECOLI_LARGE are analogous to ECOLI and ECOLI_FULL but share no PPIs between them.

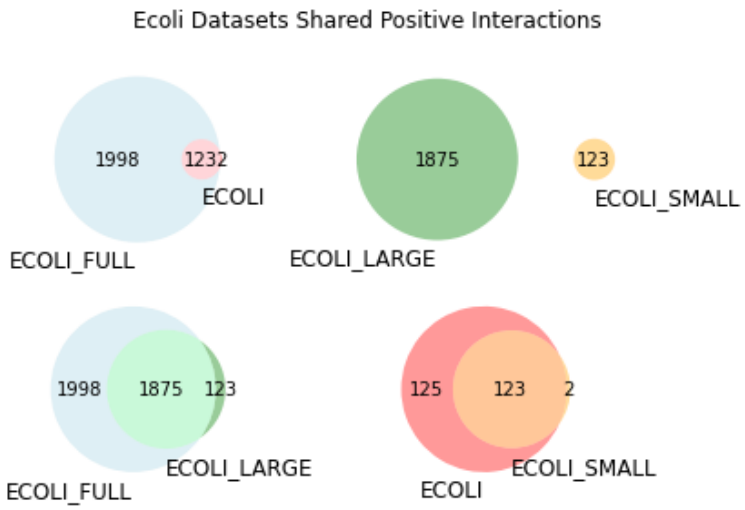

Figure 13: Relationships of number of shared PPIs in *E. coli* datasets.

The ECOLI_LARGE and ECOLI_SMALL datasets are used in "LARGESMALL" evaluations to indicate the effect of size and quality of training data on predictions. For this, the model is trained with the ECOLI_LARGE dataset to make predictions on the ECOLI_SMALL dataset. If the LARGESMALL results are similar to the ECOLI_FULL k-CV results, then this indicates that the predictor benefits from more training data. However, if the LARGESMALL results are similar to the ECOLI k-CV results, then this indicates that the predictor requires higher-quality training data.

Table 5: Dataset creation details.

Dataset	Conservative Filters	Confidence Level	Reviewed Proteins Only	Identity Threshold	Negatives Different Cellular Locations	Approx. Runtime (s)
ECOLI	Yes	1	Yes	0.6	No	8.44
ECOLI_FULL	Yes	0	Yes	0.6	No	17.19
YEAST	Yes	2	Yes	0.6	No	51.42
HUMAN	Yes	2	Yes	0.6	No	179.35

The P&M cross-validation datasets built from ECOLI_FULL are shown in Table 6. This shows the relative sizes for each *k*-fold training and C1, C2, C3 test sets. Noticeably as mentioned earlier, the sizes of C1, C2, and C3 decrease as there is a decreasing likelihood of finding proteins not found in the training pairs. Note that these values represent the total (positive + negative) number of protein pairs and are balanced sets.

Table 6: Park and Marcotte dataset sizes used in cross-validations.

k-Fold Set	Number of Training Pairs	Number of C1 Test Pairs	Number of C2 Test Pairs	Number of C3 Test Pairs
1	2,796	1,140	48	12
2	2,796	1,150	42	8
3	2,796	1,144	50	6
4	2,796	1,140	50	10
5	2,796	1,148	44	8
6	2,796	1,150	42	8
7	2,796	1,168	24	8
8	2,796	1,152	40	8
9	2,796	1,154	38	8
10	2,796	1,150	40	10
Average (rounded up)	2,796	1,150	42	9

Referring again to Table 4, DEEPFE_ECOLI is the DIP (circa 2011) dataset containing only positive pairs established by [46], but sourced from the GitHub repository provided by [73]. Negative pairs have been generated in this thesis by random sampling to build a balanced dataset for evaluating DEEPFE_ECOLI.

The PIPR_YEAST and DEEPFE_YEAST datasets are the benchmark DIP (version DIP_20070219) datasets widely used and established by [83], but have been sourced from the GitHub repos provided by [74] and [73], respectively. These are used in this thesis to validate local implementations of each predictor. Further investigation of these datasets is presented in the following section and displays the need for more current and robust data preprocessing for use in computational PPI prediction.

The YEAST and HUMAN datasets were compiled as independent validations of local implementations for predictors. The yeast and human PPIs were also extracted from BIOGRID-ORGANISM Release 4.4.198 using the files for the *Saccharomyces cerevisiae* S288c strain and homo sapiens, respectively. In addition, the YEAST_REDUCED and HUMAN_REDUCED dataset are subsets of YEAST and HUMAN with the same dataset size as ECOLI_FULL to evaluate the effects of quantity of pairs versus organism for prediction performance. The next subsection also makes some comparisons between the independent YEAST dataset and the benchmark yeast datasets.

3.2.2 Comparisons of Benchmark Datasets

Previously mentioned, many studies have used “benchmark” datasets in an effort to directly compare themselves with previous methods. However, our initial analysis identified that the “same” benchmark dataset used in two different studies were not actually identical; in some cases, they are completely dissimilar. An analysis of protein pairs among the datasets was investigated to decipher commonalities and differences between them. The data used for sequence-based predictors are typically represented in FASTA format whereby the protein identifier is followed by its amino acid sequence. For

example, a protein ID could be P12345 and have the sequence ABCDEF, so a PPI dataset can be viewed by pairing either protein IDs or pairing sequences. Ideally, each protein ID and sequence is unique in the dataset; however, “different” proteins have been found to share IDs or sequences. Therefore, the following figures display this dataset analysis using the PPIs by their UniProt protein IDs (left images) and by their respective sequences (right images). Exact sequences were used for comparisons between datasets.

First, a comparison between the PIPR_YEAST, DEEPFE_YEAST, and the BioGRID YEAST datasets can be seen in Figure 14 to Figure 16. For each dataset, duplicated proteins and pairs were removed to compare only unique instances. Considering that the PIPR and DEEPFE yeast datasets reference the same source, one would expect them to be identical. However, Figure 14 shows almost no overlap between the shared interactions using either the IDs or sequences. More importantly, Figure 15 shows no shared positive PPIs between the two sets. This is a shocking finding, considering that both datasets are claimed to represent the original benchmark yeast PPI dataset defined in [83]. Figure 16 and Figure 17 indicate that most of the proteins within the datasets are common, suggesting that the labelled pairing of proteins is the main discrepancy between PIPR_YEAST and DEEPFE_YEAST.

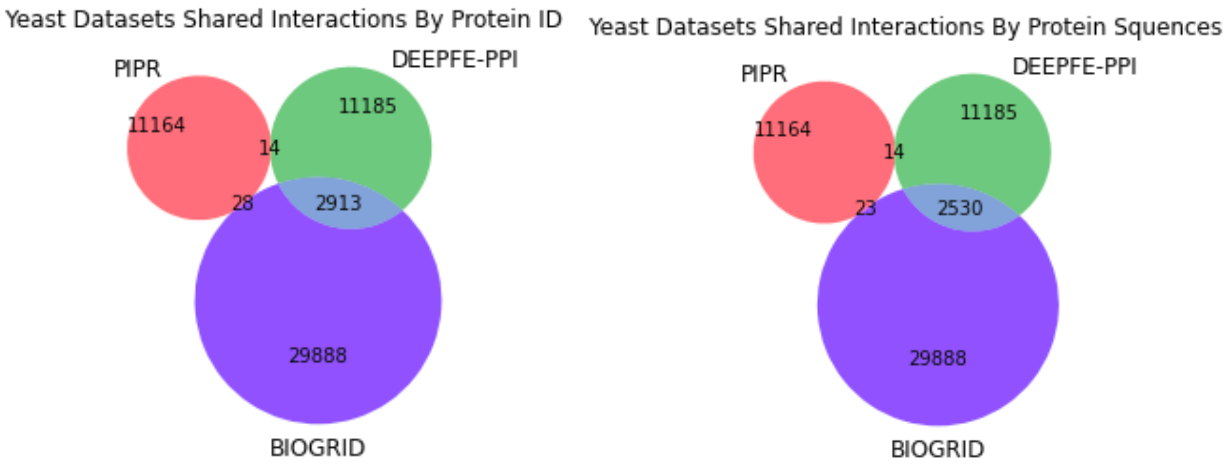

Figure 14: Comparison of yeast dataset protein pairs using given protein IDs (left) and their respective mapped sequences (right).

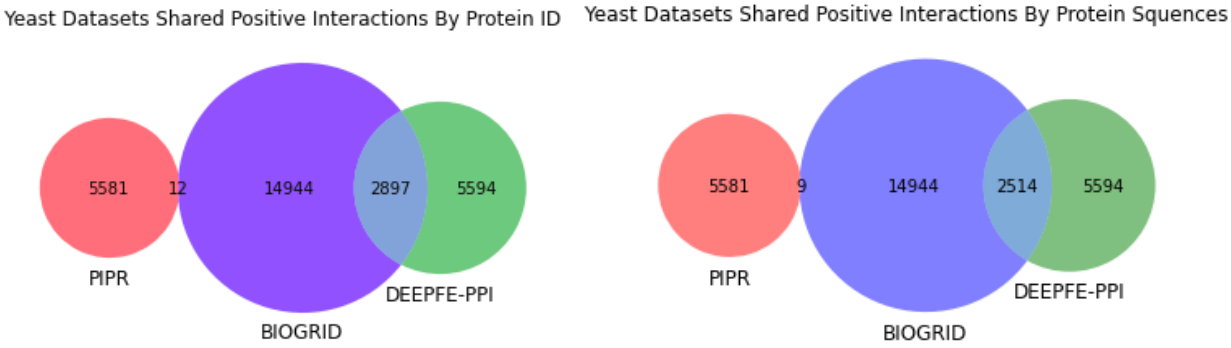

Figure 15: Comparison of yeast dataset positive PPIs using given protein IDs (left) and their respective mapped sequences (right).

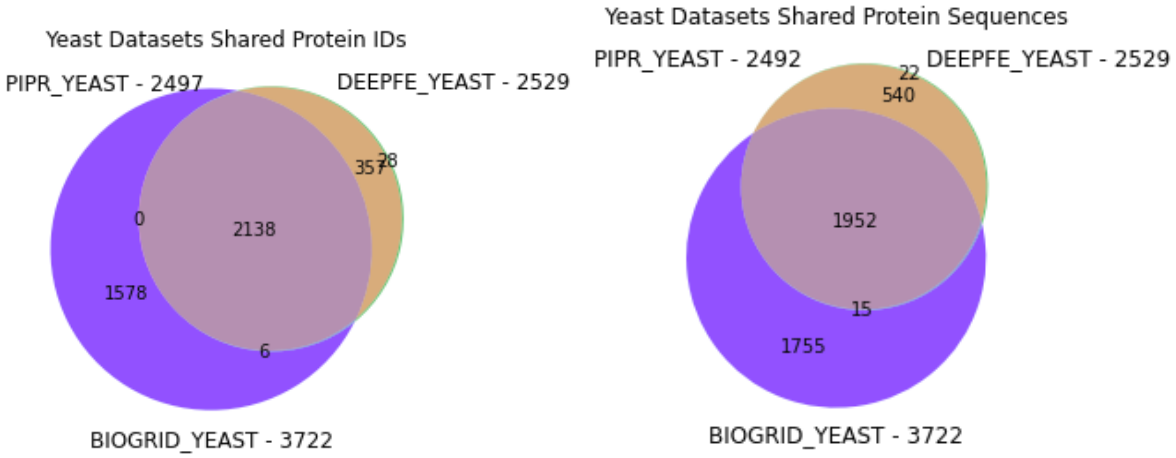

Figure 16: Overlaps of yeast protein IDs (left) and sequences (right) within each dataset.

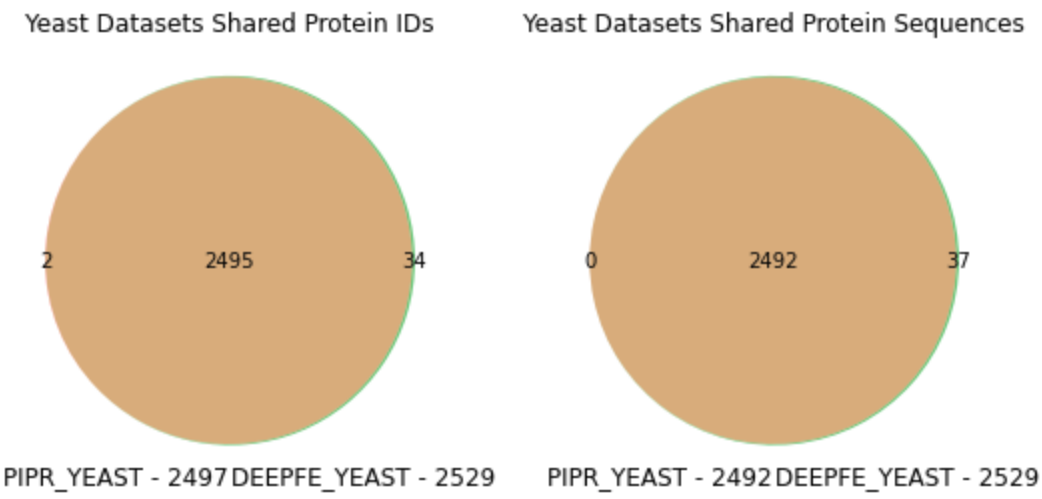

Figure 17: Overlap of yeast protein IDs (left) and their respective mapped sequences (right) within each benchmark dataset.

Secondly, a comparison between ECOLI, ECOLI_FULL, and the benchmark DEEPFE_ECOLI datasets is illustrated in Figure 18 below. As expected, most proteins and PPIs found between ECOLI and ECOLI_FULL were shared except for a couple which were removed from ECOLI_FULL during the homology reduction step as one protein was essentially a subsequence of a longer sequence protein within the dataset. However, the DEEPFE_ECOLI dataset contained much smaller intersection with the other datasets. Additionally, protein sequences were not completely overlapping between DEEPFE and ECOLI_FULL dataset. This could be partially due to comparing exact sequences and could be further analyzed by considering trimmed or sequence identity comparisons. Also, 853 protein IDs and 177 sequences in DEEPFE_ECOLI were not within the reviewed SwissProt database. This can be seen in Figure 19 in comparing the proteins of each dataset in the *E. coli* proteome.

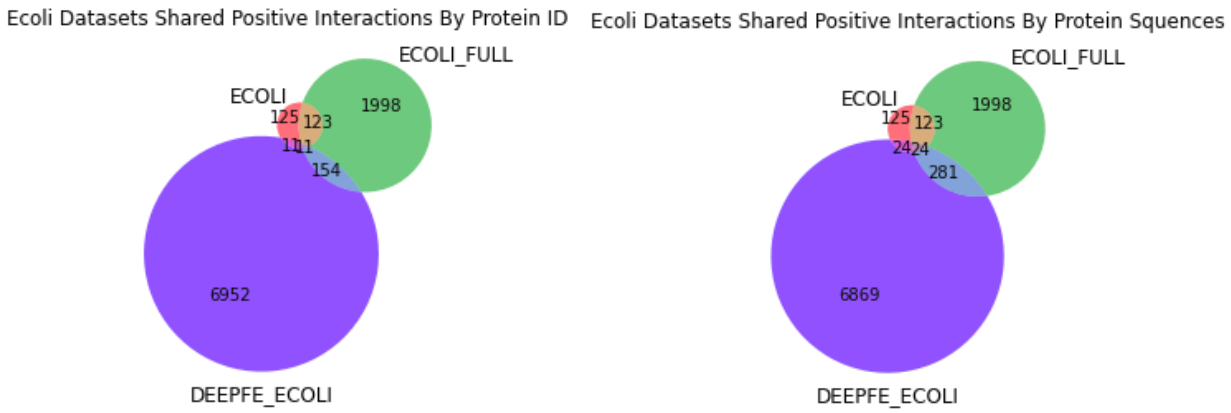

Figure 18: Comparison of *E. coli* dataset protein pairs using protein IDs (left) and their respective mapped sequences (right).

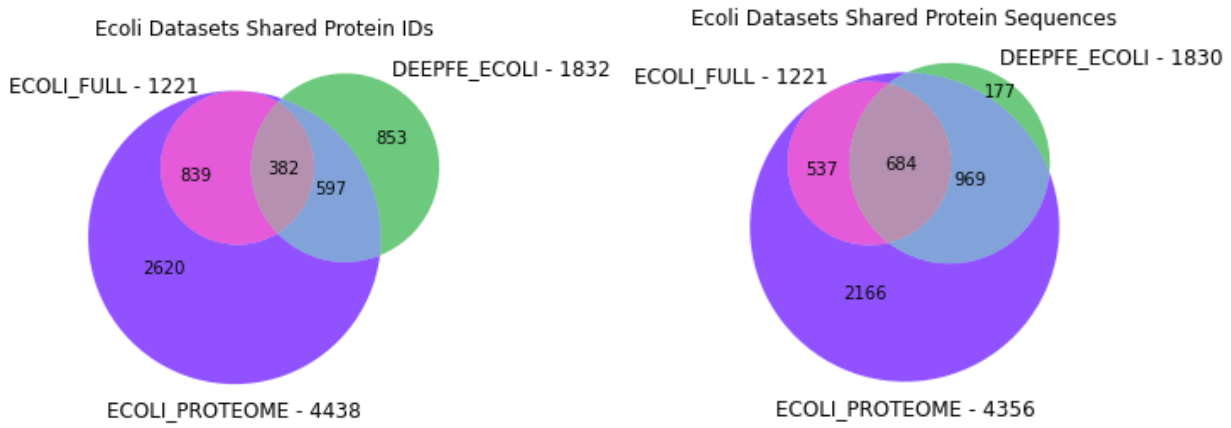

Figure 19: Overlap of *E. coli* protein IDs (left) and sequences (right) within each dataset.

Finally, a visualization of the distribution of sequence lengths for proteins used in the datasets created here and the *E. coli* proteome is shown in Figure 20. This indicates that datasets constructed in this thesis comprise proteins that are a fair representation of the entire proteome. Therefore, providing computational models with these datasets for training should allow sufficient variability in sequences to more confidently predict the entire interactome.

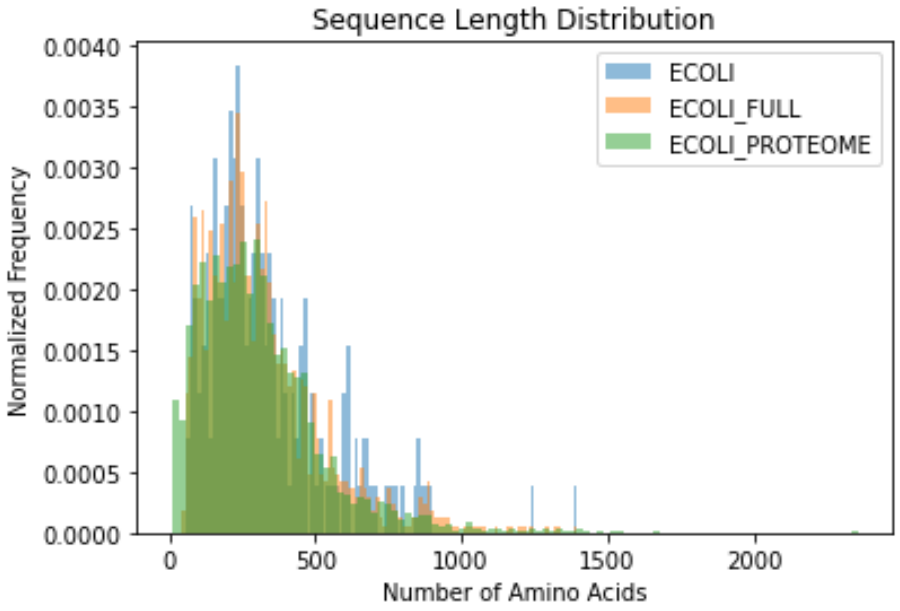

Figure 20: Comparison of protein sequence lengths for *E. coli* datasets with proteome.

3.2.3 Conclusion

Although a more in-depth assessment on these datasets can be investigated, the illustrations above provide clear justification for creating and utilizing independently constructed datasets in this thesis. Despite benchmark datasets being widely used, there is an apparent lack of conformity to their established sources. Additionally, these benchmarks may not contain reliable information for training prediction models. If we are to assess the state of the art systematically and objectively in PPI prediction, we must carefully create independent PPI datasets.

In summary of the sections above, the following conclusions can be drawn:

1. The growth of available PPI data for *E. coli* has become stagnant in recent years with a small number of research groups contributing most of the known PPI data.
2. PPI datasets can benefit from careful preprocessing techniques to increase data quality.

3. Widely used benchmark PPI datasets are outdated and may not accurately represent the current knowledgebase of PPIs.
4. Discrepancies exist in benchmark datasets used in state-of-the-art classifier evaluations, which may cause misconstrued results.
5. The algorithm presented here can rectify some of the problems above by producing updated, reliable, and high-quality datasets.
6. The datasets compiled here can be used as new benchmarks in future studies of state-of-the-art methods.

3.3 SPRINT Performance

This section uses the datasets and performance evaluation methodologies developed in the previous section to systematically evaluate SPRINT, the first of four methods representing the state-of-the-art in sequence-based PPI prediction.

3.3.1 Repeatability of Claims

Although this thesis seeks to investigate PPI prediction for bacteria, first a comparison to the author's originally published results was independently evaluated and presented below using the HUMAN dataset derived as explained Section 3.2.1 and a cross-validation scheme. This was performed to independently assess the consistency of implementing SPRINT locally. In the original SPRINT publication, a BioGRID dataset of 215,029 PPIs was partitioned into P&M subsets containing 100,000 training pairs and 10,000 testing pairs each. It is presumed that the testing pairs contained equal positive and negative samples. They were evaluated with auROC and auPR producing an average over each C1, C2, C3 sets of 0.8415 and 0.8538, respectively.

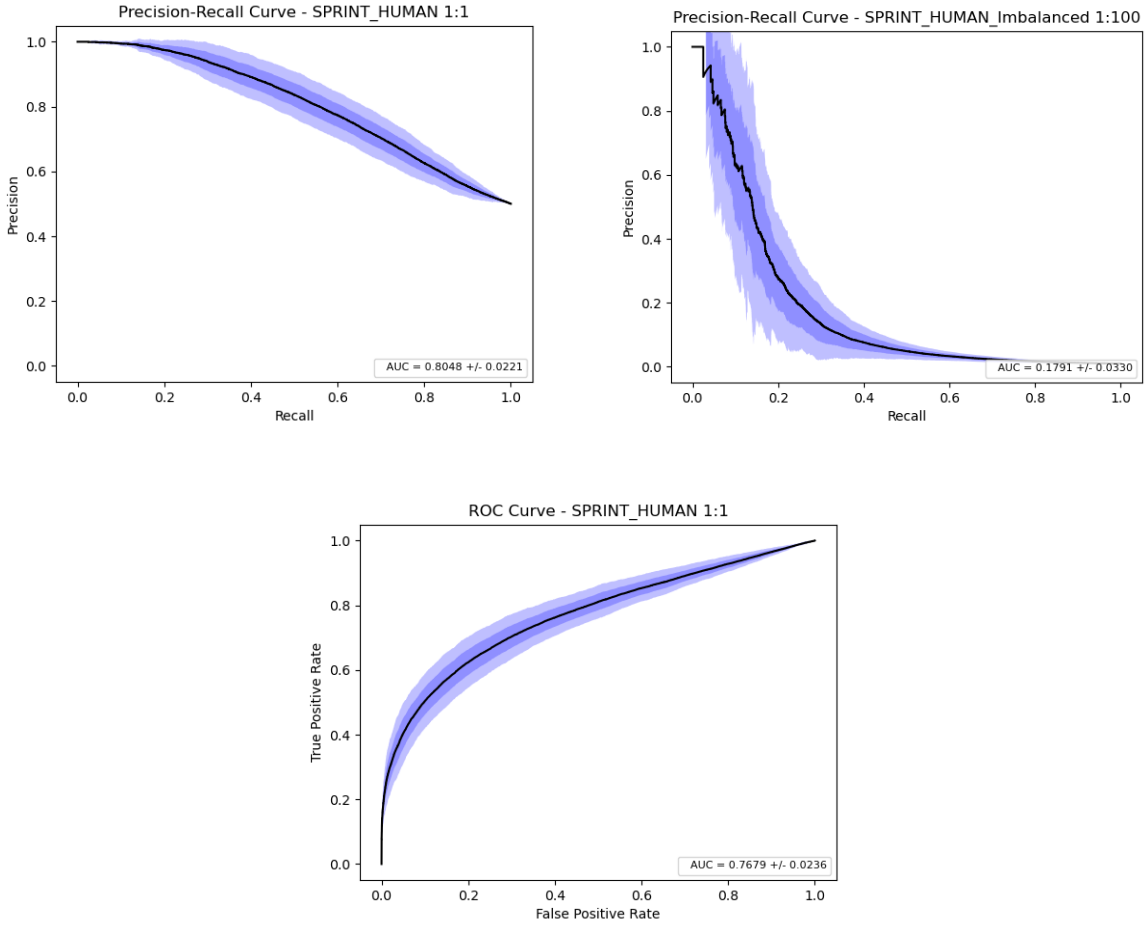

Figure 21: Repeatability of SPRINT using 10-CV on the HUMAN dataset.

Figure 21 displays results of this independent evaluation; note from here on that the innermost coloured area is one standard deviation from the mean and the outermost coloured area is two standard deviations. The auPR for balanced and 1:100 class-imbalanced predictions were 0.8048 ± 0.0221 and 0.1791 ± 0.0330, respectively. The auROC was 0.7679 ± 0.0236. Here, a 10-CV was performed for evaluation due to P&M subset creation not being possible. This was due to much fewer proteins among PPIs being available for partitioning, meaning it was difficult to obtain test sets with proteins not found in the training set. However, comparing these results to the above auPR (0.8538)

and auROC (0.8415) averages shows a small performance drop here likely attributable to less training data (e.g., about 7,587 samples here versus 10,000 samples, per *k*-fold).

3.3.2 Investigation of Cross-Species Predictions

A test was performed whereby SPRINT was trained using positive HUMAN pairs and tested on ECOLI_FULL pairs. This was to explore any potential that SPRINT's performance on the HUMAN dataset may translate to predicting *E. coli* PPIs after training on human PPIs. These results are presented in Figure 22 and clearly display no such potential.

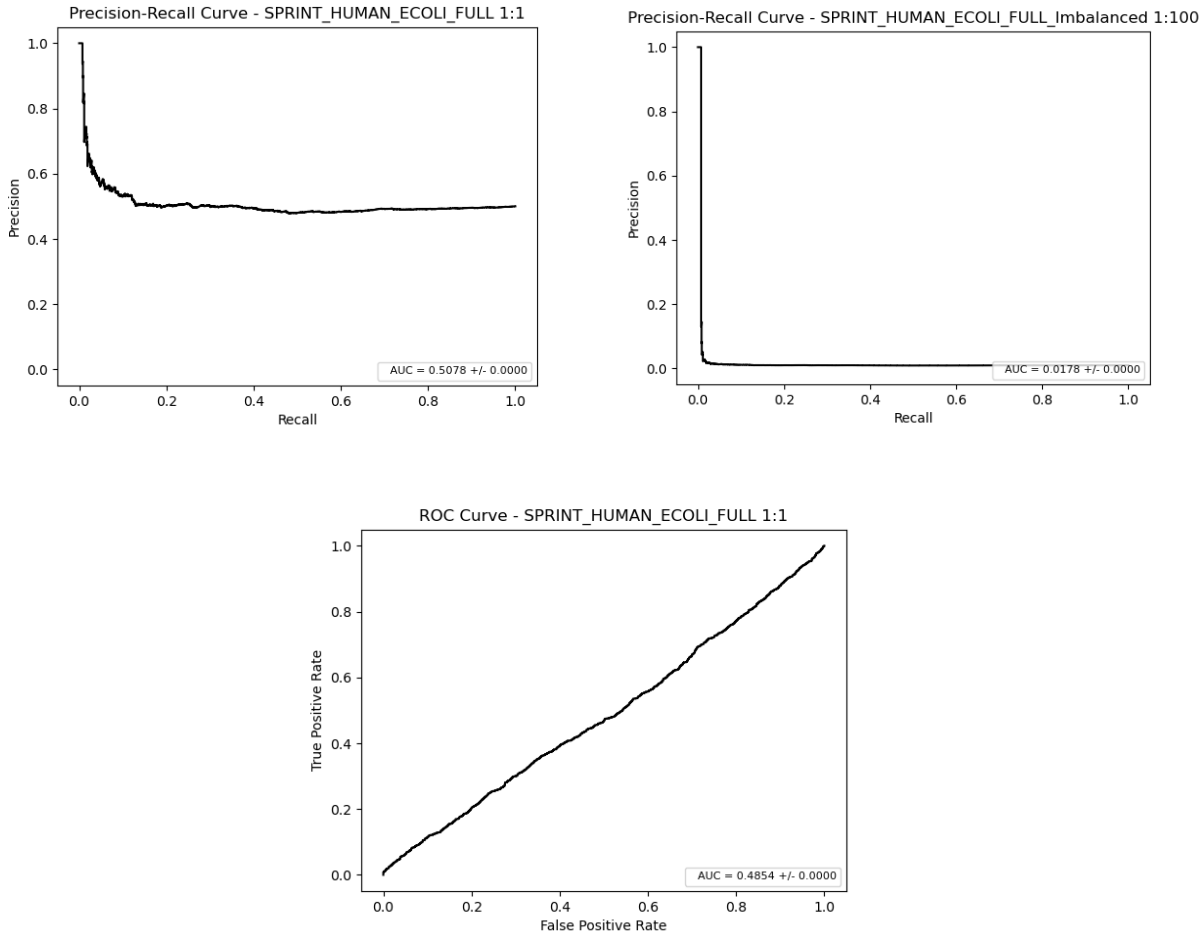

Figure 22: Investigation of cross-species predictions using SPRINT for *E. coli*.

These results indicate that SPRINT is unable to learn from human PPIs to predict *E. coli* PPIs. This is not an unexpected result as eukaryotic proteins may greatly differ from prokaryotic ones. Therefore, orthologous sequences may be lacking between proteomes in these datasets and thus very few HSPs may exist to increase scores for interactions.

3.3.3 Cross-Validation on *E. coli*

One explanation for why PPI prediction methods have historically under-performed on bacterial interactomes is that methods tend to be trained on human or yeast. To test this theory, each of the four methods explored in this thesis are systematically retrained and evaluated specifically on *E. coli.*

SPRINT is evaluated here to determine if it can be retrained using *E. coli* data to increase performance for this species. A 10-CV was used for both ECOLI and ECOLI_FULL datasets as well as a LARGESMALL evaluation described in Section 3.2.1. Figure 23 below shows results of both cross-validations and the LARGESMALL evaluation. The auPR for 10-CV on ECOLI and ECOLI_FULL under balanced evaluations are 0.4660 ± 0.1041 and 0.6277 ± 0.0280, respectively and class-imbalanced evaluation results in auPR of 0.0091 ± 0.0764 and 0.0243 ± 0.0389, respectively. The auPR for the LARGESMALL test under balanced and imbalanced evaluations are 0.6354 and 0.0226 (similar performance as ECOLI_FULL CV). Likewise, the auROC for ECOLI and ECOLI_FULL CV resulted in 0.4895 ± 0.1408 and 0.5979 ± 0.0208, respectively, with the LARGESMALL test resulting in an auROC of 0.5931.

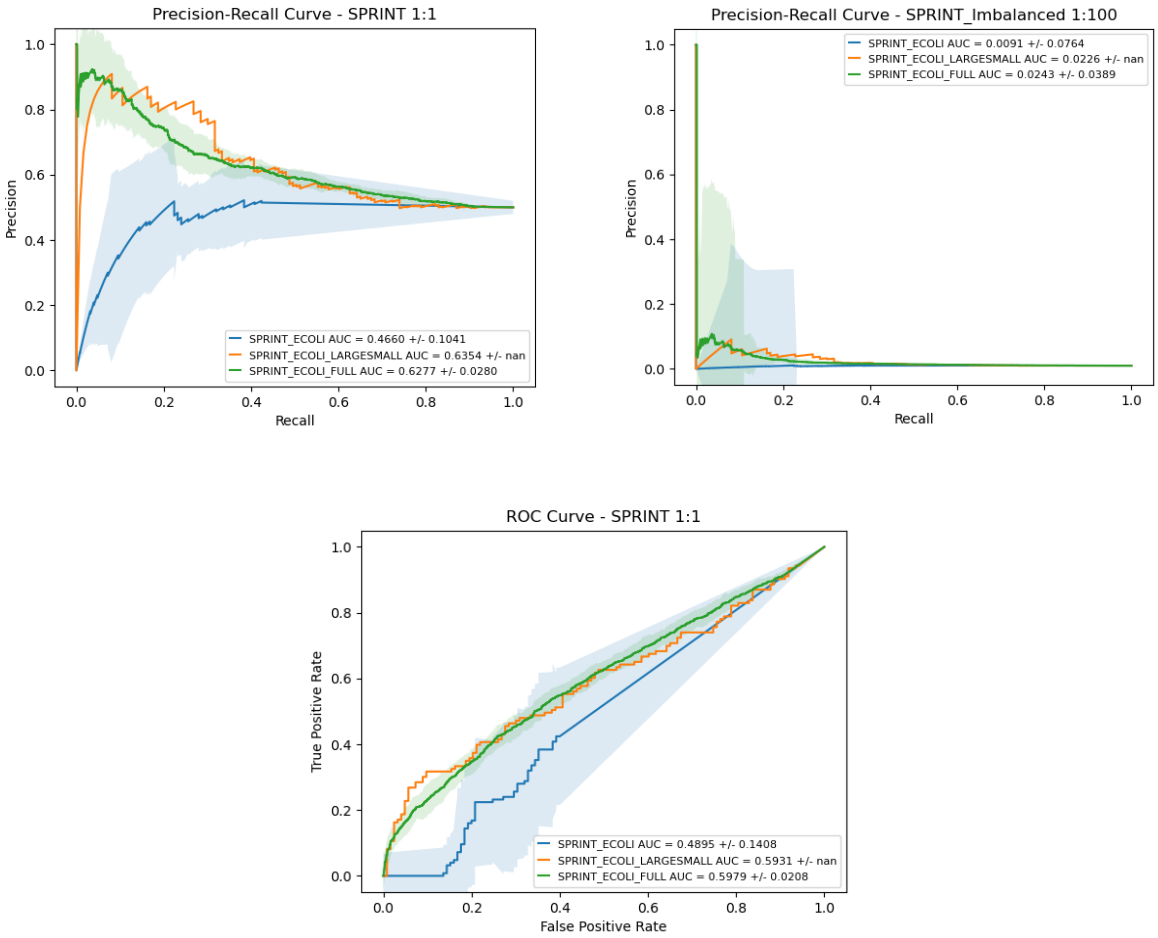

Figure 23: Evaluation of SPRINT on *E. coli* datasets using 10-fold cross-validation and a LARGESMALL evaluation scheme.

As expected, SPRINT benefits greatly with more *E. coli* training data. Note from here and throughout the thesis, a significant sudden drop in precision at low recall in the PR curve is simply due the lack of TPs and FPs at high decision thresholds for binary classification. For example, a very high decision threshold may yield no pairs classified as positive, so precision and recall start at zero. However, as the decision threshold is lowered, more pairs may be classified as positive. If more pairs become classified as false positives than those for true positives, precision will drop as recall increases.

3.3.4 Park & Marcotte Evaluation on *E. coli*

As described in Section 3.1, the P&M evaluation scheme examines the decreasing PPI prediction performance as we progressively constrain the test set to be increasingly dissimilar to the training set. The results of P&M evaluation when training and testing SPRINT on *E. coli* are shown in Figure 24. The auPR for C1, C2, and C3 test sets were 0.6563 ± 0.0077, 0.6175 ± 0.0798, and 0.5905 ± 0.1021, respectively under balanced evaluation. For class-imbalanced evaluation, the corresponding auPR values drop to 0.0257 ± 0.0153, 0.0706 ± 0.0923, and 0.0139 ± 0.1745. The auROC for C1, C2, and C3 test sets was 0.6247 ± 0.0040, 0.5760 ± 0.0875, and 0.6522 ± 0.1432, respectively.

Despite computing HSPs for all 1221 proteins prior to training SPRINT, higher PPI scoring is given for pairs with proteins found in the training data as the HSPs from these proteins are given greater weight for scoring. Consequently, the reduced performance on C3 test sets reflects the prediction of new PPIs for proteins for which there is no known experimental PPI data. On the opposite end, the performance on C1 test sets is reflective of the case where we are predicting new PPIs for which both proteins have known experimental PPI in the training data.

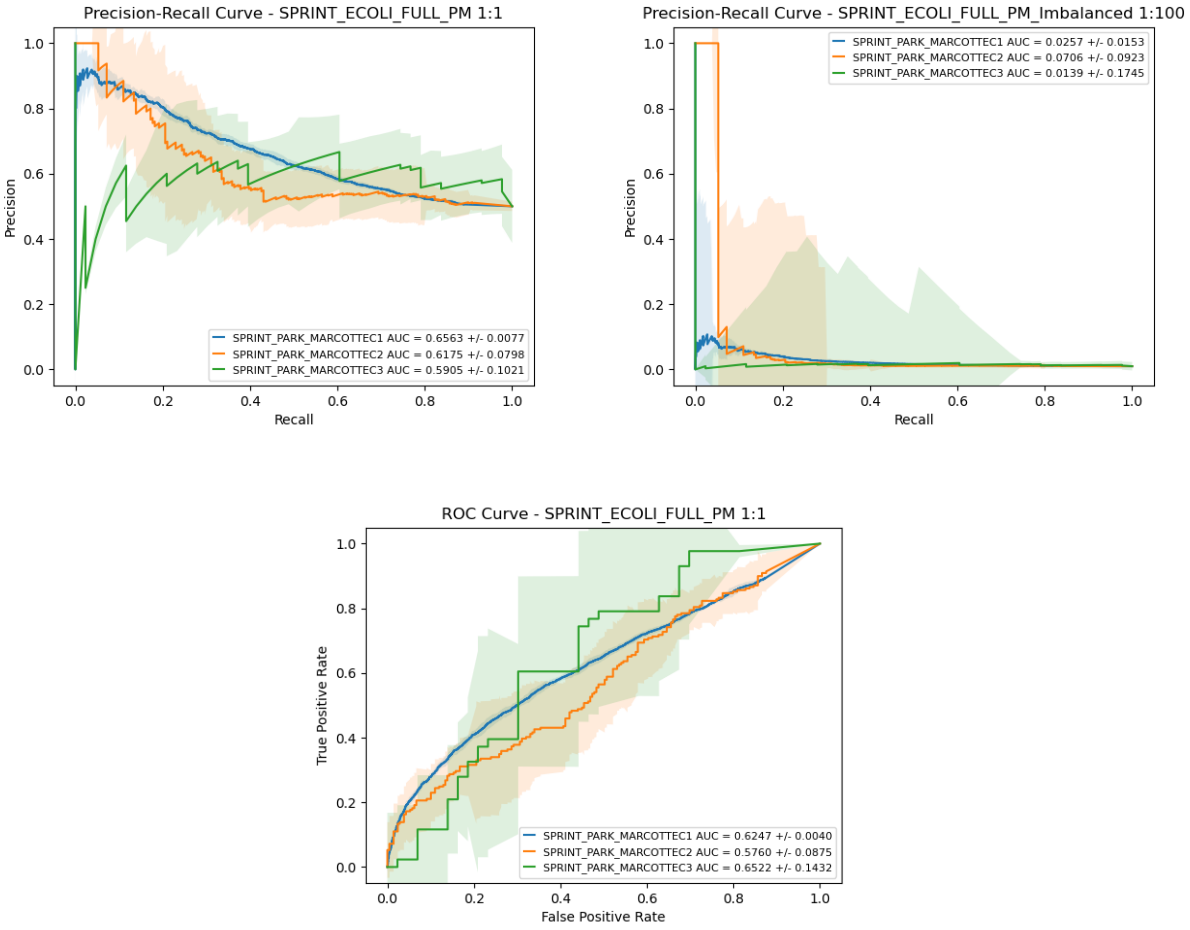

Figure 24: Park and Marcotte evaluation on *E. coli* datasets using SPRINT.

3.3.5 Comparison Between Species

Figure 25 below illustrates the difference in SPRINT's ability to predict intra-species PPIs for humans, yeast, and *E. coli*. In this evaluation, a 10-CV was performed for each dataset. Noticeably, SPRINT's predictions for *E. coli* are far less accurate than for yeast and human PPIs. The auPR for ECOLI_FULL, YEAST, and HUMAN were 0.6277 ± 0.0280, 0.8266 ± 0.0308, and 0.8048 ± 0.0233, respectively on balanced evaluations. For class-imbalanced evaluations, the auPR was 0.0243 ± 0.0389, 0.1684 ± 0.0440, and

0.1791 ± 0.0348. The auROC for ECOLI_FULL, YEAST, and HUMAN was 0.5979 ± 0.0208, 0.7939 ± 0.0402, and 0.7679 ± 0.0249, respectively.

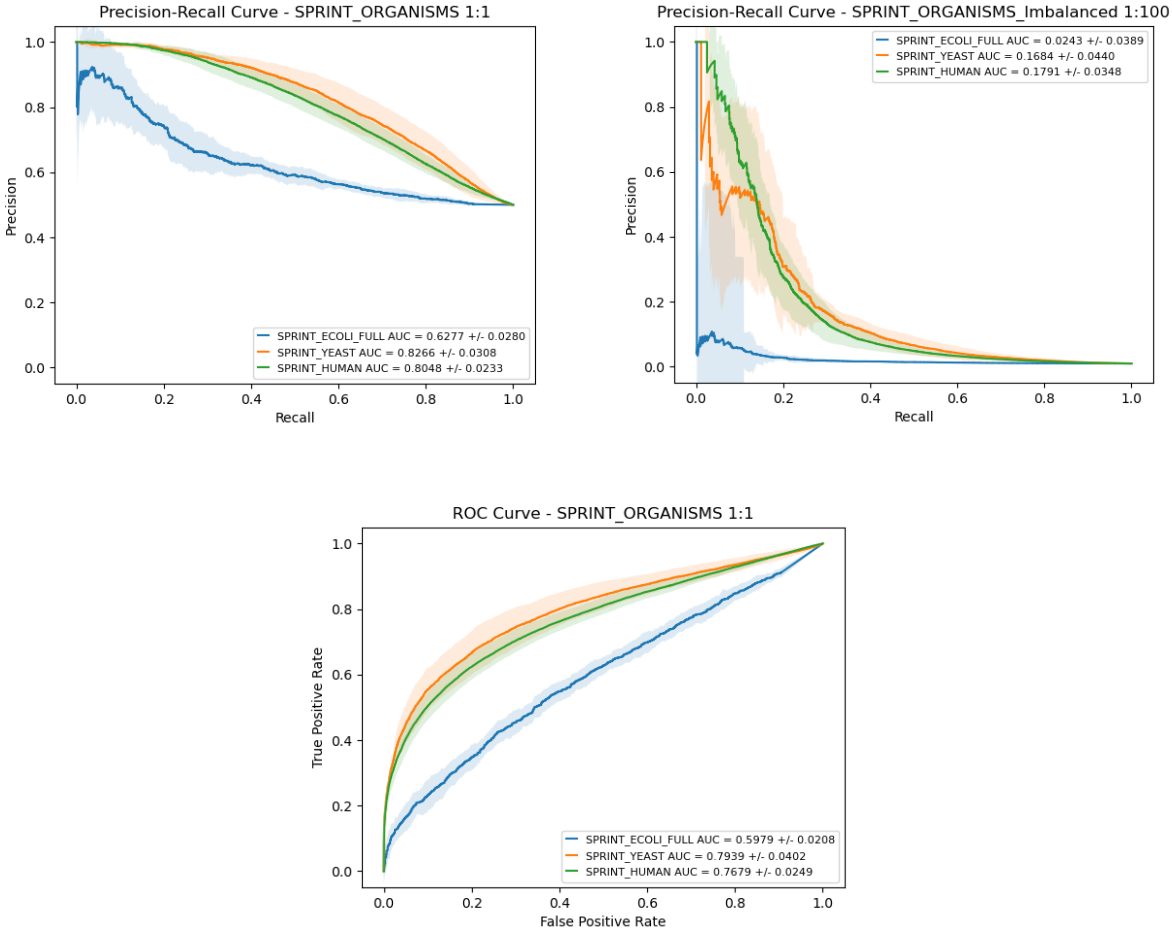

Figure 25: Performance of SPRINT on HUMAN, YEAST, and ECOLI_FULL datasets.

The above results could be due to biological differences in each species' sequences that affect computing HSPs and scoring but is more likely due to dataset sizes as indicated by the results in Figure 26. To explore this, both the HUMAN and YEAST datasets were artificially reduced to balanced datasets equal in size to ECOLI_FULL. The same evaluation was performed, and the comparisons can be seen below. The auPR for YEAST_REDUCED and HUMAN_REDUCED changed to 0.7326 ± 0.0280 and 0.6226 ±

0.0495, respectively on balanced evaluations. For class-imbalanced evaluations, the auPR became 0.0606 ± 0.0967 and 0.0655 ± 0.0479. The auROC for YEAST_REDUCED and HUMAN_REDUCED changed to 0.7041 ± 0.0664 and 0.5885 ± 0.0582, respectively.

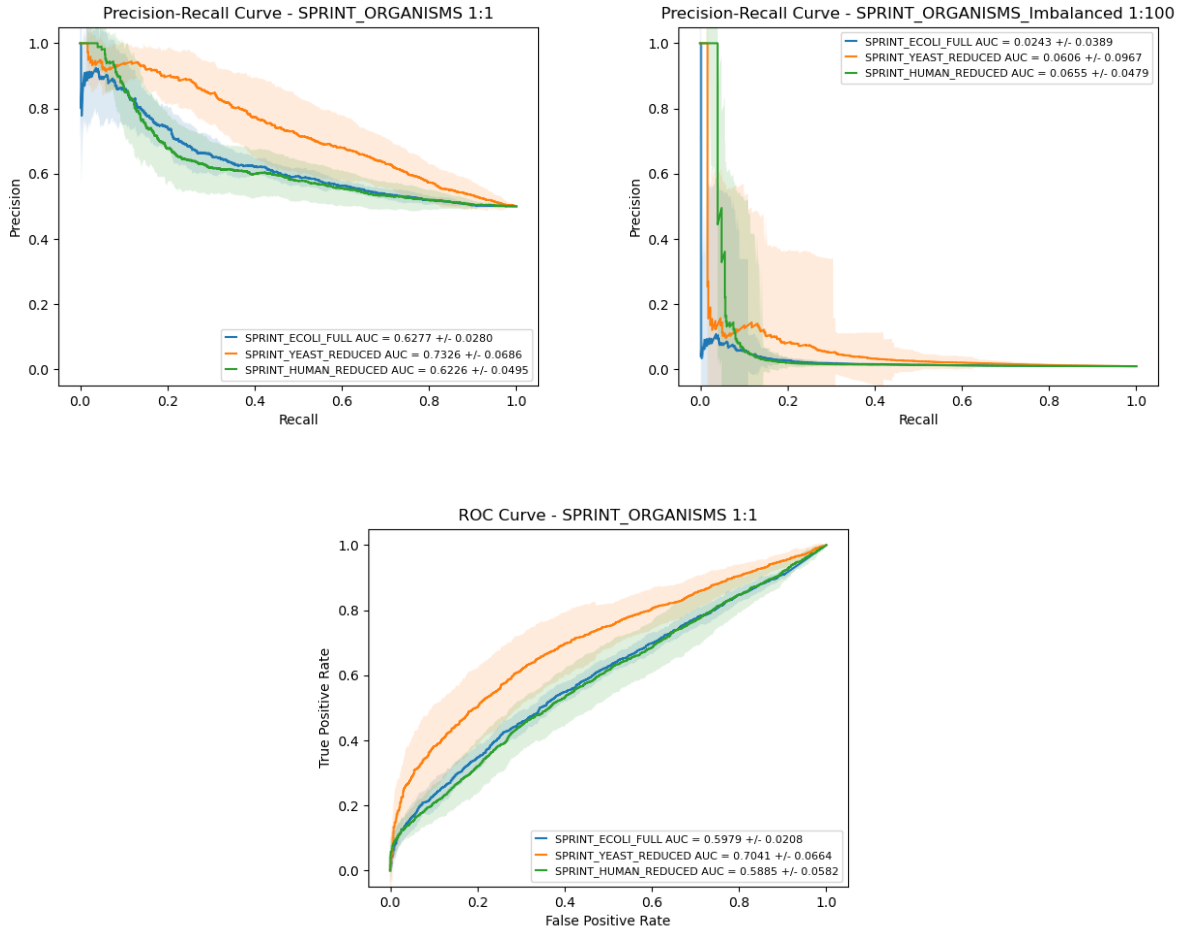

Figure 26: Performance of SPRINT on HUMAN_REDUCED, YEAST_REDUCED, and ECOLI_FULL datasets.

These results indicate that SPRINT is best at predicting yeast PPIs irrespective of dataset size. Although a performance in human PPI prediction decreased significantly with the reduced dataset size compared to that for yeast, it should be noted that there are more unique protein sequences in the HUMAN_REDUCED dataset (3,142 sequences) compared to the YEAST_REDUCED dataset (2117 sequences) and may have

contributed to the decrease in AUC. Likewise, there are twice the number of unique sequences in the YEAST_REDUCED dataset than those in ECOLI_FULL (1,221 sequences). Overall, it can then be intuitively thought that the weak performance of SPRINT on *E. coli* PPI prediction is even more likely to not only be attributed to dataset size but also to some organism-level difference in the way that PPI are mediated/enabled by protein sequences.

3.4 DPPI Performance

This section uses the datasets and performance evaluation methodologies developed in the previous section to systematically evaluate DPPI, the second of four methods representing the state-of-the-art in sequence-based PPI prediction.

3.4.1 Repeatability of Claims

One of the evaluations presented by DPPI was performed using 5-fold cross-validation on a benchmark yeast dataset constructed by Guo *et al*. [83] that consists of 11,188 (5,594 positive and 5,594 negative) PPIs. Although this benchmark dataset was unable to be accessed directly, two other methods (DEEPFE and PIPR) have both provided their copies (DEEPFE_YEAST and PIPR_YEAST) of this dataset as discussed in Section 3.2. Here, a 5-CV test was performed using each version of this benchmark yeast dataset, in addition to the independent YEAST dataset constructed in this thesis, to repeatably verify the claims presented by the DPPI authors and to further characterize its performance.

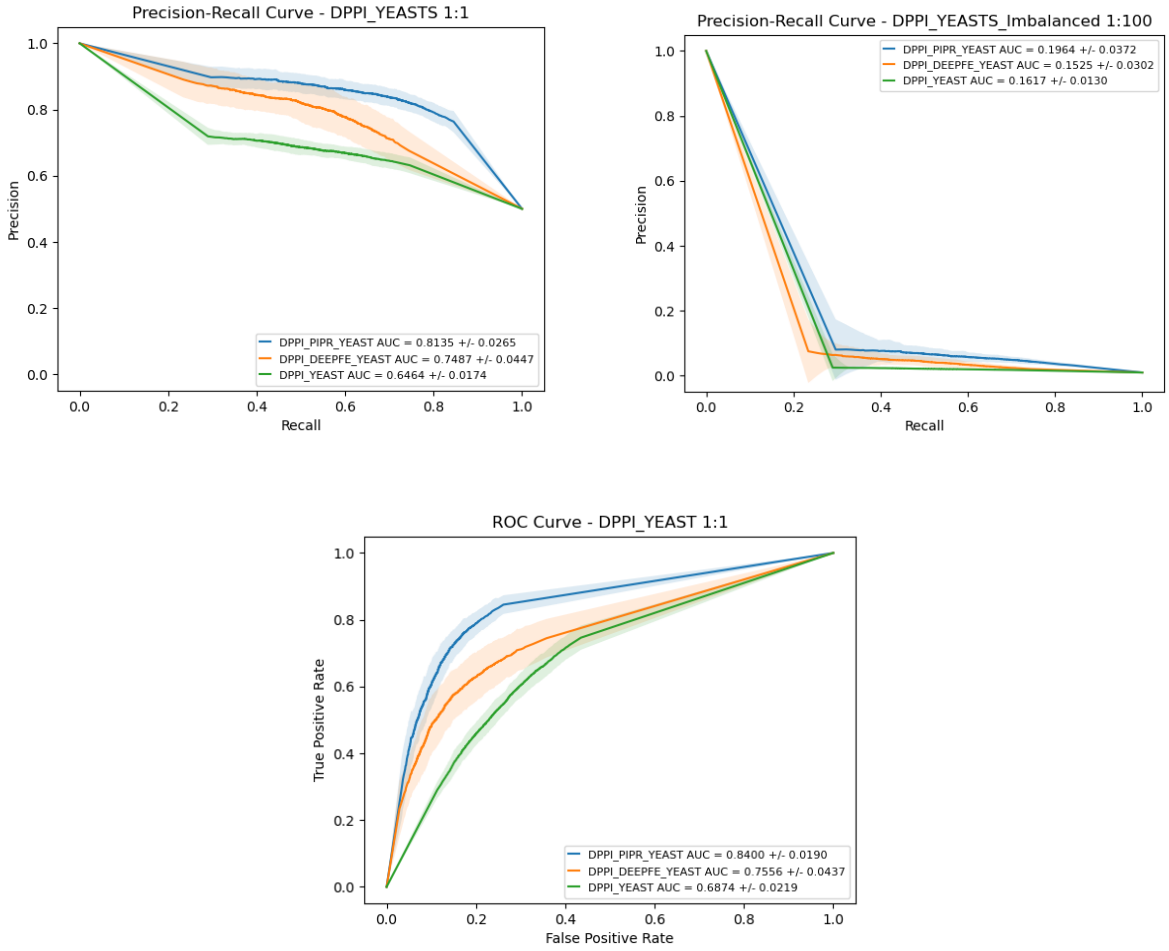

Figure 27: Repeatability of DPPI using 5-CV on the yeast datasets.

As claimed in [72], DPPI resulted in a 0.9668 precision, 0.9224 recall, and 0.9455 accuracy at a threshold of 0.5 for binary classification. Results in Figure 27 above provide a more complete picture of performance; however, using a 0.5 threshold for the repeated evaluation here resulted in values presented in Table 7 below. Nonetheless, the auPR for balanced evaluations of PIPR_YEAST, DEEPFE_YEAST, and YEAST were 0.8135 ± 0.0265, 0.7487 ± 0.0447, and 0.6464 ± 0.0174, respectively. The auPR for imbalanced evaluations were 0.1964 ± 0.0372, 0.1525 ± 0.0302, and 0.1617 ± 0.0130. The auROC for each was 0.8400 ± 0.0190, 0.7556 ± 0.0437, and 0.6874 ± 0.0219, respectively.

Table 7: DPPI repeatability experiment performance metrics using 0.5 classification threshold.

Dataset	Precision	Recall	Accuracy
PIPR_YEAST	0.85091 ± 0.02071	0.63955 ± 0.03948	0.76390 ± 0.02332
DEEPFE_YEAST	0.80716 ± 0.04665	0.53918 ± 0.04289	0.70477 ± 0.03133
YEAST	0.67887 ± 0.01503	0.53868 ± 0.03667	0.64213 ± 0.01710
Claimed Performance	0.9668	0.9224	0.9455

The table above shows some discrepancy from reported metrics which could be due to possible differences in the yeast dataset used in DPPI. Using a 0.5 decision threshold was not invalidated as these metrics were also computed using decision thresholds between 0 and 1 in increments of 0.1 with no significant change in performance values. More interesting, there is a difference in results among benchmark datasets. Without the original yeast benchmark dataset, it is difficult to draw conclusions about the repeatability in this case; however, DPPI loses its ability to make correct PPI predictions for the independent YEAST dataset despite it containing about triple the number of PPIs for training as the two yeast benchmark datasets.

3.4.2 Investigation of Cross-Species Predictions

In [72], to show generalizability of DPPI to other organisms, the benchmark yeast dataset was used to train the model to make predictions on the positive 6,954 *E. coli* PPIs (same data as positives within DEEPFE_ECOLI). An accuracy (recall) of 0.9666 was reported for predicting these *E. coli* interactions using the yeast-trained model. To investigate DPPI's ability to predict *E. coli* PPIs, DPPI was separately trained with PIPR_YEAST and DEEPFE_YEAST datasets to make predictions on DEEPFE_ECOLI. The performance of

this evaluation is shown in Figure 28. The auPR using PIPR_YEAST- and DEEPFE_YEAST-trained models were 0.5115 and 0.5228 for balanced and 0.1136 and 0.0700 for imbalanced evaluations, respectively. The auROC were 0.5208 and 0.5173.

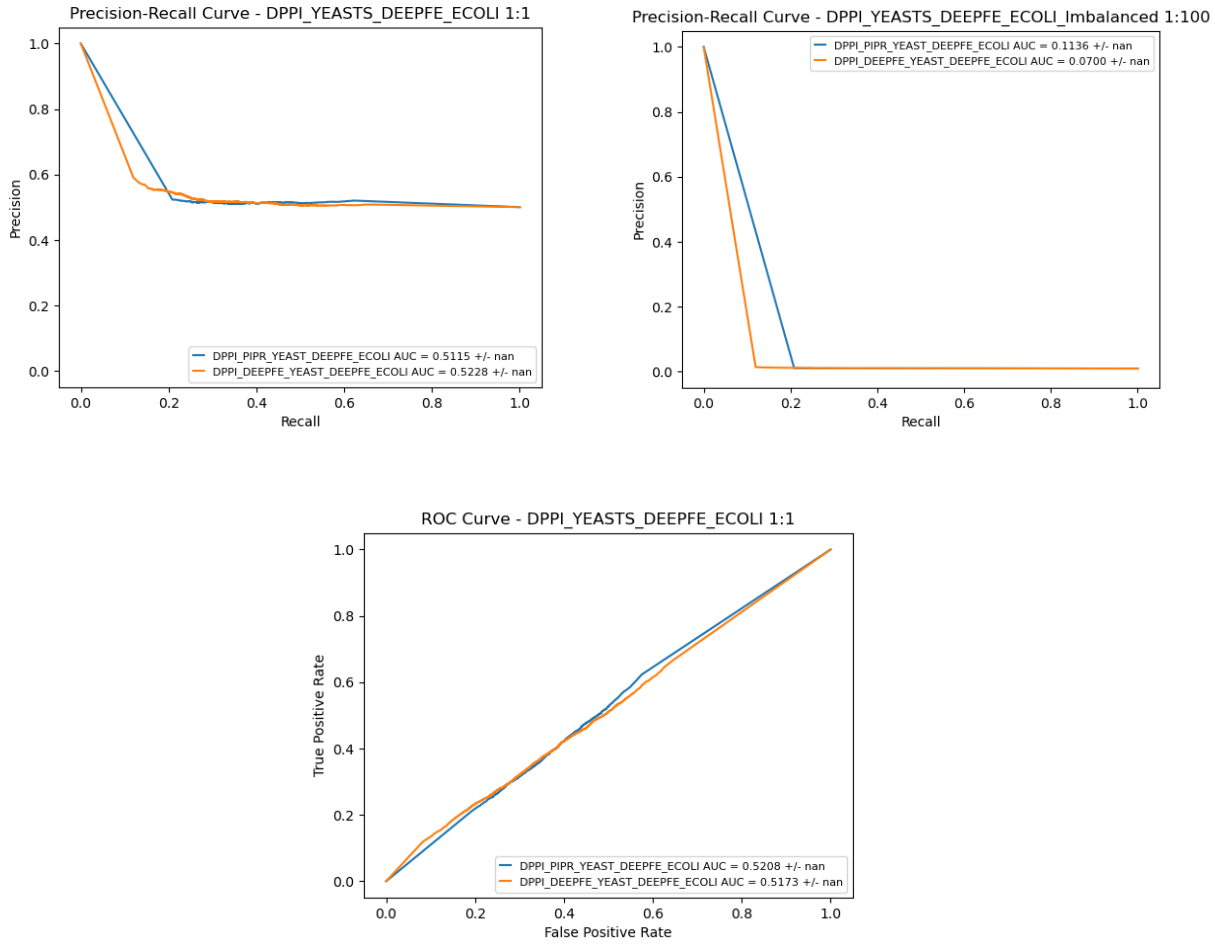

Figure 28: Evaluation of DPPI on DEEPFE_ECOLI using PIPR_YEAST and DEEPFE_YEAST trained models.

Considering only the positive PPIs in DEEPFE_ECOLI for equal comparison to the reported accuracy, and assuming a decision threshold of 0.5, DPPI correctly predicted only 2,663 out of 6,952 positives using PIPR_YEAST for training and only 2,596 out of 6,952 positives using DEEPFE_YEAST for training. This would equate to roughly 37-38% accuracy (recall).

This evaluation was repeated using ECOLI_FULL instead of DEEPFE_ECOLI as well as using a YEAST-trained model and is displayed in Figure 29 below with no significant change in performance. Overall, DPPI can be seen to be a weak predictor of *E. coli* PPIs regardless of which yeast-trained models or *E. coli*-tested datasets are used.

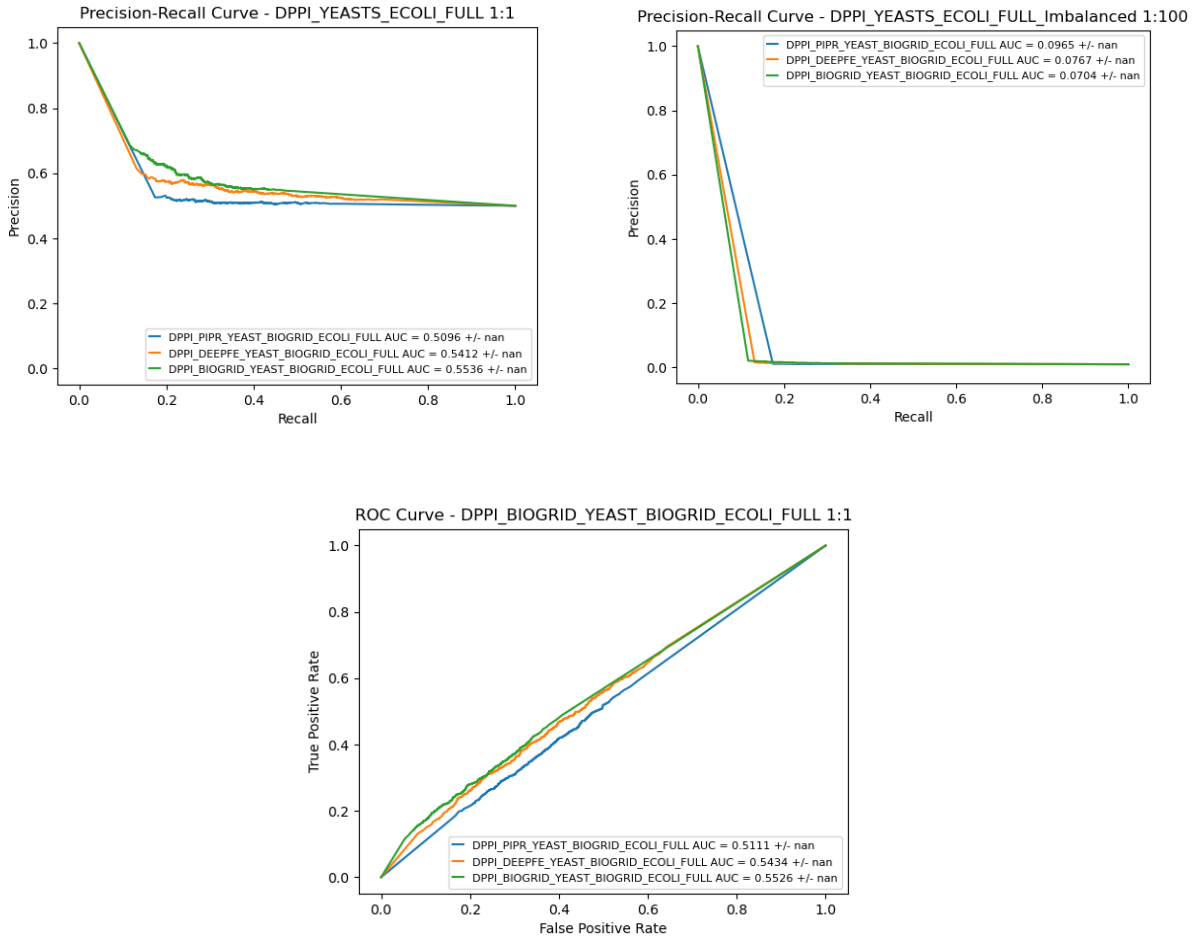

Figure 29: Evaluation of DPPI on ECOLI_FULL using each yeast-trained model.

3.4.3 Cross-Validation on *E. coli*

To assess if DPPI could perform well using an *E. coli* trained model to predict PPIs of the same species, a 10-CV experiment using ECOLI and ECOLI_FULL was performed as well as the LARGESMALL evaluation. These results can be seen in Figure 30. The auPR

for ECOLI CV, ECOLI_FULL CV, and LARGESMALL evaluations were 0.7706 ± 0.1378, 0.6189 ± 0.0586, and 0.7152, respectively for balanced evaluations. For class-imbalanced evaluation, the auPR were 0.0743 ± 0.3157, 0.0315 ± 0.0162, and 0.0379, respectively. The auROC for each was 0.8278 ± 0.1505, 0.6393 ± 0.0792, and 0.7776, respectively.

Unexpectedly, DPPI performed better on the ECOLI dataset than on ECOLI_FULL, albeit with expected greater variance in predictions. The LARGESMALL results indicate that more training data should not have decreased performance as seen with the ECOLI_FULL CV. One explanation for these results is that when training DPPI, the batch size for training was 100 samples by default when using ECOLI_FULL but was changed to 25 samples when training ECOLI due to dataset size constraints. So, the larger batch size may have reduced DPPI's ability to generalize by causing it to train faster (minimize loss) at the expense of converging on a local minimum. Otherwise, this result does not indicate that DPPI requires higher-quality training data over higher quantity, since there is divergence between ECOLI and ECOLI_FULL CV results. Overall, training DPPI with *E. coli* data performs better than a yeast-trained DPPI model for *E. coli* PPI prediction.

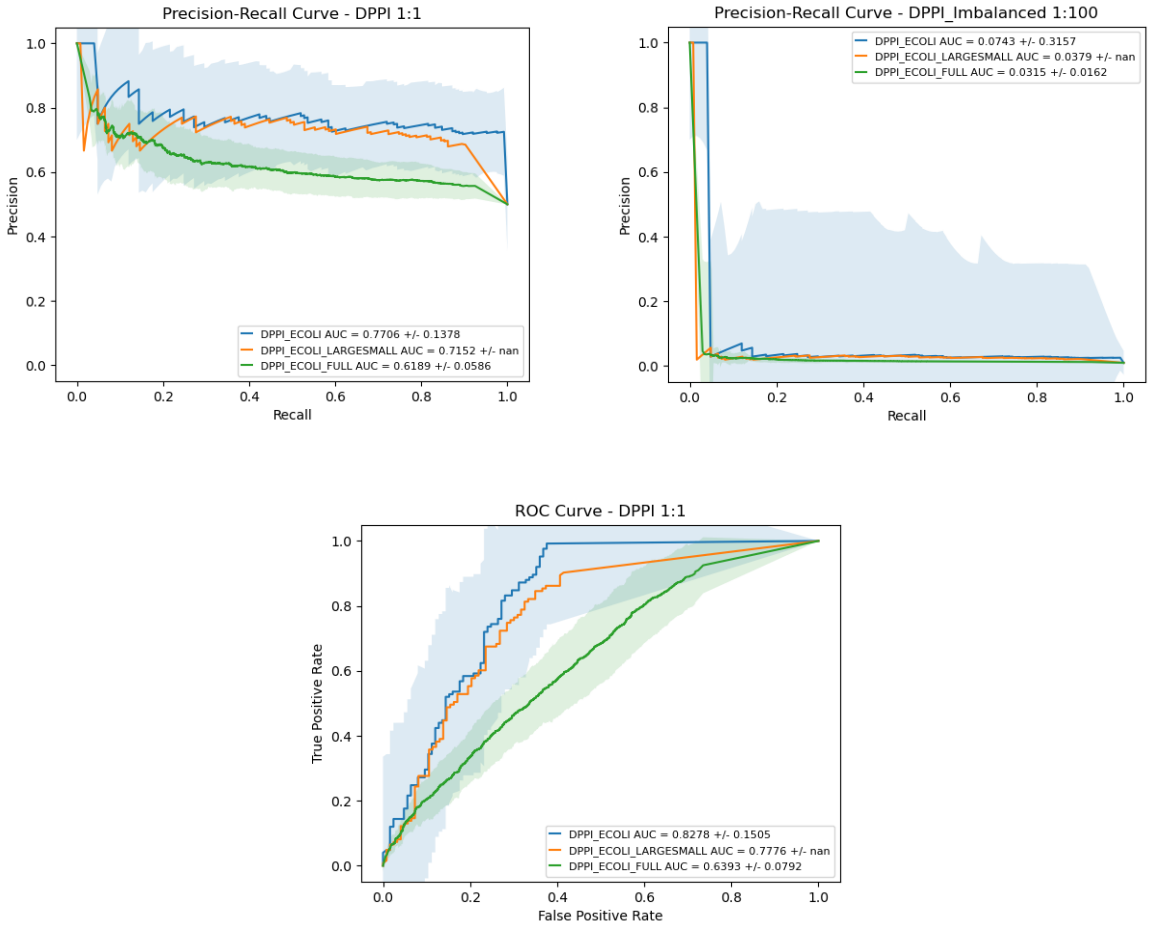

Figure 30: Evaluation of DPPI on *E. coli* datasets using 10-fold cross-validation and a LARGESMALL evaluation scheme.

3.4.4 Park & Marcotte Evaluation on *E. coli*

Similar results for the P&M evaluation below are consistent with the cross-validations above. The auPR for C1, C2, and C3 test sets were 0.6278 ± 0.0223, 0.8397 ± 0.0997, and 0.7334 ± 0.1905, respectively, under balanced evaluation. For class-imbalanced evaluation, the auPR was reduced to 0.0307 ± 0.0103, 0.0735 ± 0.2920, and 0.2033 ± 0.3896. The auROC for C1, C2, and C3 test sets was 0.6362 ± 0.0239, 0.8792 ± 0.0790, and 0.7301 ± 0.2518, respectively.

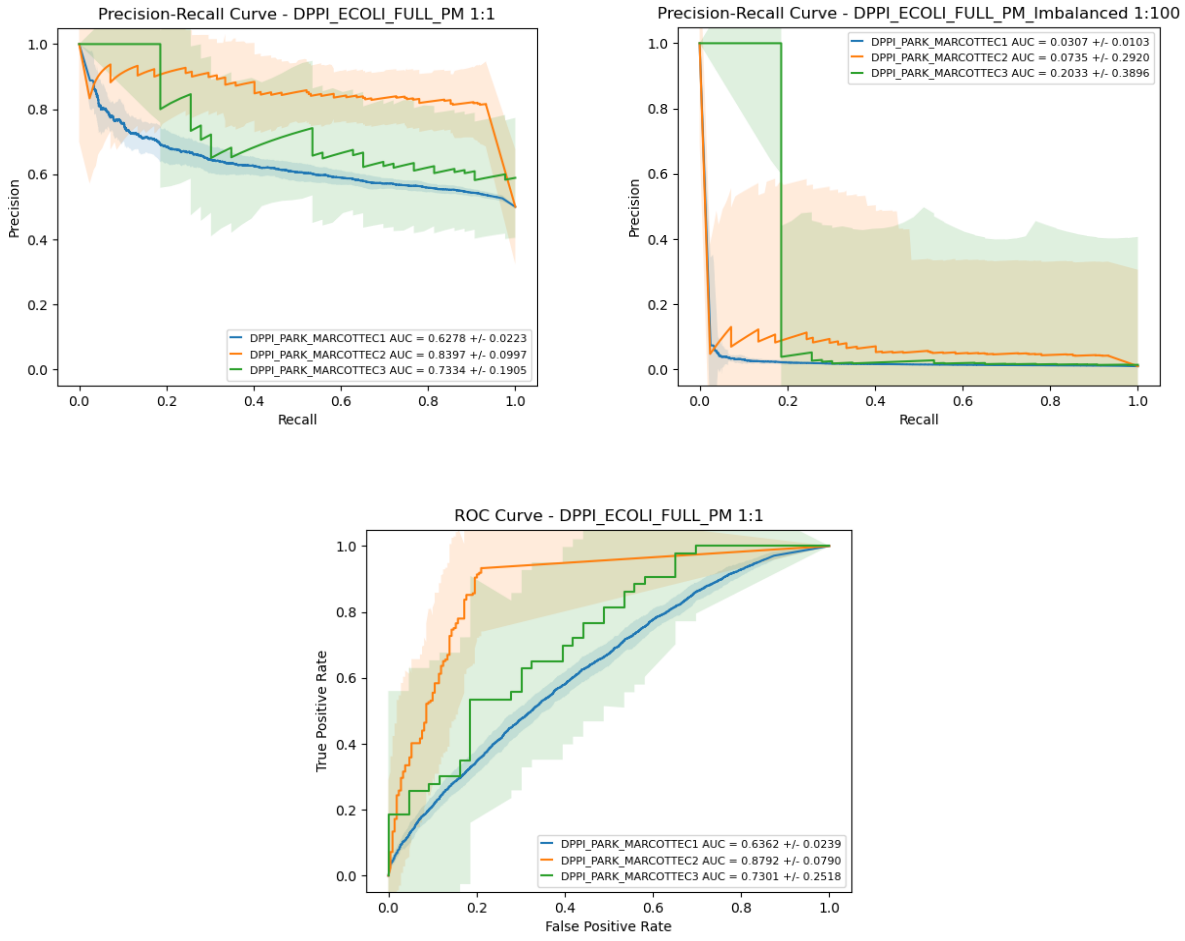

Figure 31: Park and Marcotte evaluation of *E. coli* datasets using DPPI.

3.4.5 Comparison Between Species

As with SPRINT, a 10-CV was performed on the HUMAN, YEAST, and ECOLI_FULL datasets. Results shown in Figure 32 below display poor performance on human data and best performance on yeast data. The auPR for ECOLI_FULL, YEAST, and HUMAN were 0.6189 ± 0.0586, 0.6464 ± 0.0174, and 0.4925 ± 0.0815, respectively, on balanced evaluations. For class-imbalanced evaluations, the auPR was 0.0315 ± 0.0162, 0.1617 ± 0.0130, and 0.0289 ± 0.0389. The auROC for ECOLI_FULL, YEAST, and HUMAN was 0.6393 ± 0.0792, 0.6874 ± 0.0219, and 0.4487 ± 0.1171, respectively.

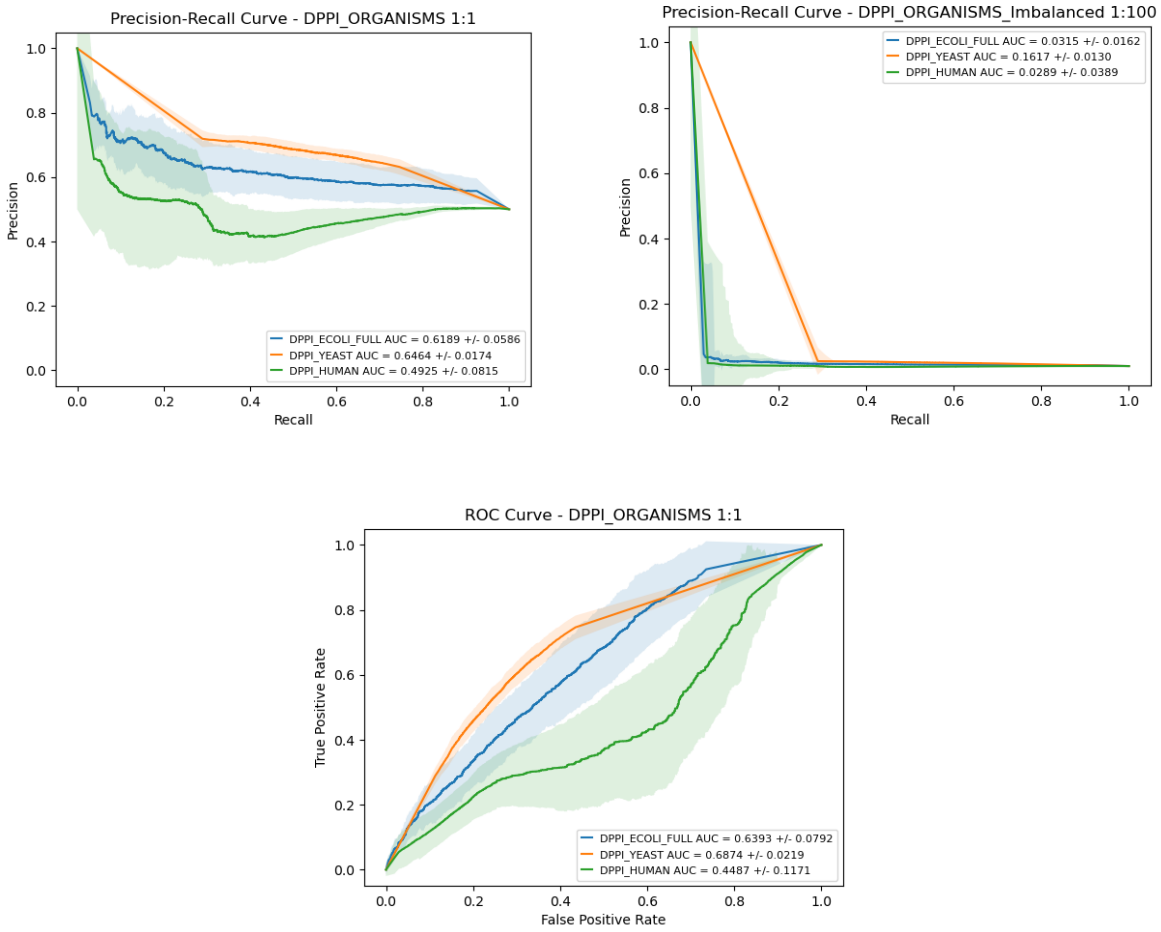

Figure 32: Performance of DPPI on HUMAN, YEAST, and ECOLI_FULL datasets.

Again, reduced yeast and human datasets were investigated showing consistent results in Figure 34 as above indicating DPPI is insensitive to relative dataset size effects on species performance. The auPR for YEAST_REDUCED and HUMAN_REDUCED changed to 0.6123 ± 0.0387 and 0.5207 ± 0.0356, respectively on balanced evaluations. For class-imbalanced evaluations, the auPR became 0.0846 ± 0.0223 and 0.0427 ± 0.0137. The auROC for YEAST_REDUCED and HUMAN_REDUCED changed to 0.6435 ± 0.0404 and 0.5109 ± 0.0546, respectively.

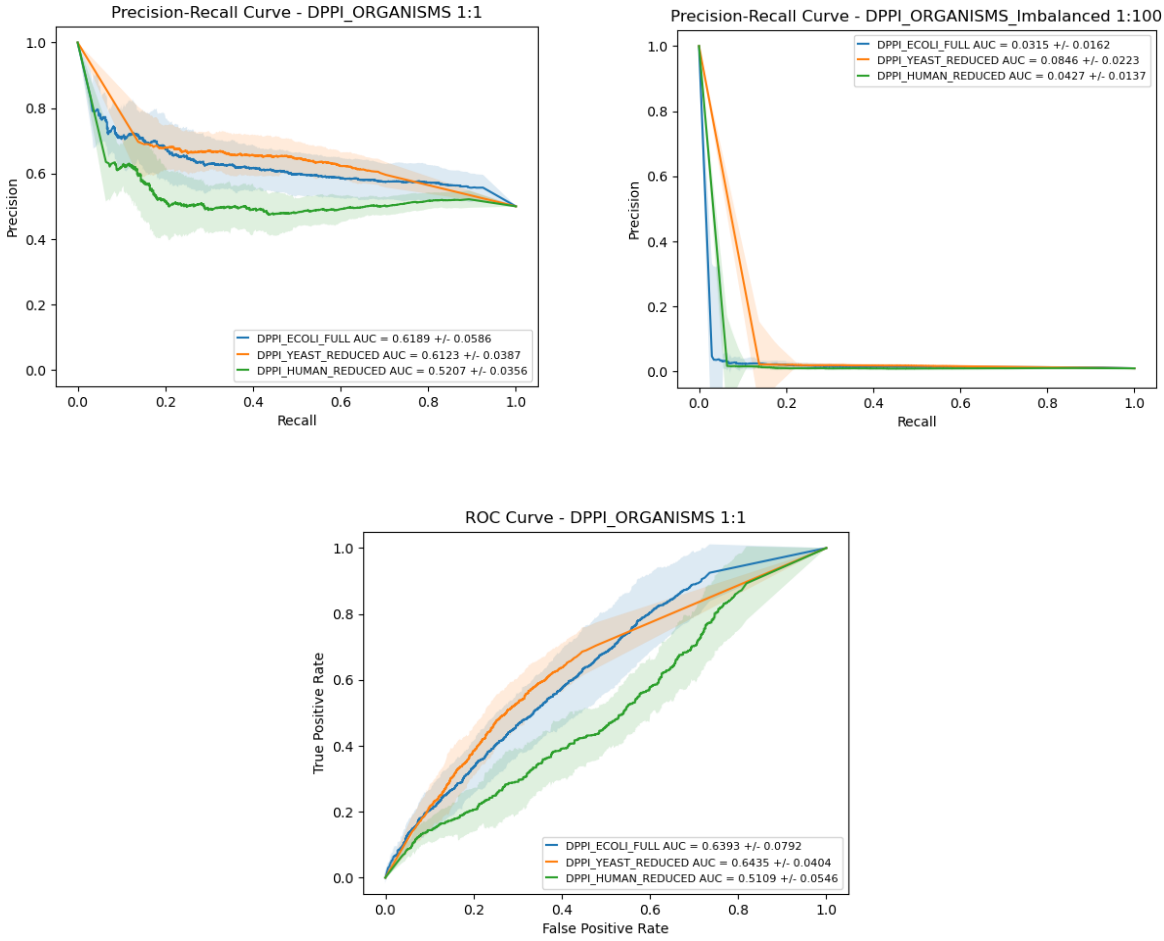

Figure 33: Performance of DPPI on HUMAN_REDUCED, YEAST_REDUCED, and ECOLI_FULL datasets.

3.5 DeepFE-PPI Performance

This section uses the datasets and performance evaluation methodologies developed in the previous section to systematically evaluate DEEPFE, the third of four methods representing the state-of-the-art in sequence-based PPI prediction.

3.5.1 Repeatability of Claims

In this direction, DEEPFE was independently evaluated with a local implementation of the evaluation scheme conducted in [73]. First, a 5-CV was performed using the yeast

benchmark dataset represented by DEEPFE_YEAST. The reported performance metrics are shown in Table 8 with a comparison to similar results obtained here, indicating that the evaluation is repeatable.

Table 8: DEEPFE repeatability experiment results using a 0.5 decision threshold.

Results	Accuracy	Recall	Precision	MCC	auPR	auROC
Reported	0.9478 ± 0.0061	0.9299 ± 0.0066	0.9645 ± 0.0087	0.8962 ± 0.0123	0.9863	0.9829
Repeated	0.94530 ± 0.00398	0.92588 ± 0.01486	0.96361 ± 0.00991	0.89143 ± 0.00781	0.98713	0.98347

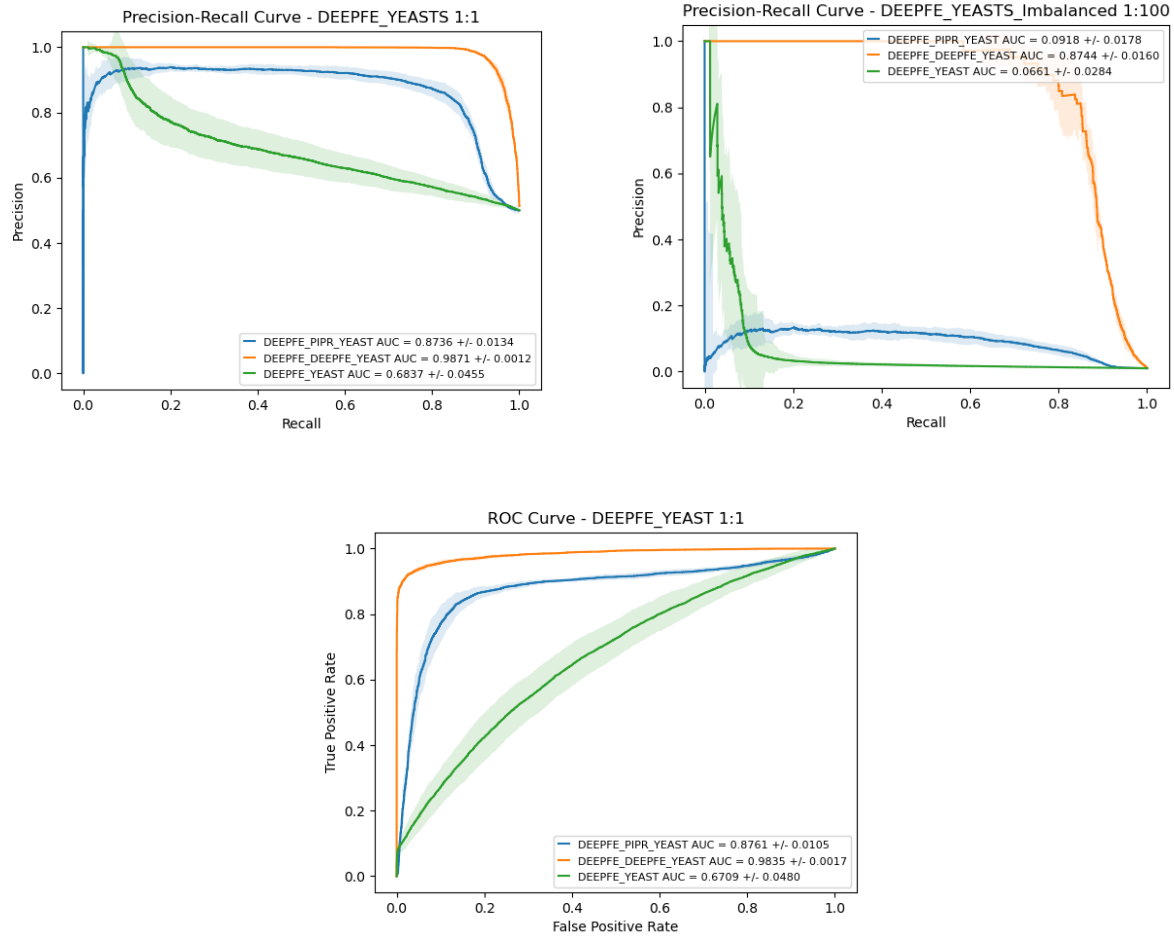

Figure 34: Performance curves of DEEPFE using CV of yeast datasets.

To explore the consistency of using DEEPFE for predicting yeast PPIs and to gain insight of its overall performance, a cross-validation was also performed using the PIPR_YEAST, and YEAST datasets. These results are shown in Figure 34 above. The auPR for balanced evaluations of PIPR_YEAST, DEEPFE_YEAST, and YEAST were 0.8736 ± 0.0134, 0.9871 ± 0.0012, and 0.6837 ± 0.0455, respectively. The auPR for imbalanced evaluations were 0.0918 ± 0.0178, 0.8744 ± 0.0160, and 0.0661 ± 0.0284. The auROC for each was 0.8761 ± 0.0105, 0.9835 ± 0.0017, and 0.6709 ± 0.0480, respectively.

The performance above shows that DEEPFE is best at making predictions using the version of the benchmark yeast dataset that the authors provided and was consistent with the reported claims as expected. A drop in performance occurred when using the PIPR_YEAST version of the benchmark yeast dataset. Performance declined further when using the independently compiled YEAST dataset despite having greater training samples. This suggests that DEEPFE greatly overfits the PIPR_YEAST and YEAST datasets and that the model's hyperparameters are finely tuned for the DEEPFE_YEAST dataset. Thus, this DEEPFE model is very sensitive to the dataset used for training.

3.5.2 Investigation of Cross-Species Predictions

The significant drop in performance shown by the YEAST cross-validation experiment prompted further investigation of DEEPFE for predicting yeast PPIs in general. In this effort, DEEPFE was trained using DEEPFE_YEAST to make predictions of the YEAST dataset built here, based on BioGRID. The performance curves in Figure 35 below indicate that DEEPFE fails to generalize to prediction of yeast PPI outside of its training

set, providing further evidence that the model is overfit and appears to be dataset-specific even within the same species.

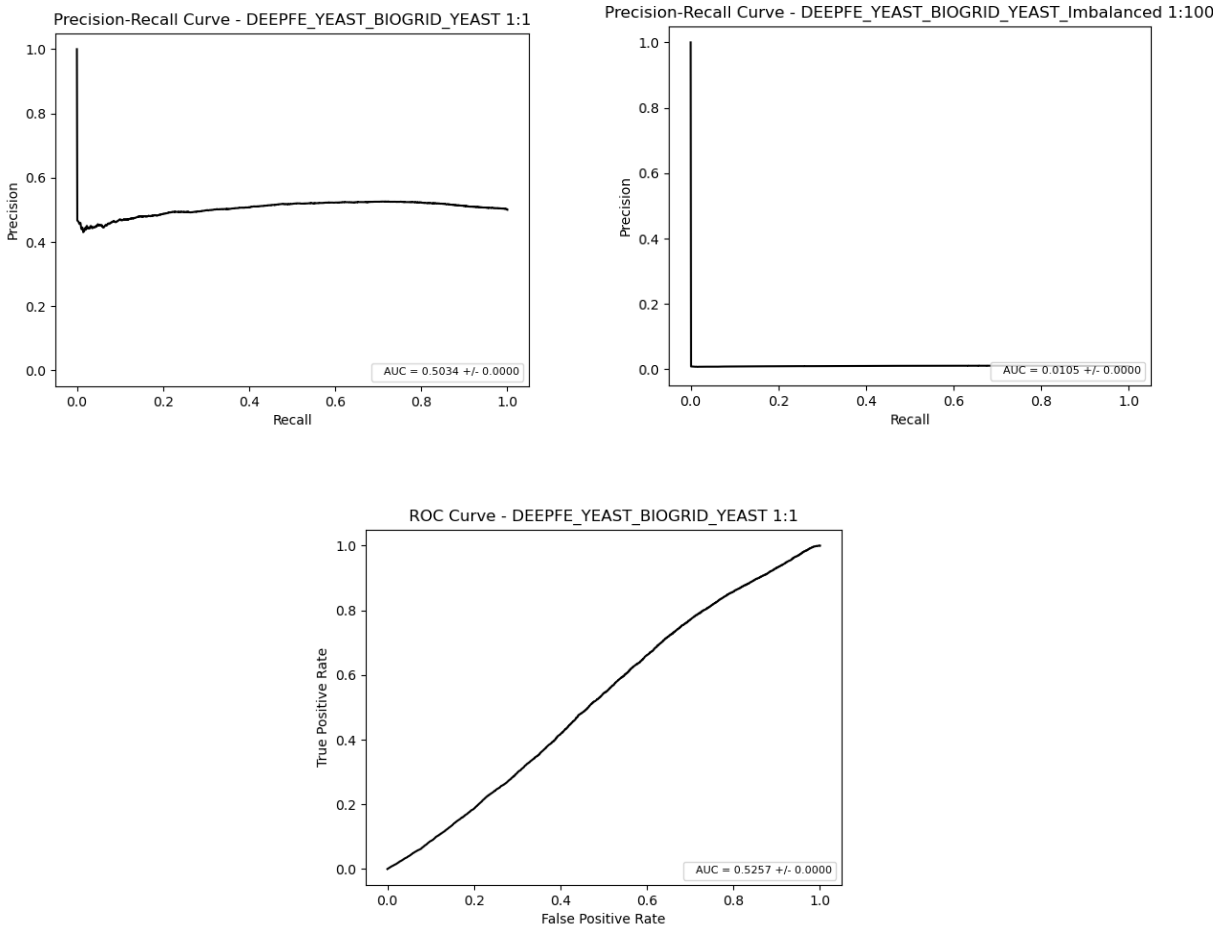

Figure 35: Evaluation of DEEPFE trained with DEEPFE to predict the YEAST dataset.

Finally, DEEPFE prediction performance requires further investigation for its potential use in bacterial PPI prediction. The original DEEPFE model trained with DEEPFE_YEAST was tested on DEEPFE_ECOLI, ECOLI and ECOLI_FULL to determine its ability to predict bacterial PPIs. The reported claim of 100% accuracy (recall) in predicting DEEPFE_ECOLI positive-only PPIs tested here. Of the 6,952 PPIs in the DEEPFE_ECOLI dataset, only 5,809 true positives were predicted, representing an

83.6% accuracy (recall). Results shown in Figure 36 suggest again that DEEPFE seems specific to the yeast benchmark dataset and may not generalize well to predicting PPIs from other datasets or species.

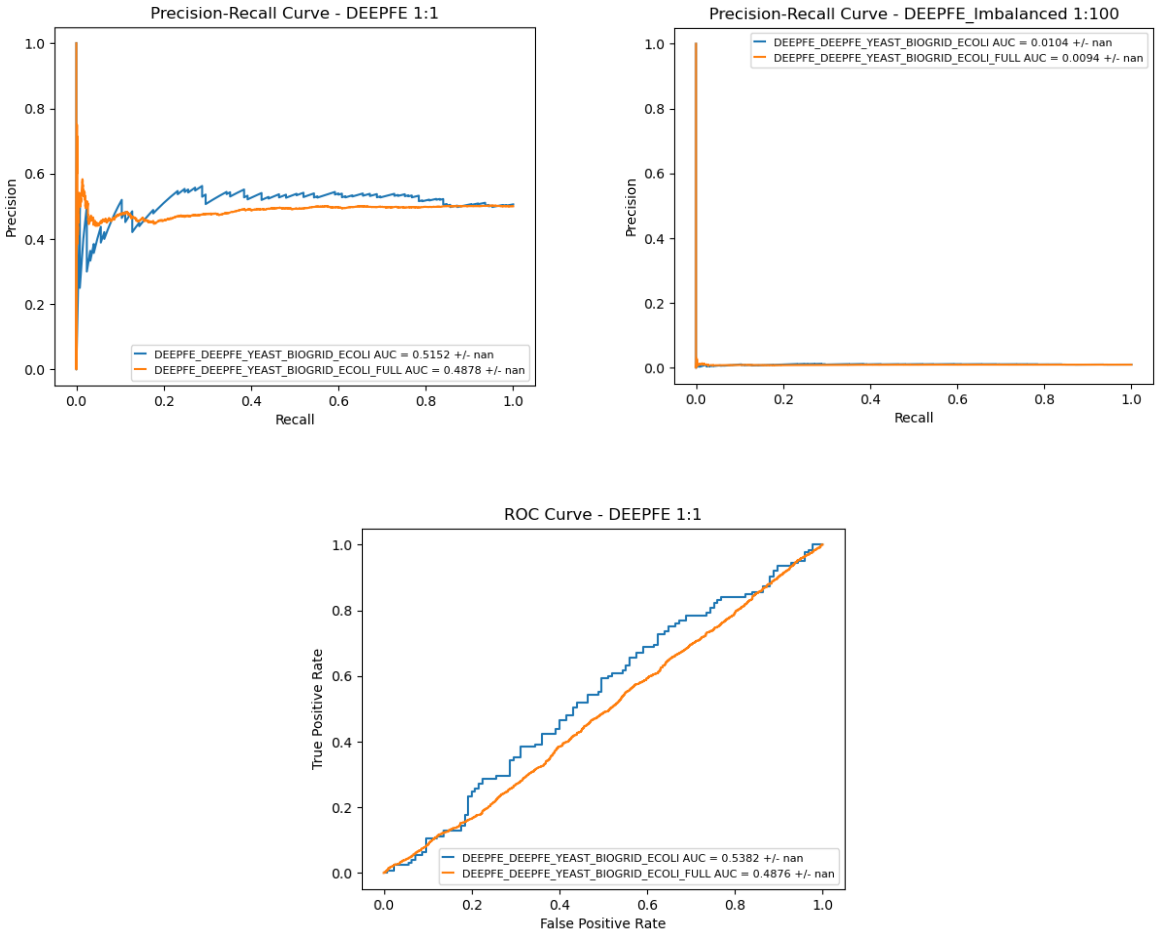

Figure 36: Evaluation of DEEPFE trained with DEEPFE_YEAST to predict ECOLI and ECOLI_FULL datasets.

3.5.3 Cross-Validation on *E. coli*

A 10-CV was performed for *E. coli* datasets to investigate if bacterial PPI predictions can improve with a DEEPFE model trained with the same species. Results in Figure 37 indicates weak potential for DEEPFE to predict *E. coli* PPIs even when trained with *E. coli* data. Further evidence of overfitting can also be seen in Figure 74 under Appendix B.

An increase in performance can be seen with increased samples used for training as shown by the ECOLI_FULL and LARGESMALL performance curves coinciding. The auPR for ECOLI CV, ECOLI_FULL CV, and LARGESMALL evaluations were 0.4237 ± 0.0816, 0.5621 ± 0.0502, and 0.5576, respectively for balanced evaluations. For class-imbalanced evaluation, the auPR were 0.0074 ± 0.0365, 0.0135 ± 0.0377, and 0.0213, respectively. The auROC for each was 0.3896 ± 0.1676, 0.5379 ± 0.0492, and 0.4950, respectively.

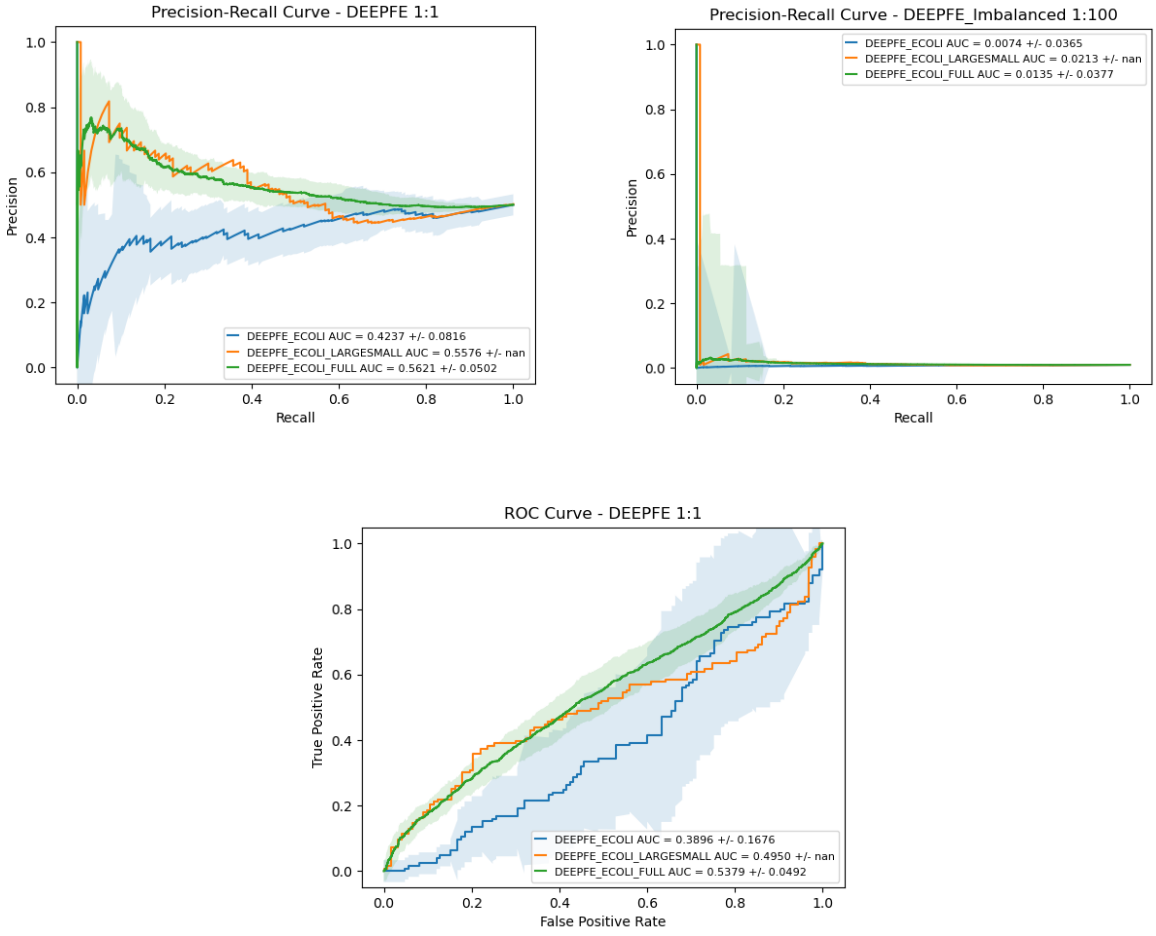

Figure 37: Evaluation of DEEPFE on *E. coli* datasets using 10-fold cross-validation and a LARGESMALL evaluation scheme.

3.5.4 Park & Marcotte Evaluation on *E. coli*

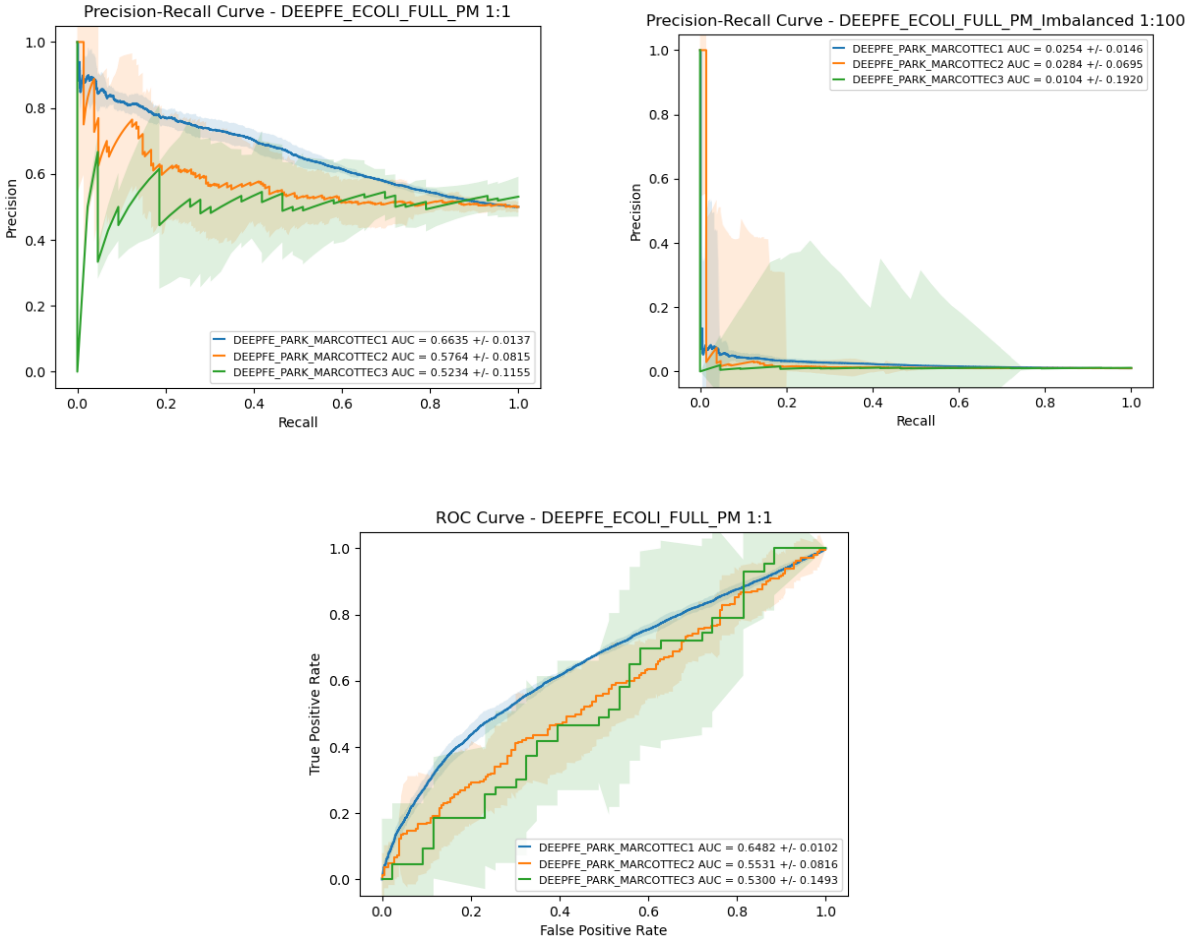

Figure 38: Park and Marcotte evaluation of *E. coli* datasets using DEEPFE.

As with the previous two methods, Figure 38 displays declining performance when predicting PPIs for proteins with less prevalence in the training PPI data. The auPR for C1, C2, and C3 test sets were 0.6635 ± 0.0137, 0.5764 ± 0.0815, and 0.5234 ± 0.1155, respectively, under balanced evaluation. For class-imbalanced evaluation, the auPR was 0.0254 ± 0.0146, 0.0284 ± 0.0695, and 0.0104 ± 0.1920. The auROC for C1, C2, and C3 test sets was 0.6482 ± 0.0102, 0.5531 ± 0.0816, and 0.5300 ± 0.1493, respectively.

3.5.5 Comparison Between Species

An investigation of differences between species' datasets when using DEEPFE can be seen in Figure 39. Again, a 10-CV was performed for each dataset. The auPR for ECOLI_FULL, YEAST, and HUMAN were 0.5621 ± 0.0502, 0.6837 ± 0.0445, and 0.5180 ± 0.1189, respectively, on balanced evaluations. For class-imbalanced evaluations, the auPR was 0.0135 ± 0.0377, 0.0661 ± 0.0284, and 0.0108 ± 0.0109. The auROC for ECOLI_FULL, YEAST, and HUMAN was 0.5379 ± 0.0492, 0.6709 ± 0.0480, and 0.5342 ± 0.1393, respectively.

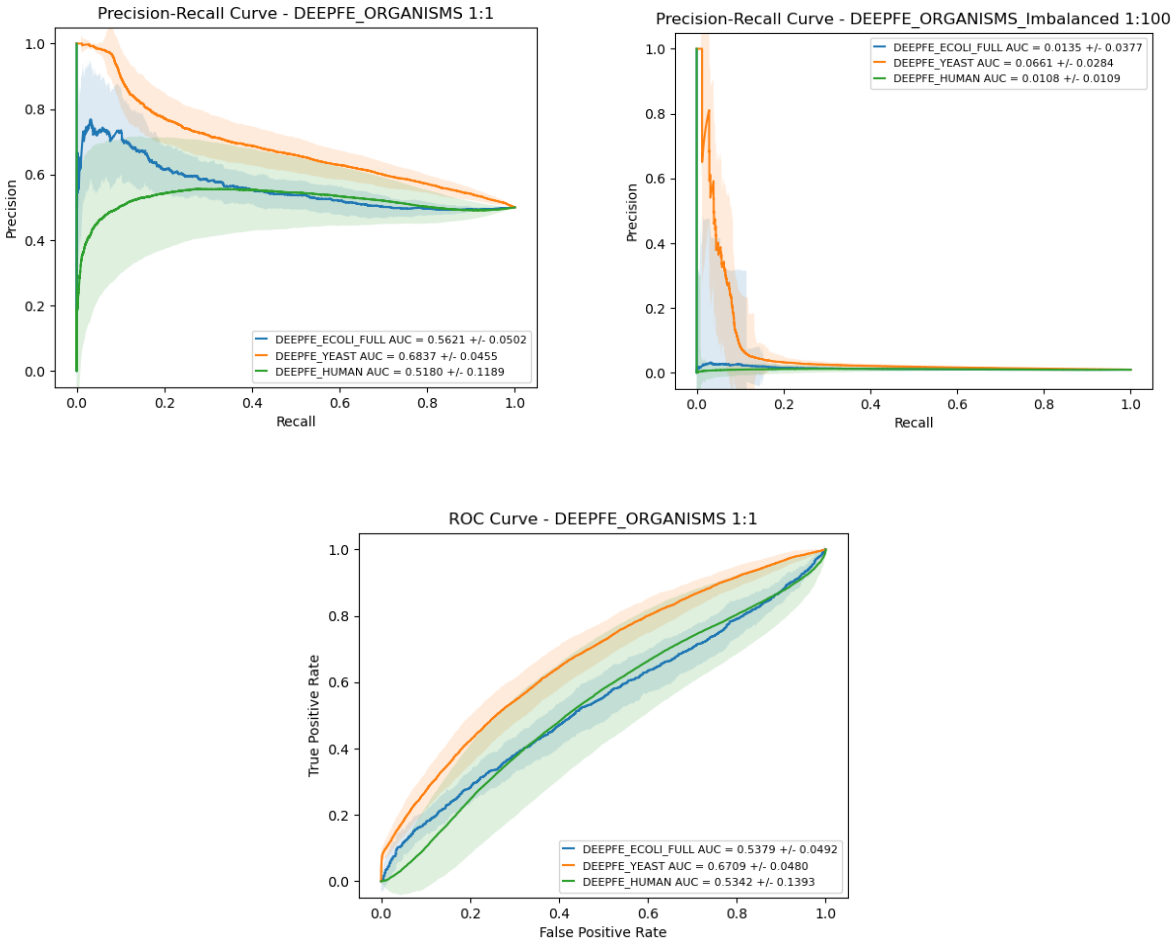

Figure 39: Performance of DEEPFE using 10-CV on HUMAN, YEAST, and ECOLI_FULL datasets.

The same evaluation as above was conducted here where the yeast and human datasets are artificially constrained to be the same size as the ECOLI_FULL dataset. This resulted in relatively consistent performance curves shown in Figure 40 as above. This indicates DEEPFE is best at predicting yeast PPIs and is weaker for *E. coli* and human PPI prediction. Interestingly, DEEPFE appears to perform better at predicting *E. coli* PPIs than human PPIs. The auPR for YEAST_REDUCED and HUMAN_REDUCED changed to 0.6565 ± 0.0798 and 0.3947 ± 0.0573, respectively, on balanced evaluations. For class-imbalanced evaluations, the auPR became 0.0270 ± 0.0978 and 0.0067 ± 0.0017. The auROC for YEAST_REDUCED and HUMAN_REDUCED changed to 0.6357 ± 0.0742 and 0.3374 ± 0.1151, respectively.

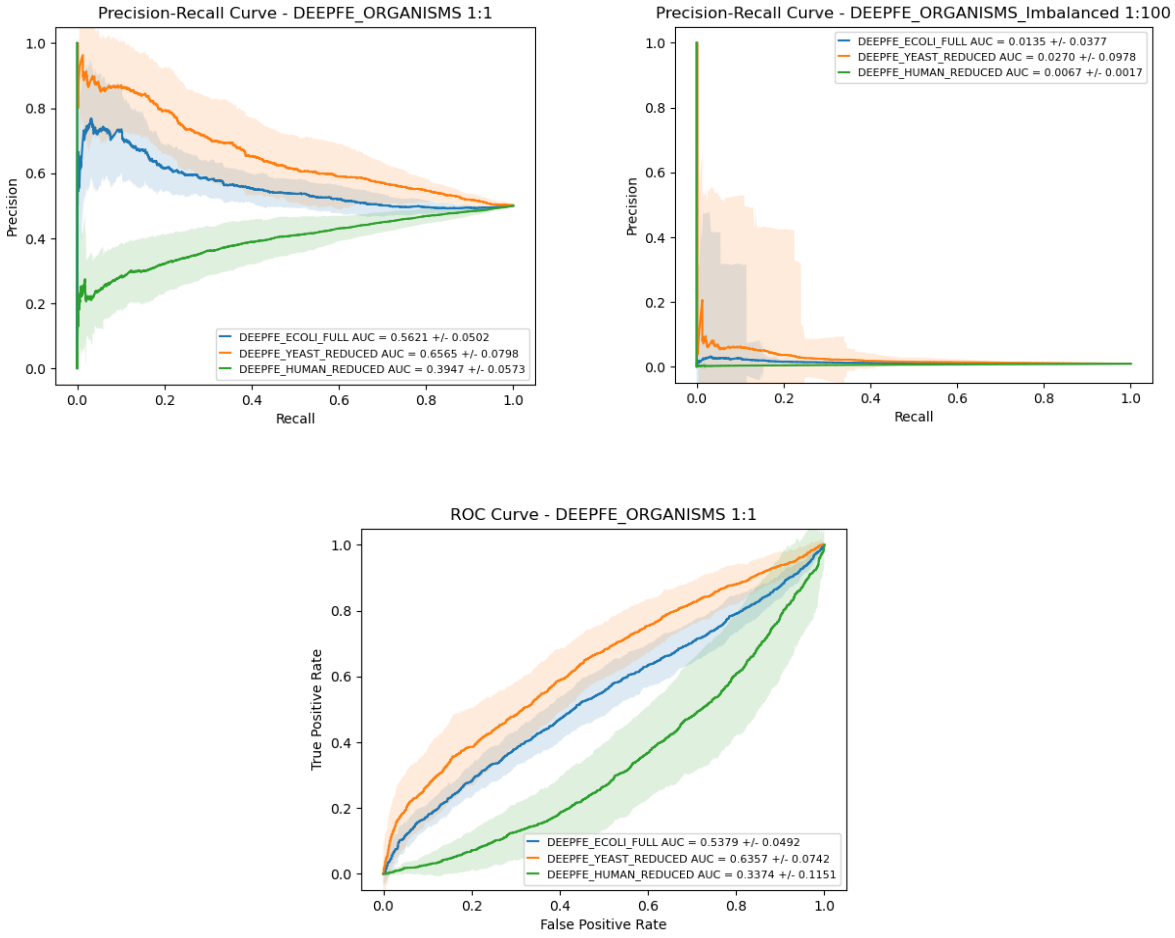

Figure 40: Performance of DEEPFE on HUMAN_REDUCED, YEAST_REDUCED, and ECOLI_FULL datasets.

3.6 PIPR Performance

This section uses the datasets and performance evaluation methodologies developed in the previous section to systematically evaluate PIPR, the last of four methods representing the state-of-the-art in sequence-based PPI prediction.

3.6.1 Repeatability of Claims

For binary classification evaluation, PIPR was originally evaluated using a 5-CV on the benchmark yeast dataset from Guo *et al*. [83]; several performance metrics were

reported, including accuracy, precision, sensitivity, specificity, F1, and MCC on the positive interactions. Using the version of the yeast benchmark provided by the authors (PIPR_YEAST), repeating this experiment resulted in similar prediction performance seen in Table 9. This indicates that the local implementation of PIPR is consistent with the originally published method.

Table 9: PIPR repeatability experiment results using a 0.5 decision threshold.

Results	Accuracy	Recall	Precision	MCC
Reported	0.9709 ± 0.0024	0.9717 ± 0.0044	0.9700 ± 0.0065	0.9417 ± 0.0048
Repeated	0.95672 ± 0.01548	0.95060 ± 0.03314	0.96287 ± 0.00812	0.91424 ± 0.02958

Performance curves in Figure 41 from a 10-CV evaluation show differences in PIPR's prediction capabilities when using different yeast datasets. As with DEEPFE, PIPR performed best using its own version of the benchmark dataset. Again, significant loss in performance is apparent when using the YEAST dataset compiled here. The auPR for balanced evaluations of PIPR_YEAST, DEEPFE_YEAST, and YEAST were 0.9830 ± 0.0080, 0.9744 ± 0.0037, and 0.5783 ± 0.0224, respectively. The auPR for imbalanced evaluations were 0.4674 ± 0.1611, 0.6634 ± 0.0651, and 0.0420 ± 0.0104. The auROC for each was 0.9860 ± 0.0053, 0.9704 ± 0.0040, and 0.5721 ± 0.0262, respectively.

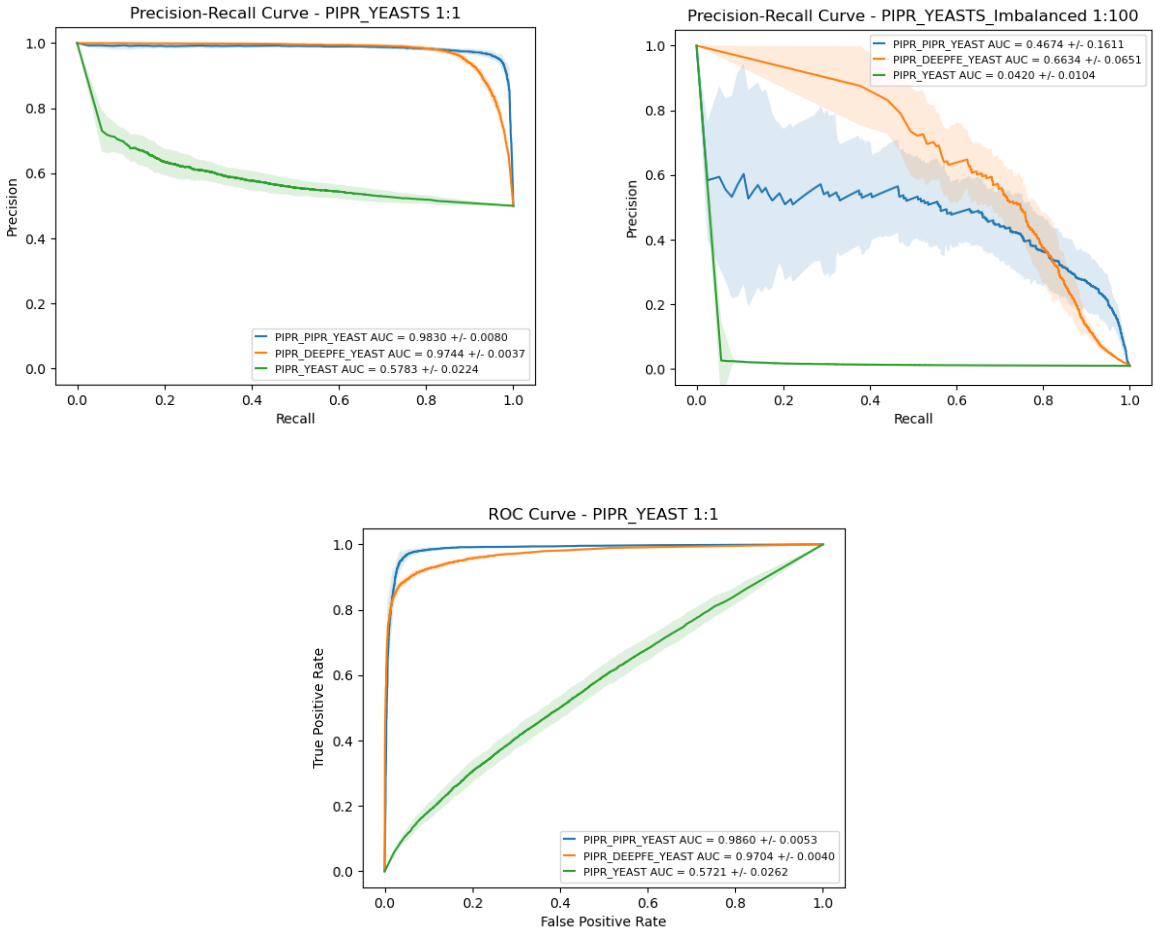

Figure 41: Performance curves of PIPR using CV of yeast datasets.

3.6.2 Investigation of Cross-Species Predictions

To independently investigate PIPR's application for bacterial PPI prediction, PIPR was trained using PIPR_YEAST to predict ECOLI and ECOLI_FULL interactions. Results in Figure 42 below are similar to DEEPFE and show weak ability to predict *E. coli* PPIs.

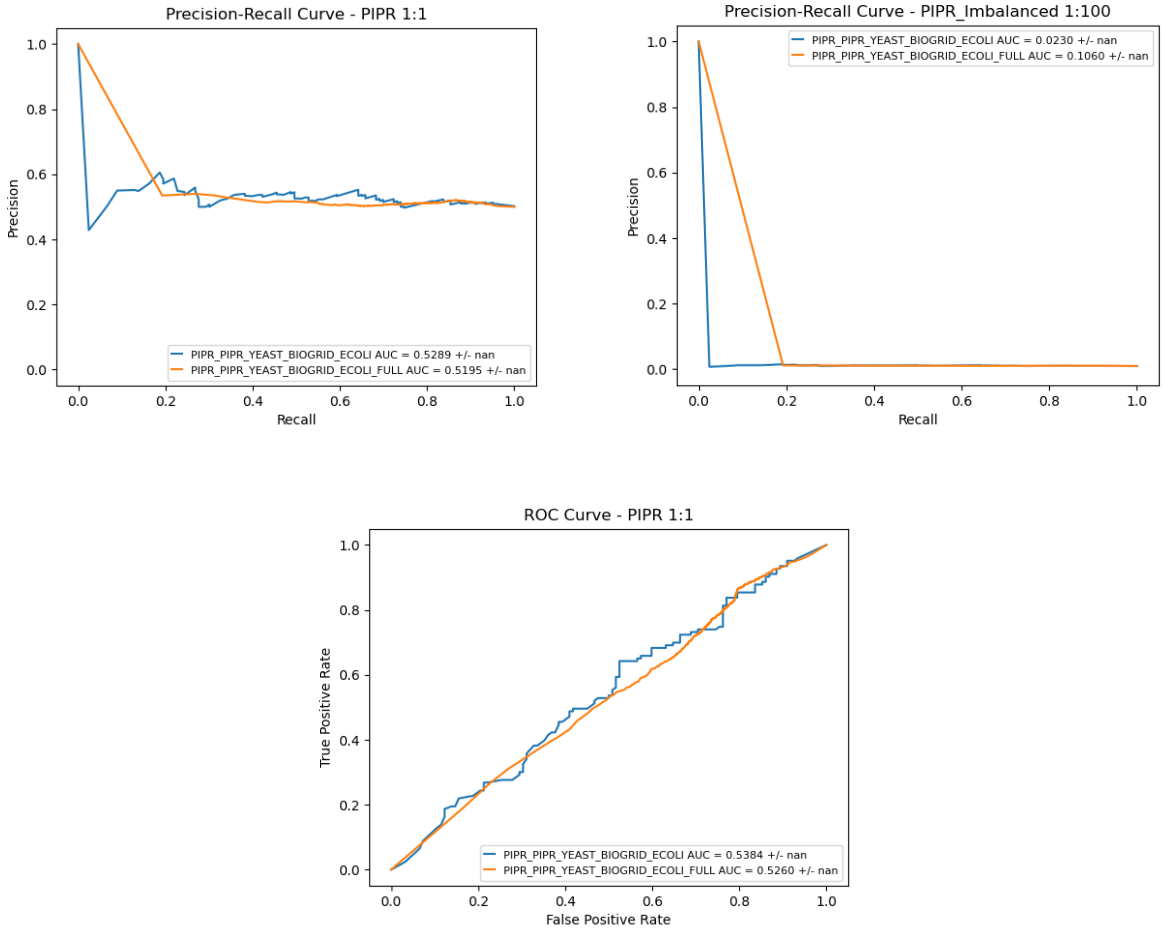

Figure 42: Evaluation of PIPR trained with PIPR_YEAST to predict ECOLI and ECOLI_FULL datasets.

3.6.3 Cross-Validation on *E. coli*

Instead of using a model trained on yeast data, PIPR was trained with *E coli* data to evaluate ability to predict PPIs of the same species. For this, a 10-CV test was performed for ECOLI and ECOLI_FULL datasets as well as the LARGESMALL evaluation scheme. Performance curves in Figure 43 show that additional training data does not significantly affect PIPR's ability to make correct predictions, unlike previous classifiers. The auPR for ECOLI CV, ECOLI_FULL CV, and LARGESMALL evaluations were 0.6157 ± 0.0496, 0.5922 ± 0.0260, and 0.5492, respectively, for balanced evaluations. For class-

imbalanced evaluation, the auPR were 0.0398 ± 0.1015, 0.0598 ± 0.0256, and 0.1001, respectively. The auROC for each was 0.6024 ± 0.0435, 0.5937 ± 0.0313, and 0.5394, respectively.

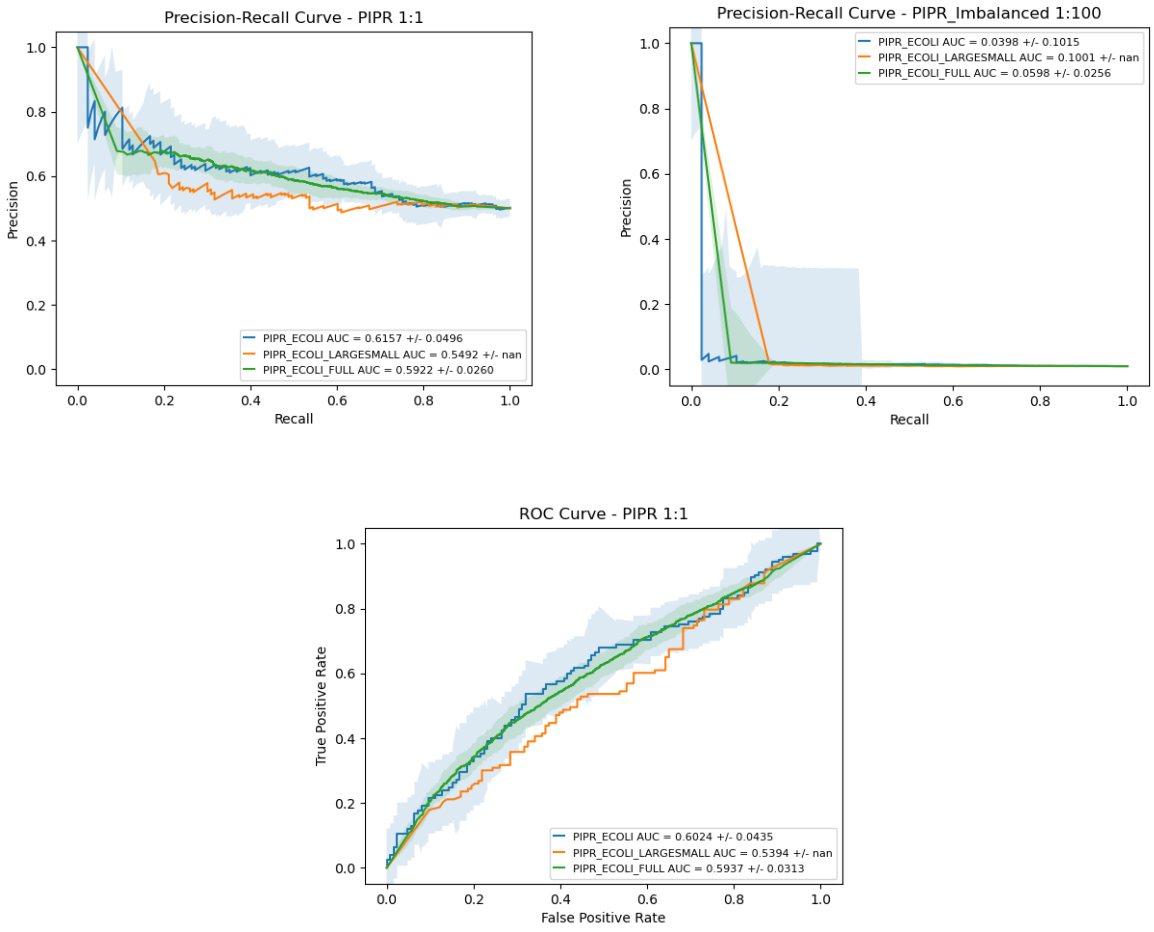

Figure 43: Evaluation of PIPR on *E. coli* datasets using 10-fold cross-validation and a LARGESMALL evaluation scheme.

3.6.4 Park & Marcotte Evaluation on *E. coli*

The resulting curves shown by Figure 44 in this P&M evaluation are contrary to expectations of declining performance through C1, C2, and C3 test sets. This could simply be the result of very few testing pairs in the C3 sets, so the likelihood of true positives may be over-represented in the C3 test versus the C1 test because there are

less samples from which to make predictions and evaluate. This could also explain similar results seen in the P&M evaluation for DPPI. The auPR for C1, C2, and C3 test sets were 0.6032 ± 0.0122, 0.5588 ± 0.0692, and 0.7301 ± 0.1103, respectively under balanced evaluation. For class-imbalanced evaluation, the auPR was 0.0647 ± 0.0215, 0.0976 ± 0.0873, and 0.1847 ± 0.2183. The auROC for C1, C2, and C3 test sets was 0.6064 ± 0.0137, 0.5938 ± 0.0743, and 0.6952 ± 0.1714, respectively.

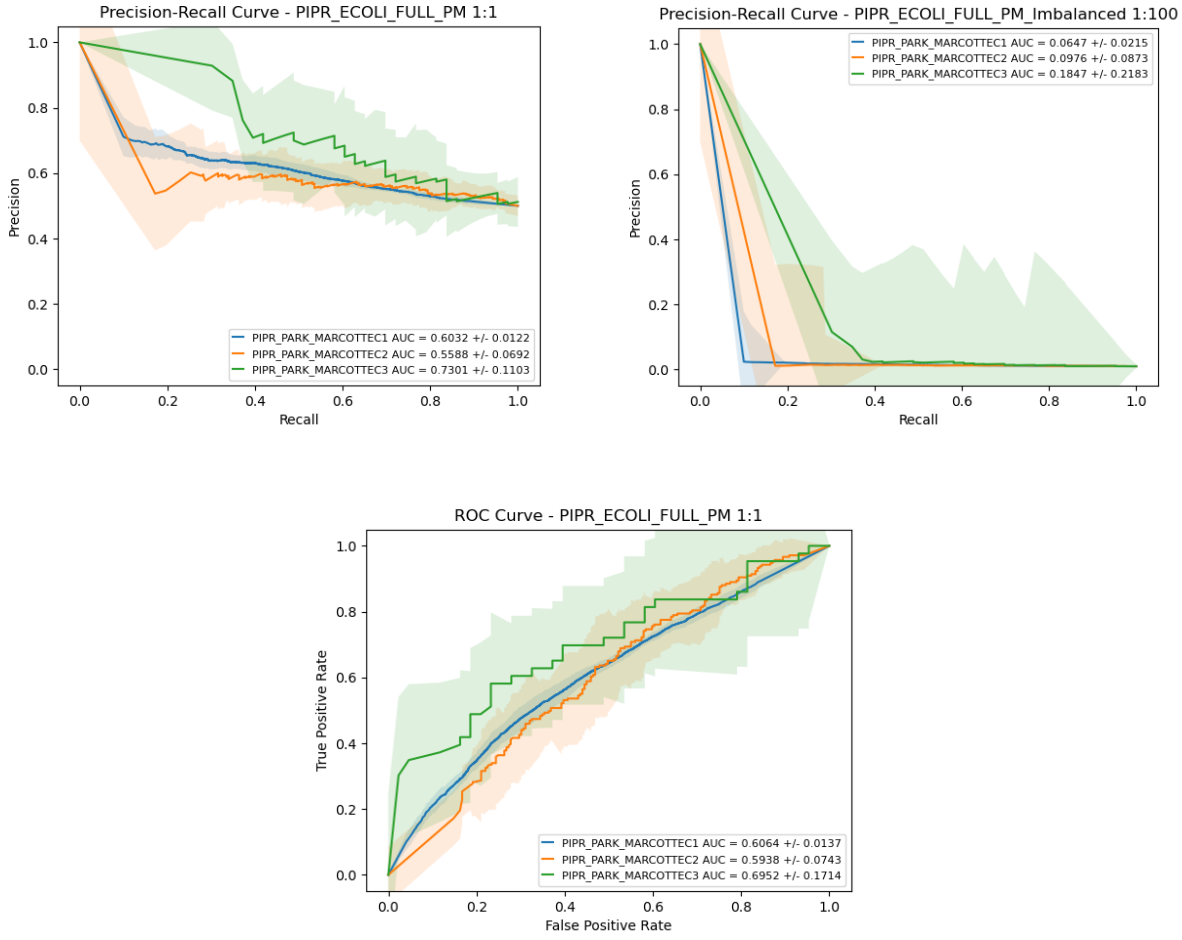

Figure 44: Park and Marcotte evaluation of *E. coli* datasets using PIPR.

3.6.5 Comparison Between Species

Finally, PIPR was evaluated on species-specific datasets. Results in Figure 45 show no significant difference in ability to predict a species' PPIs when training the model with the same species. The auPR for ECOLI_FULL, YEAST, and HUMAN were 0.5922 ± 0.0260, 0.5783 ± 0.0224, and 0.5886 ± 0.0131, respectively on balanced evaluations. For class-imbalanced evaluations, the auPR was 0.0598 ± 0.0256, 0.0420 ± 0.0104, and 0.0322 ± 0.0059. The auROC for ECOLI_FULL, YEAST, and HUMAN was 0.5937 ± 0.0313, 0.5721 ± 0.0262, and 0.5871 ± 0.0143, respectively.

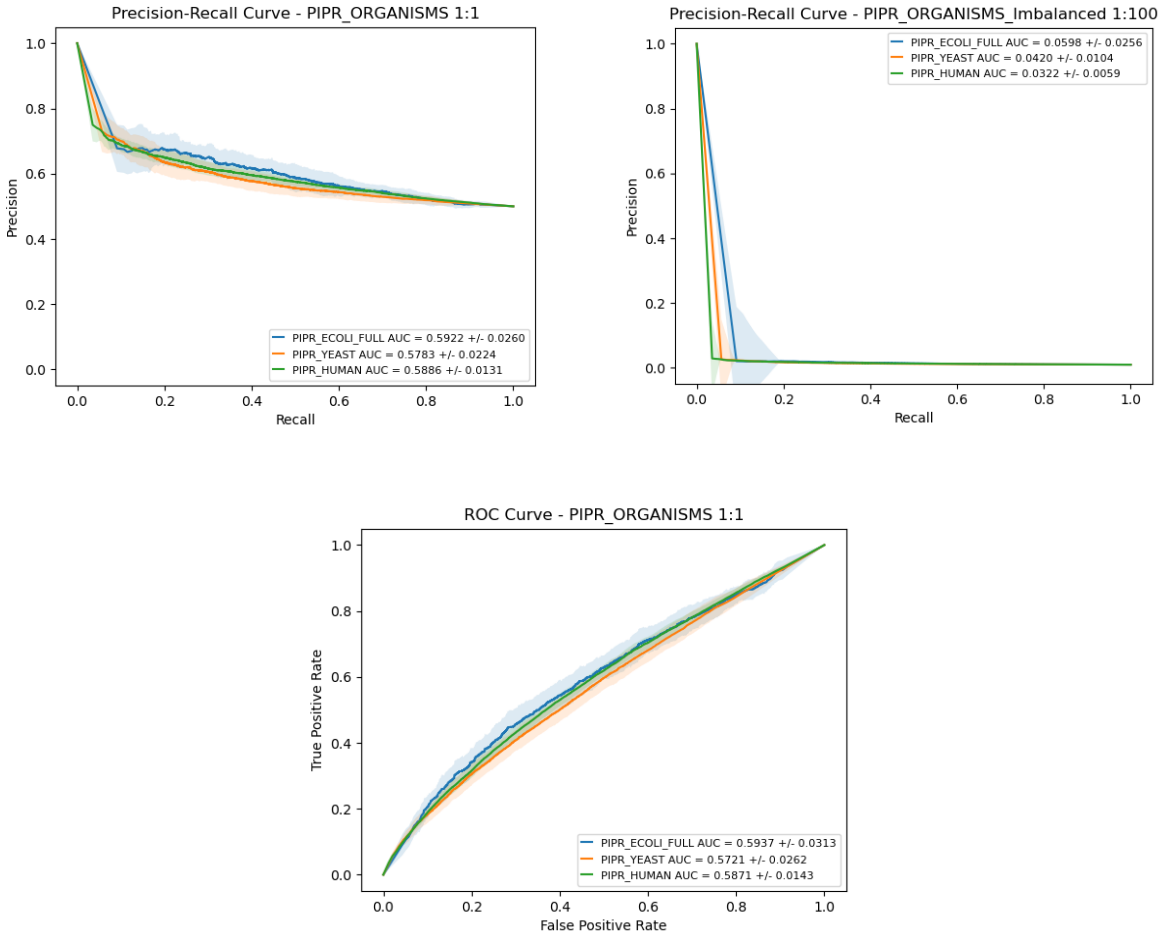

Figure 45: Performance of PIPR on HUMAN, YEAST, and ECOLI_FULL datasets.

Accounting for dataset size differences, repeating this evaluation on reduced human and yeast data shows that PIPR could actually be better at predicting *E. coli* PPIs. However, it should be reminded that a decrease in performance on the human and yeast datasets may also be due to a greater number of unique protein sequences among their PPIs than for ECOLI_FULL. The auPR for YEAST_REDUCED and HUMAN_REDUCED changed to 0.5089 ± 0.0674 and 0.5086 ± 0.0494, respectively on balanced evaluations. For class-imbalanced evaluations, the auPR became 0.0231 ± 0.0208 and 0.0137 ± 0.0055. The auROC for YEAST_REDUCED and HUMAN_REDUCED changed to 0.5150 ± 0.0710 and 0.5204 ± 0.0596, respectively.

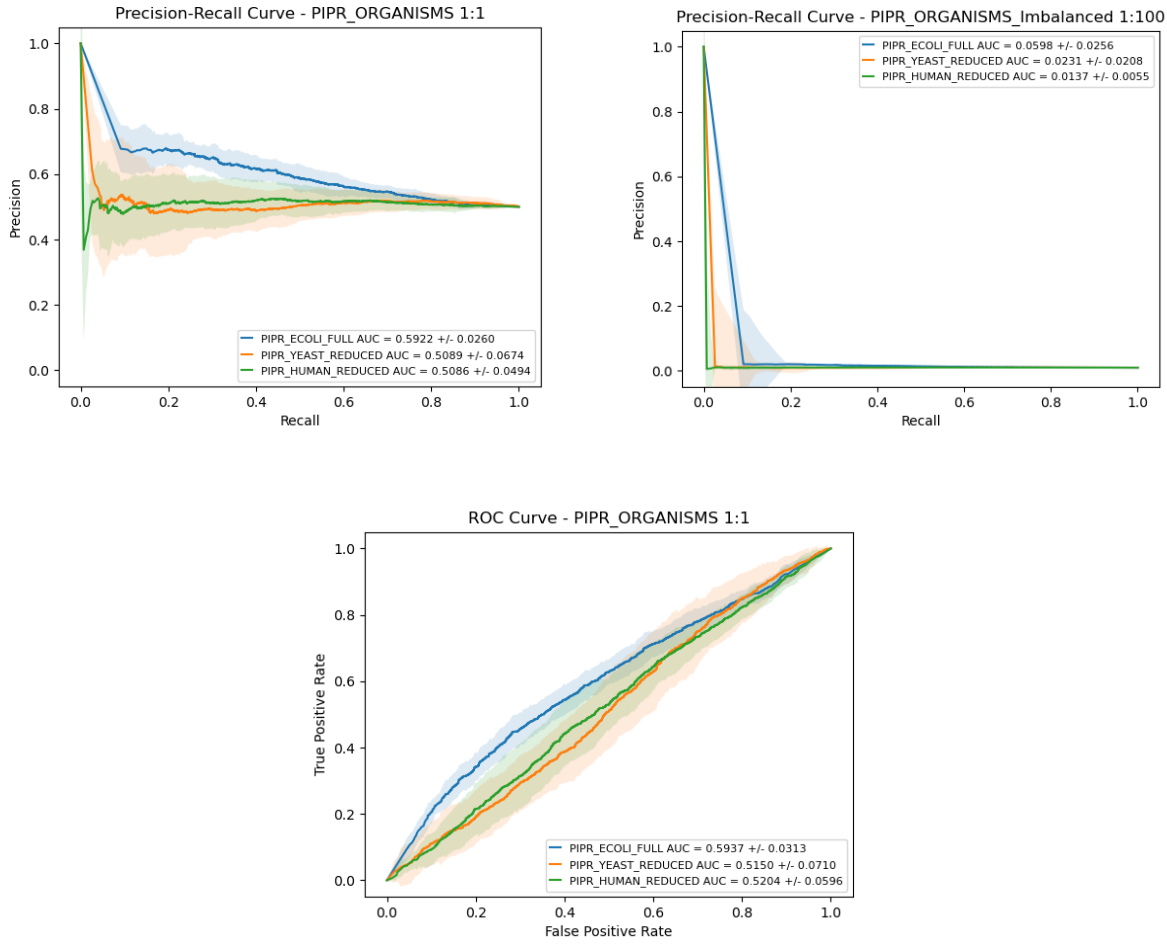

Figure 46: Performance of PIPR on HUMAN_REDUCED, YEAST_REDUCED, and ECOLI_FULL datasets.

3.7 Discussion and Conclusion

All the above evaluations provide a general perspective as to the state-of-the-art in binary PPI prediction using sequence-based predictors. These results suggest that congruency between datasets and evaluations is required to effectively progress PPI prediction research. That is, species-specific predictors are expected to outperform cross-species methods where the training and testing species differ.

First, the re-use of outdated benchmark datasets that are not consistent between method evaluations may create overly optimistic expectations when used to predict PPIs.

Therefore, there needs to be a single accessible source of established benchmark datasets to remove discrepancies. Additionally, these benchmark datasets should be updated to reflect the current knowledge of interactions as new PPI data are generated from experiments. This thesis and accompanying source code at https://github.com/GreenCUBIC/Bacterial-PPI-Prediction addresses this issue.

Secondly, a consistent and comprehensive set of performance metrics would enable classifiers to be more fairly compared. Often, a publication may only present the metrics that cast the method in the best light, which can hide overall results and prevent comparisons to be made. Additionally, classifiers should provide PR and ROC performance curves and not limit classification thresholds to 0.5 to quantify greater perspective of results. Additionally, PPI prediction is a class imbalance problem, so prevalence-corrected metrics should be calculated for more practical interpretation of results. This thesis proposes that these evaluation protocols become an established norm in PPI classification tasks.

Lastly, a reminder that the purpose of this investigation was to identify a method for accurate prediction of bacterial PPIs, using *E. coli* as a representative model organism. Results clearly indicate that each method requires models to be trained with *E. coli* data instead of using the original, non-bacterial organisms. A comparison of each method is shown in Figure 47 and Figure 48 below for the ECOLI and ECOLI_FULL datasets. As noted earlier, these results indicate that DPPI is the best classifier for this task when using ECOLI. However, note the large standard deviation in performance apparent when evaluating the methods using the ECOLI dataset, with only 125 PPIs and 125 NIPs. This uncertainty is significantly reduced when using the larger ECOLI_FULL dataset. Despite

the higher performance observed when using the ECOLI dataset, using ECOLI_FULL results in more stable performance and more confident conclusions can be drawn. This further indicates that more PPI data are required to build effective prediction models as was seen in the comparison between human and yeast evaluations above when the dataset sizes were artificially constrained to match the size of ECOLI_FULL. Thus, ECOLI_FULL will be used for further investigations in the following Chapter.

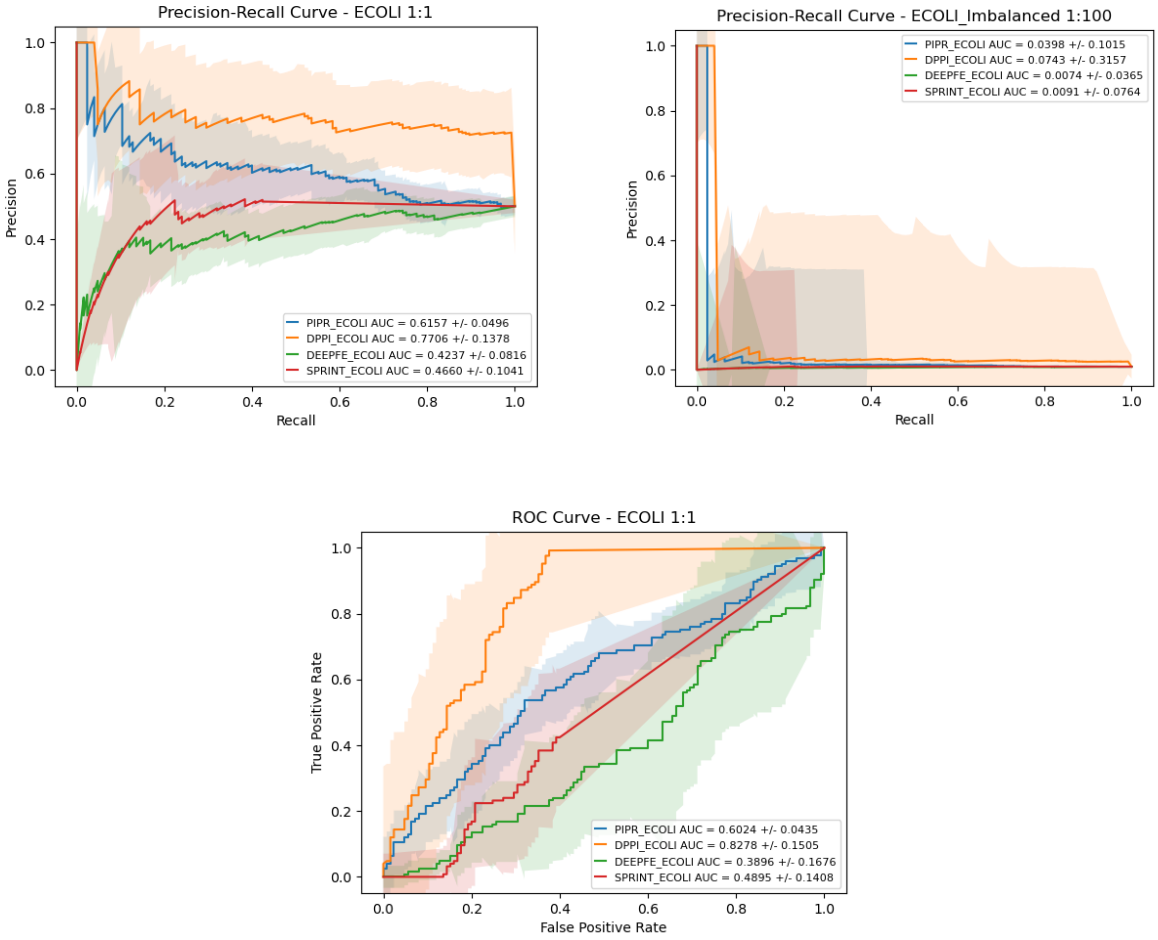

Figure 47: Comparison of base methods on ECOLI dataset using cross-validation.

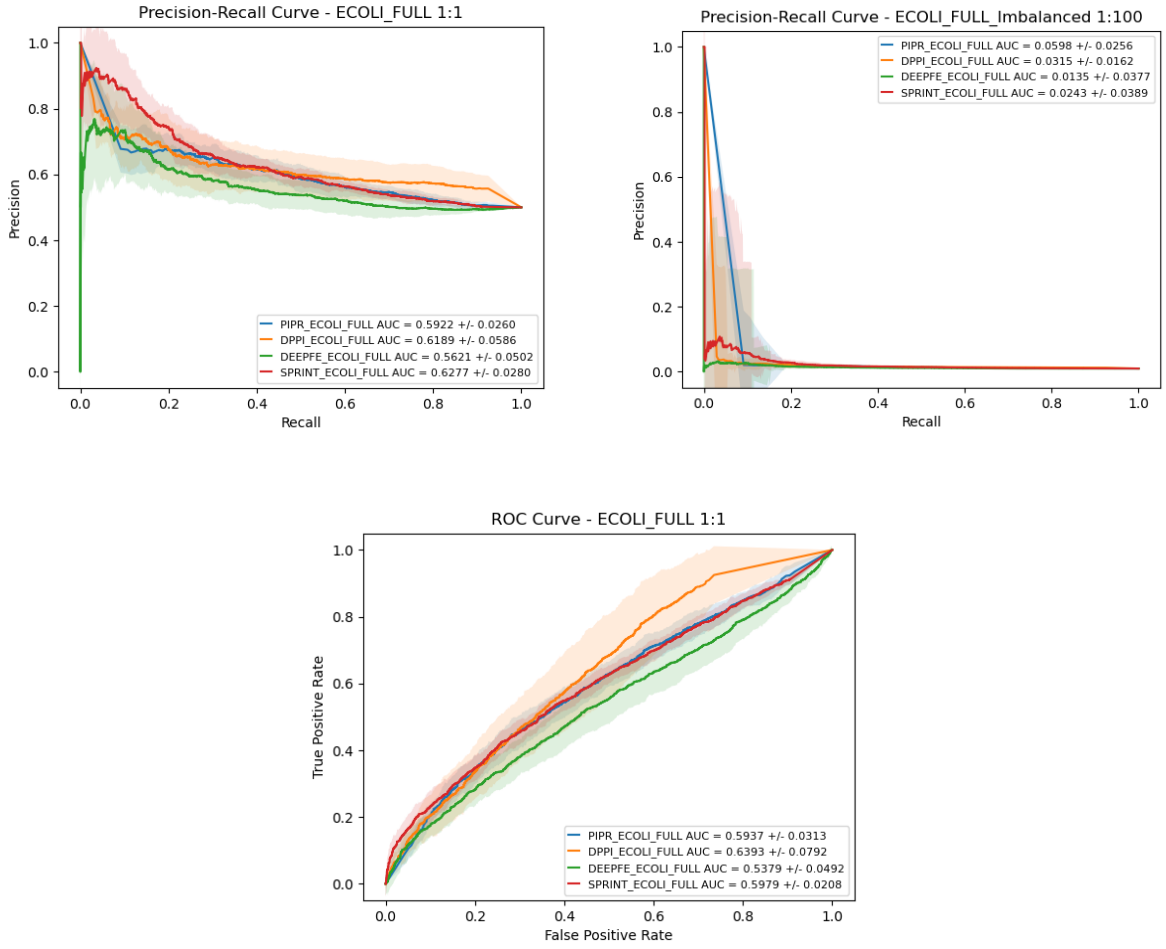

Figure 48: Comparison of base methods on ECOLI_FULL dataset using cross-validation.

In conclusion, SPRINT performed best when desiring high precision at the expense of recall which is the case for correctly identifying PPIs, a problem that is characterized by high class imbalance. Table 10 below shows which methods were significantly different from each other using auPR for ECOLI_FULL results. The values should be read left to right such that a negative F value indicates worse performance for the predictor in the row (left) compared to that in the column (top). An ANOVA test was performed and suggests that a significant difference is found among all methods, but only for an alpha value of 0.05, not for an alpha of 0.01; to be specific, the F-statistic is 3.923 and the p-

value is 0.0160. Therefore, the pairwise t-tests below can be more strictly assessed by accepting significant differences (bold) to be concluded for alpha of 0.01. PIPR and DEEPFE showed a significant decrease in performance relative to SPRINT. All other comparisons indicate no preference of method used to predict PPIs.

Table 10: T-test statistics comparing auPR of methods on ECOLI_FULL.

	SPRINT	DPPI	DEEPFE
DPPI	F = -0.501 p = 0.629		
DEEPFE	F = -4.256 **p = 0.002**	F = -2.714 p = 0.0238	
PIPR	F = -4.136 **p = 0.003**	F = -1.402 p = 0.194	F = 1.228 p = 0.250

While these results do suggest a most performant model for PPI prediction in *E. coli*, the achievable results are still far below those of other species, such as human and yeast. The results in the presence of class imbalance show that precision is limited to ~10%, meaning that only approximately one in ten positive predictions will be true. Clearly, there is a need to develop improved PPI prediction methods for *E. coli*. The following chapter examines the use of reciprocal perspective, a method recently developed here at Carleton University, to improve each of the methods explored in this chapter, both individually and in combination.

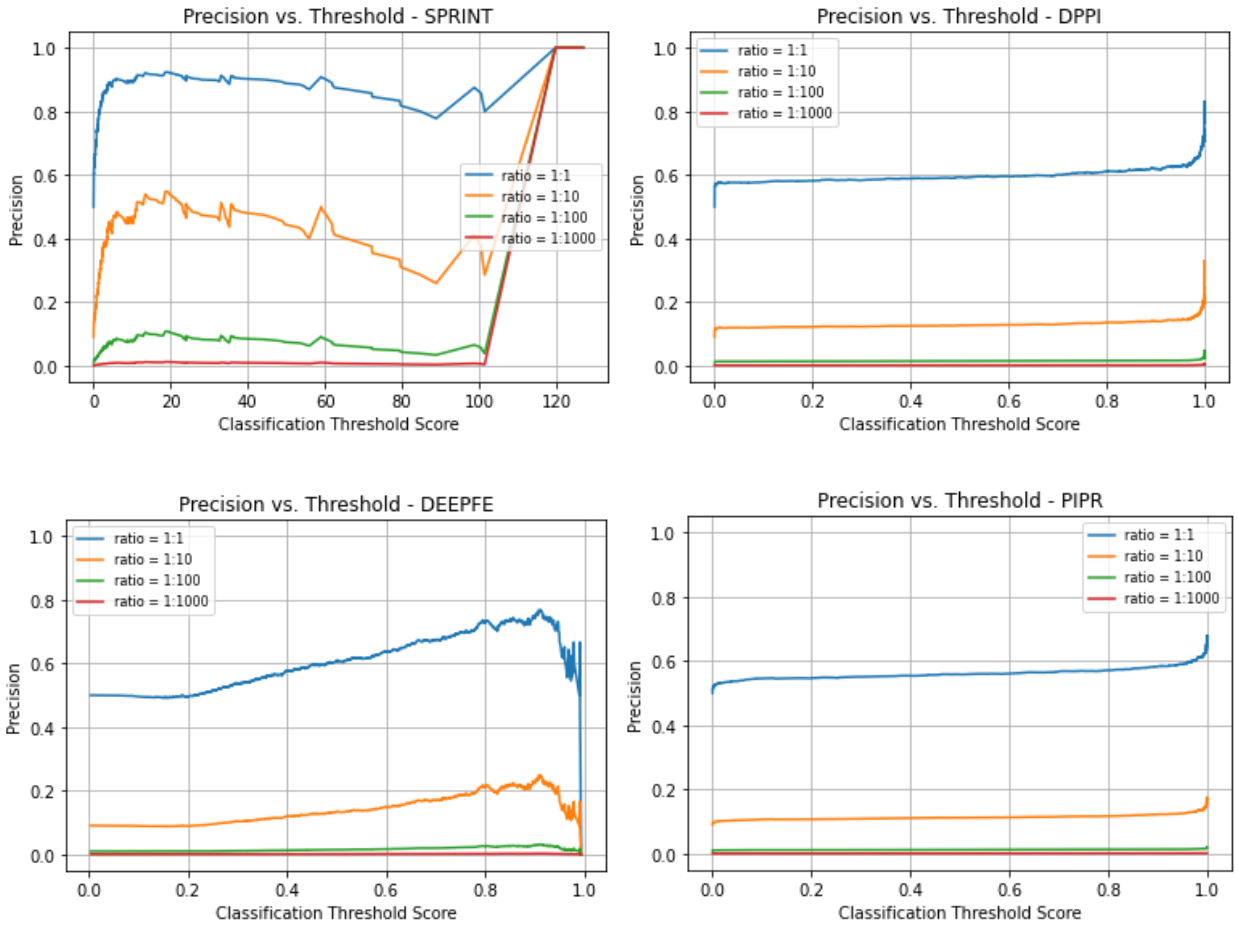

Figure 49: Precision vs. classification threshold using different class imbalance ratios for base classifiers.

In terms of obtaining useful predictions from a model, one can select PPIs that score above a classification threshold that provides the highest precision. This precision would define the probability that the predicted PPI is a true positive and could be confirmed by laboratory experiments. Figure 49 above displays the precision versus score threshold for each classifier at varied class imbalance ratios. These curves can be interpreted such that the precision at a given threshold is the highest probability of a predicted PPI being true. For example, using DEEPFE's curves (left bottom), if 100 PPI have been predicted and scored above 0.8 then about 70 of them would be correctly predicted assuming a 1:1 class imbalance (blue curve) or 20 would be correctly predicted assuming a 1:10 class

imbalance (yellow curve). Based on these results, each classifier except for SPRINT has very low precision at any threshold if an imbalance greater than 1:10 is hypothesized. Therefore, in a practical sense, only SPRINT predictions scored above 100 would provide any reasonable use for researchers to confirm as PPIs.

4 Enhancing PPI Prediction in *E. coli*

The previous section illustrated that state-of-the-art predictors can be relatively weak at making predictions for *E. coli* and presents an opportunity for improvements. Thus, this chapter describes techniques used to enhance PPI prediction models. In the first section, an extension to the reciprocal perspectives (RP) technique for constructing PPI features is presented. The last section presents the use of a multiple classifier system to enhance performance. These two enhancements are investigated and shown to improve PPI prediction capabilities for *E. coli.*

4.1 Implementation of RP

Shown in Figure 50 below are examples of O2A curves generated from SPRINT scores. Instead of using LOESS to obtain knee locations, RP is reimplemented in this thesis using the Python package Kneed [84] to detect the positions of curvature in the O2A curves. The knee position is indicated by the vertical magenta line. Note that two curves are shown in each chart, one for each protein in the pair. Green, red, and grey circles represent pairs for which the protein pair is labeled as a known positive, “known” negative, or unknown pair. For the known positive PPI, both proteins have been scored highly among all other interaction scores and are located to the left of the knees (magenta). This would correctly indicate that this PPI certainly interacts. For the negative pairs, one protein scored higher than the other on their respective O2A scoring curves, but both have relatively low scores and remain to the right of the knees for their respective O2A curves. The unknown pair has one protein ranked to the left of the knee and its paired protein to the right of its respective knee, which indicates some uncertainty if those

proteins are likely to interact and would require quantitative features to further predict a likelihood of interaction.

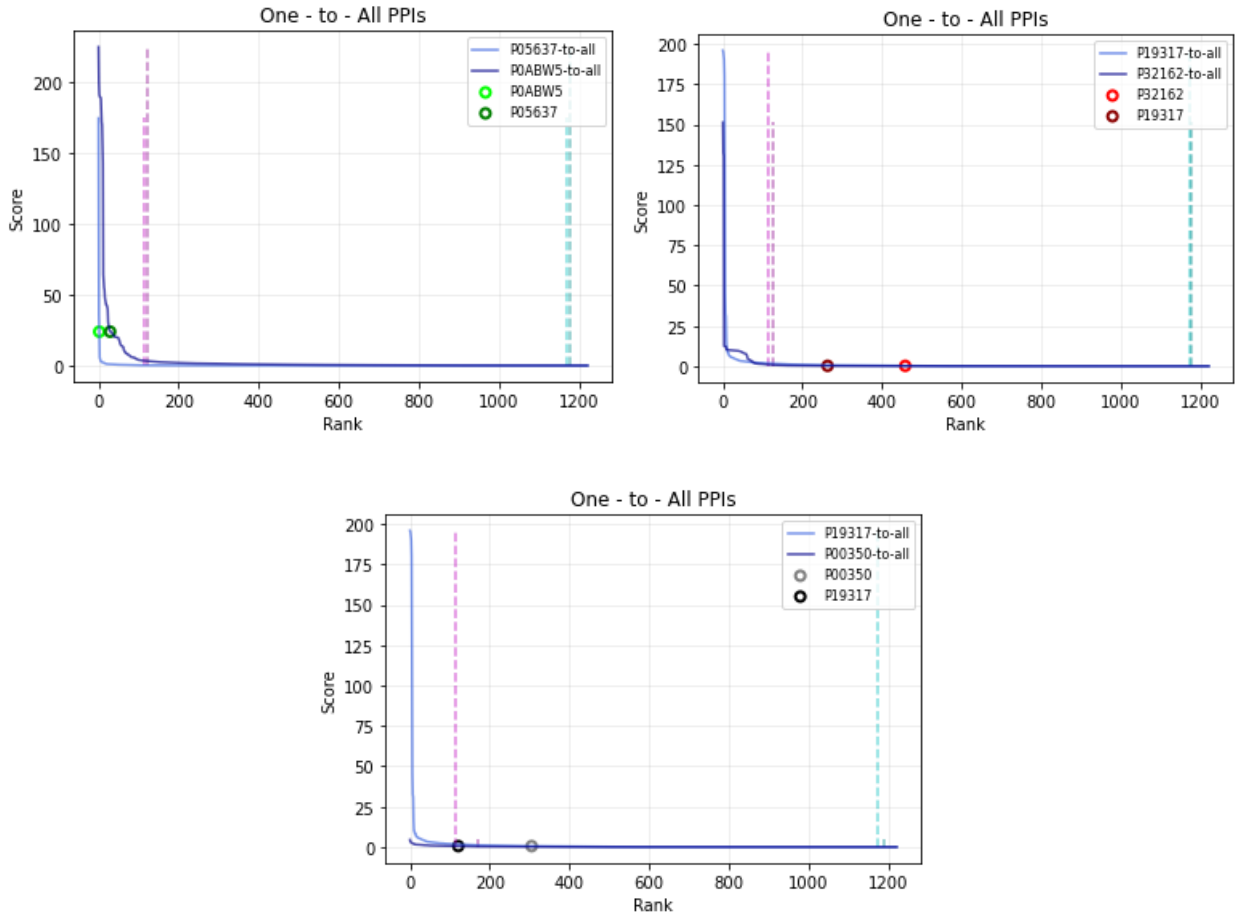

Figure 50: O2A curves from SPRINT scores for positive (green), negative (red), and unknown (grey) PPIs.

For machine learning methods such as DPPI, DEEPFE, and PIPR that produce probability scores, examples of their O2A curves can be seen in Figure 51. These probabilistic scoring methods more often produce characteristic S-shaped curves, unlike many SPRINT curves, where the scores are unbounded more often resulting in L-shaped curves. Therefore, 12 additional features presented in Table 11 can be extracted from positional relationships with the curve's elbow (cyan) and may benefit methods such as DPPI, DEEPFE, and PIPR for RP enhancement. Noticeably, the locations of knees and

elbows are more precise here than those detected in SPRINT's scoring curves using Kneed. Similar to above, positive PPIs are determined when both proteins are found to the left of the knee, negative pairs are now better determined when both are found to the right of the elbow, and unknown PPIs would be found between the knee and elbow or if each protein is in found in separate locations on their reciprocal curve.

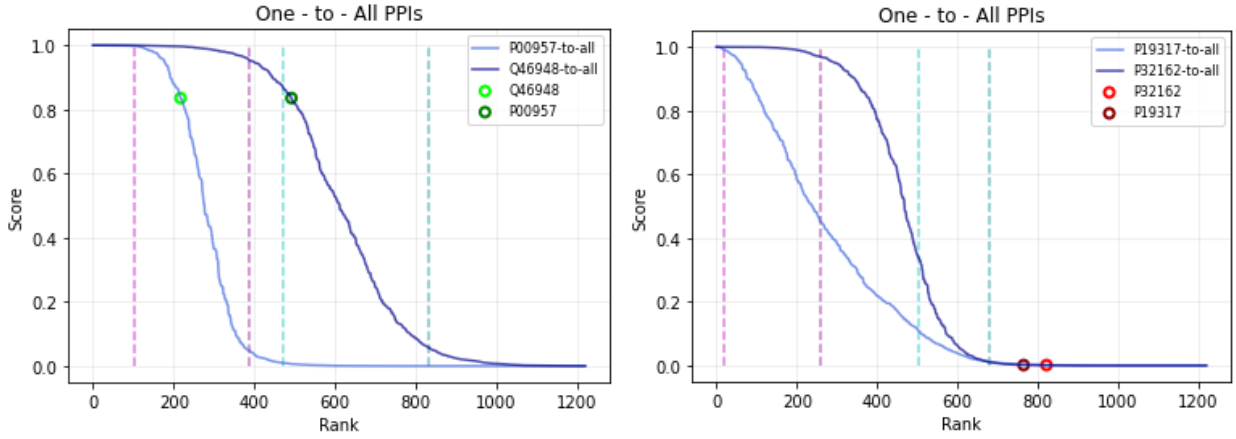

Figure 51: O2A curves from PIPR displaying S-shaped characteristics.

Table 11: Extended RP Features Developed in this Thesis

Feature Number	Description	Datatype, Range of Values
3	Score of A in B's curve ($scoreA_b$)	float, [0, maximum score]
4	Score of B in A's curve ($scoreB_a$)	float, [0, maximum score]
14	Rank of elbow in A's curve	int, [1, # of proteins]
15	Rank of elbow in B's curve	int, [1, # of proteins]
16	Score of elbow in A's curve	float, [0, maximum score]
17	Score of elbow in B's curve	float, [0, maximum score]
20	Rank of B on A's curve is above (left of) elbow	bool, [0 or 1]
21	Rank of A on B's curve is above (left of) elbow	bool, [0 or 1]
22	Scores of A and B both above global mean score	bool, [0 or 1]
23	Scores of A and B both above global median	bool, [0 or 1]
26	Fold Difference A knee (score of B in A's curve from elbow normalized)	float, [-1, 1]
27	Fold Difference B knee (score of A in B's curve from elbow normalized)	float, [-1, 1]

Besides accurate detection of knees/elbows, a factor that can affect PPI predictions based on RP features is the proteome size used to generate scoring curves. In the above examples, only the 1,221 proteins from the ECOLI_FULL dataset were used to generate scoring curves. Generating scoring curves from the proteome of 4,438 proteins can produce a more detailed perspective as shown in the RP curves in Figure 52. In this example, increasing the number of proteins used in scoring would change the unknown

PPI from a negative classification to a positive one. Therefore, a more complete scoring curve provides greater O2A context from which to extract RP features.

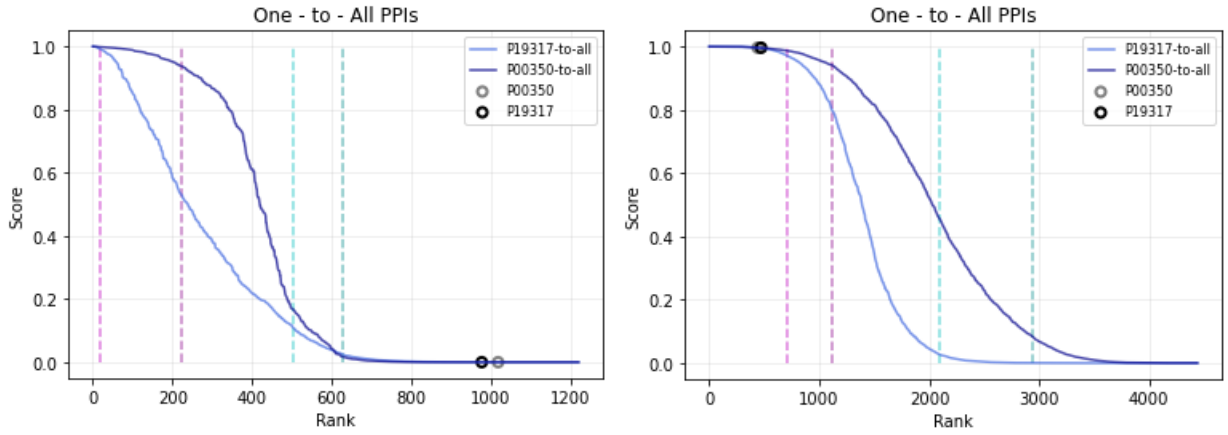

Figure 52: Comparing O2A scoring curves generated from only dataset proteins versus the entire proteome.

4.1.1 Applying RP to State-of-the-Art Methods

To estimate the performance of RP-enhanced models, a 10-CV scheme presented in Figure 53 is implemented. The dataset was split into 10 train/test subsets containing equal number of positives and negatives. These subsets are identical to the subsets used in the base classifier cross-validations. For each subset, the base classifier is provided the training data to make all-to-all predictions (i.e., predict a score for every possible pairing of proteins within the training set). Based on the O2A curve generated strictly from the training data, RP features are then extracted from the predictions for PPIs in the training data and testing data. Then the meta-classifier is provided with the RP training data and makes predictions on the RP testing data. Finally, the RP-enhanced PPI predictions over the test set can be evaluated and an average performance over all subsets is reported.

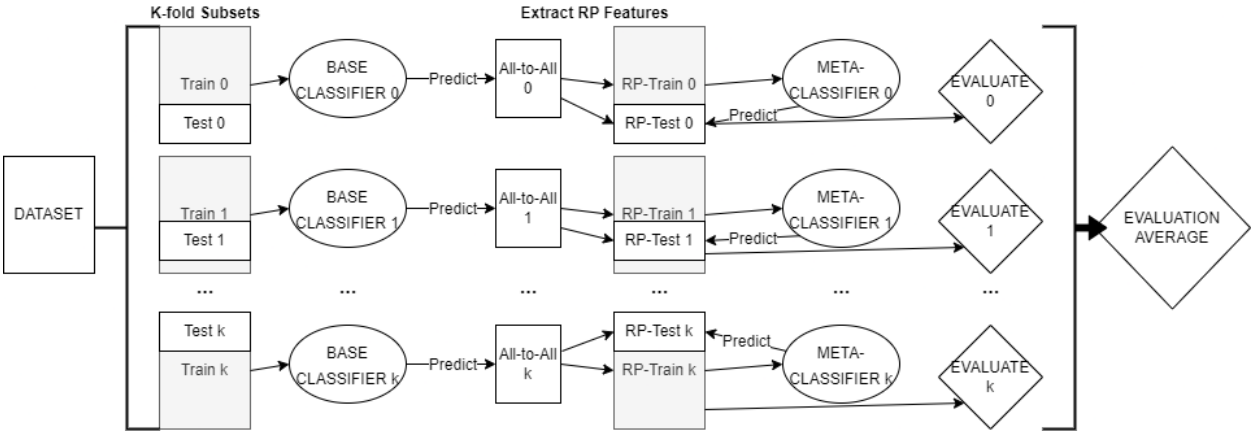

Figure 53: Procedure for implementing RP-enhanced cross-validation.

Unlike in [13], the meta-classifier used here is an ensemble decision tree-based gradient-boosting machine (GBM). Specifically, this was implemented using LightGBM [85]. LightGBM is a recent GBM technique that builds leaf-wise trees instead of level-wise tree to improve training efficiency. Note that a random forest, XGBoost, and SVM classifiers were also explored with non-exhaustive grid searches for parameter tuning, using 10-CV of ECOLI_FULL and auPR, to be used as the meta-classifier but were found to be outperformed by a LightGBM model. After performing a grid search of parameters for the LightGBM model (see Appendix B, Table 16), optimal parameters that produced the highest auPR were Gradient-base One-Side Sampling (GOSS) as the boosting method, 150 trees with a maximum depth of 10, 50 leaves per tree, minimum data in leaf of 50, a learning rate of 0.1, 0.1 smoothing, and the remaining default parameters.

4.1.2 RP-Enhancement Results

The results displayed by Figure 54, Figure 56, Figure 58, and Figure 60 below compare each base classifier (blue) performance to the RP-enhanced classifier (orange) performance in the 10-CV evaluations on ECOLI_FULL. Note that the RP features here

were extracted using O2A scoring curves of only the 1,221 proteins in the ECOLI_FULL dataset with the assumption that they were representative of the proteome as described by Figure 20. In fact, little or no significant change to performance was seen in RP enhancement when using the entire proteome for RP feature extraction (see Appendix C, Figure 75, Figure 76, Figure 77, Figure 78), suggesting ECOLI_FULL provided enough resolution for accurate RP features. Below each set of performance curves are the contributions of the 27 features to building the meta-classifier where the feature importance was determined by the number of times a feature, or node in the decision trees, was used in training the model.

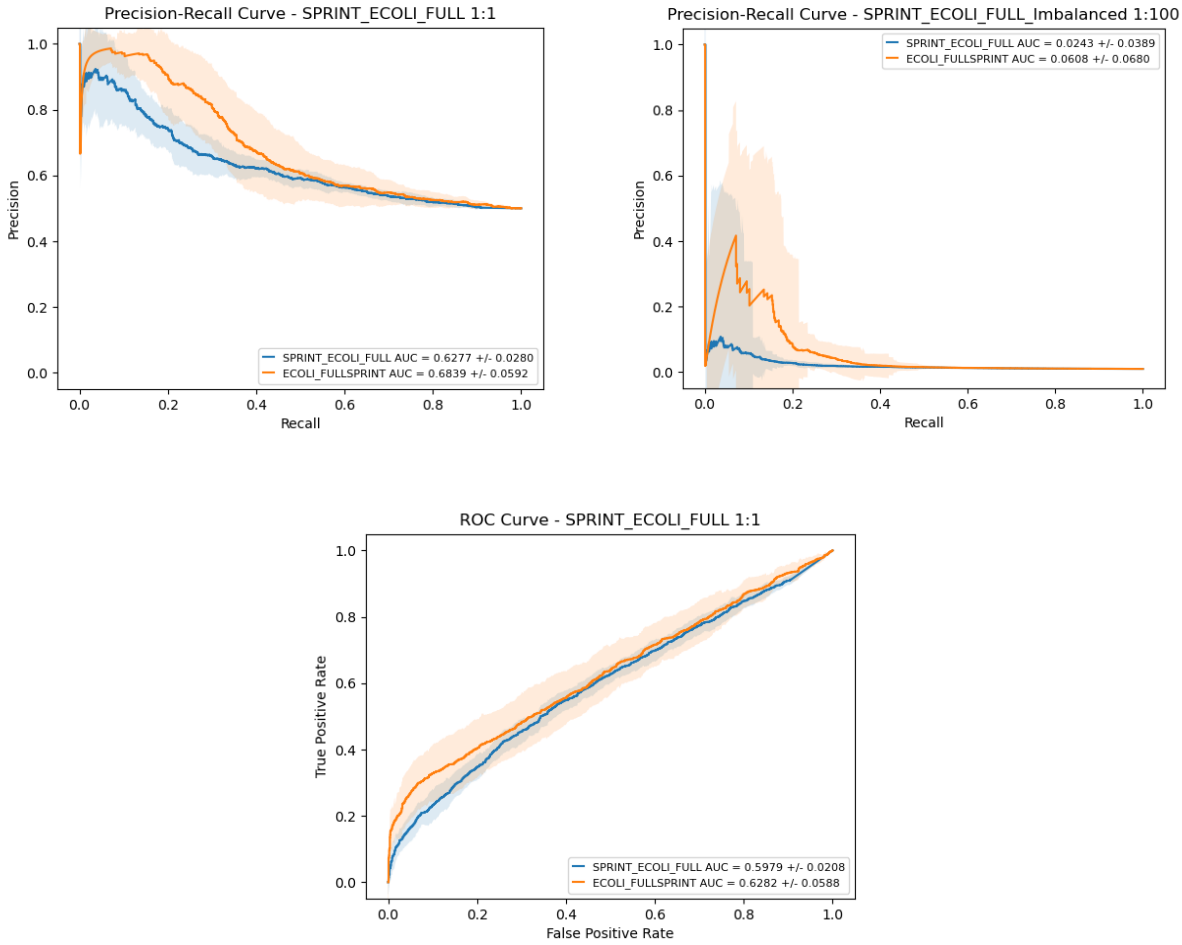

Figure 54: RP-enhancement of SPRINT.

Using a 10-CV with the ECOLI_FULL dataset, the auPR for SPRINT was 0.6277 ± 0.0280, which increased to 0.6839 ± 0.0592 using RP-enhancement, for balanced evaluations. For class-imbalanced evaluation, the auPR increased from 0.0243 ± 0.0389 to 0.0608 ± 0.0680 after RP-enhancement. The auROC was also increased from 0.5979 ± 0.0208 to 0.6282 ± 0.0588 after RP-enhancement. Of the 27 RP features, 5 features had no contribution to building the model. The additional elbow-based features developed in this thesis can be seen to contribute substantially to SPRINT's RP-enhanced model.

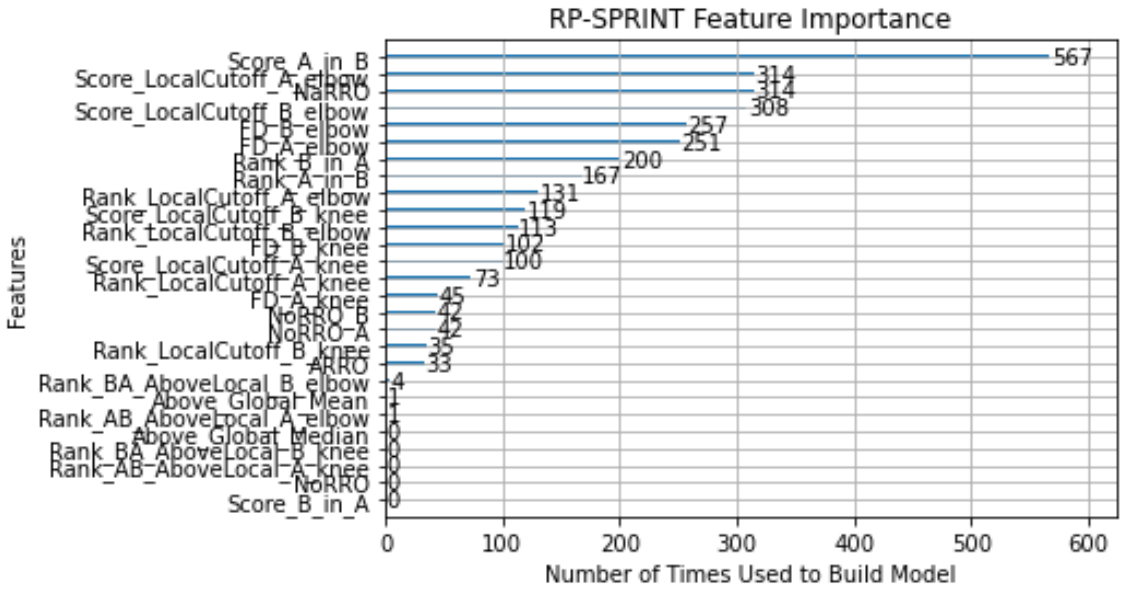

Figure 55: Contribution of RP features to RP-SPRINT model.

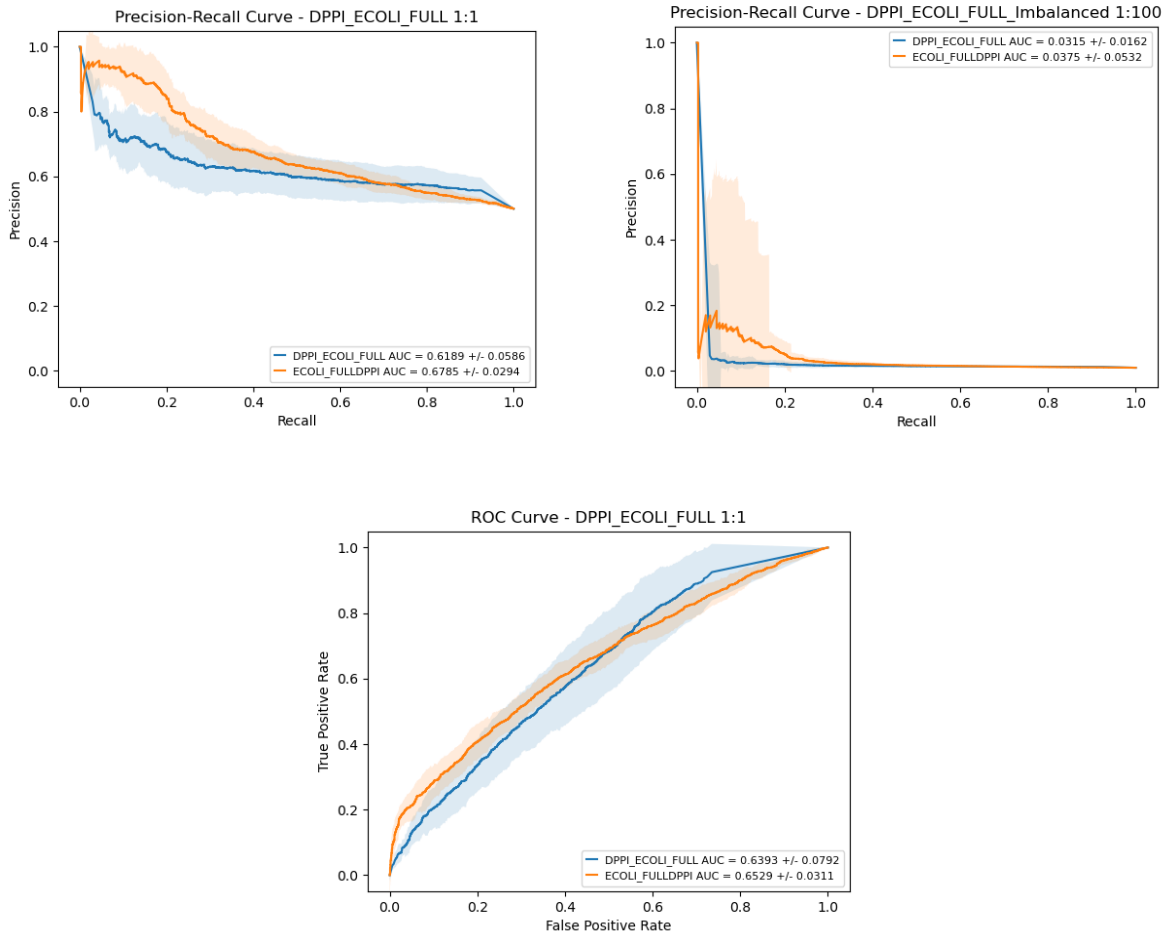

Figure 56: RP-enhancement of DPPI.

Using a 10-CV with the ECOLI_FULL dataset, the auPR for DPPI was 0.6189 ± 0.0586, which increased to 0.6785 ± 0.0294 using RP-enhancement, for balanced evaluations. For class-imbalanced evaluation, the auPR increased from 0.0315 ± 0.0162 to 0.0375 ± 0.0532 after RP-enhancement. The auROC was also increased from 0.6393 ± 0.0792 to 0.6529 ± 0.0311 after RP-enhancement. Of the 27 RP features, 5 features had no contribution to building the model, some different than those for SPRINT's RP-enhanced model. Again, the additional elbow-based features developed in this thesis can be seen to contribute substantially to DPPI's RP-enhanced model.

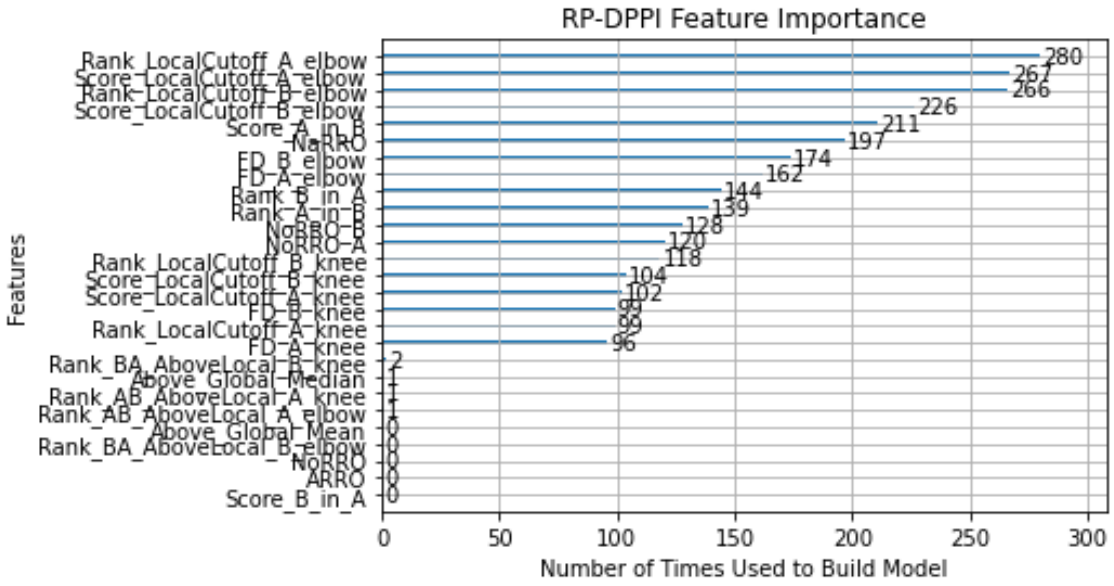

Figure 57: Contribution of RP features to RP-DPPI model.

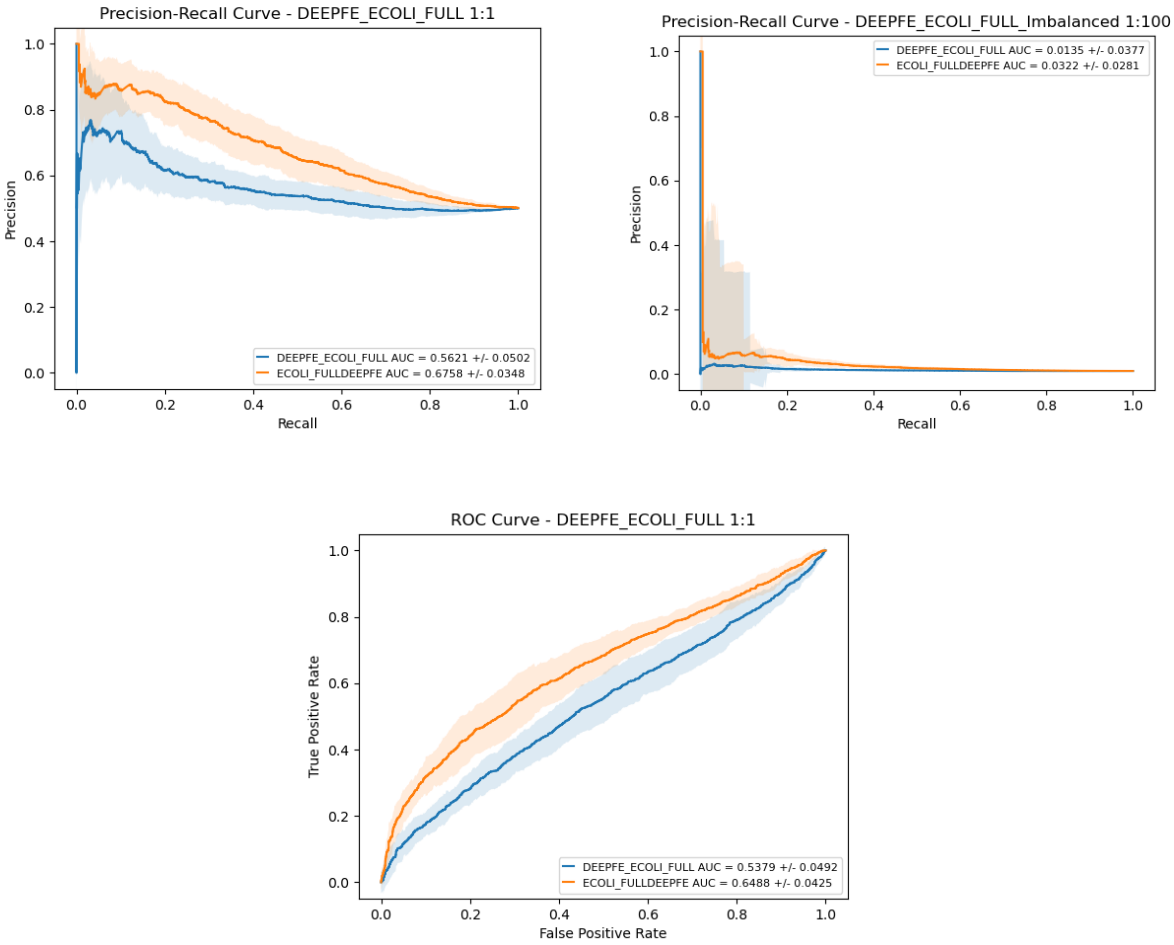

Figure 58: RP-enhancement of DEEPFE.

Using a 10-CV with the ECOLI_FULL dataset, the auPR for DEEPFE was 0.5621 ± 0.0502, which increased to 0.6758 ± 0.0348 using RP-enhancement, for balanced evaluations. For class-imbalanced evaluation, the auPR increased from 0.0135 ± 0.0377 to 0.0322 ± 0.0281 after RP-enhancement. The auROC was also increased from 0.5379 ± 0.0492 to 0.6488 ± 0.0425 after RP-enhancement. Of the 27 RP features, 3 features had no contribution to building the model. As with the previous two models, the additional elbow-based features developed in this thesis can be seen to contribute to PIPR's RP-enhanced model.

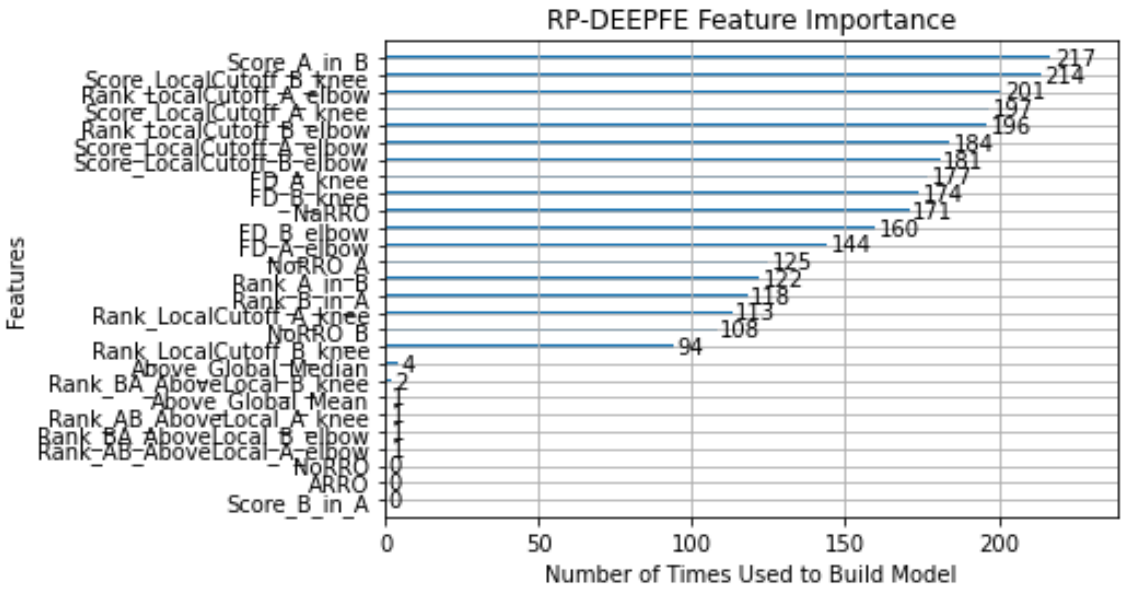

Figure 59: Contribution of RP features to RP-DEEPFE model.

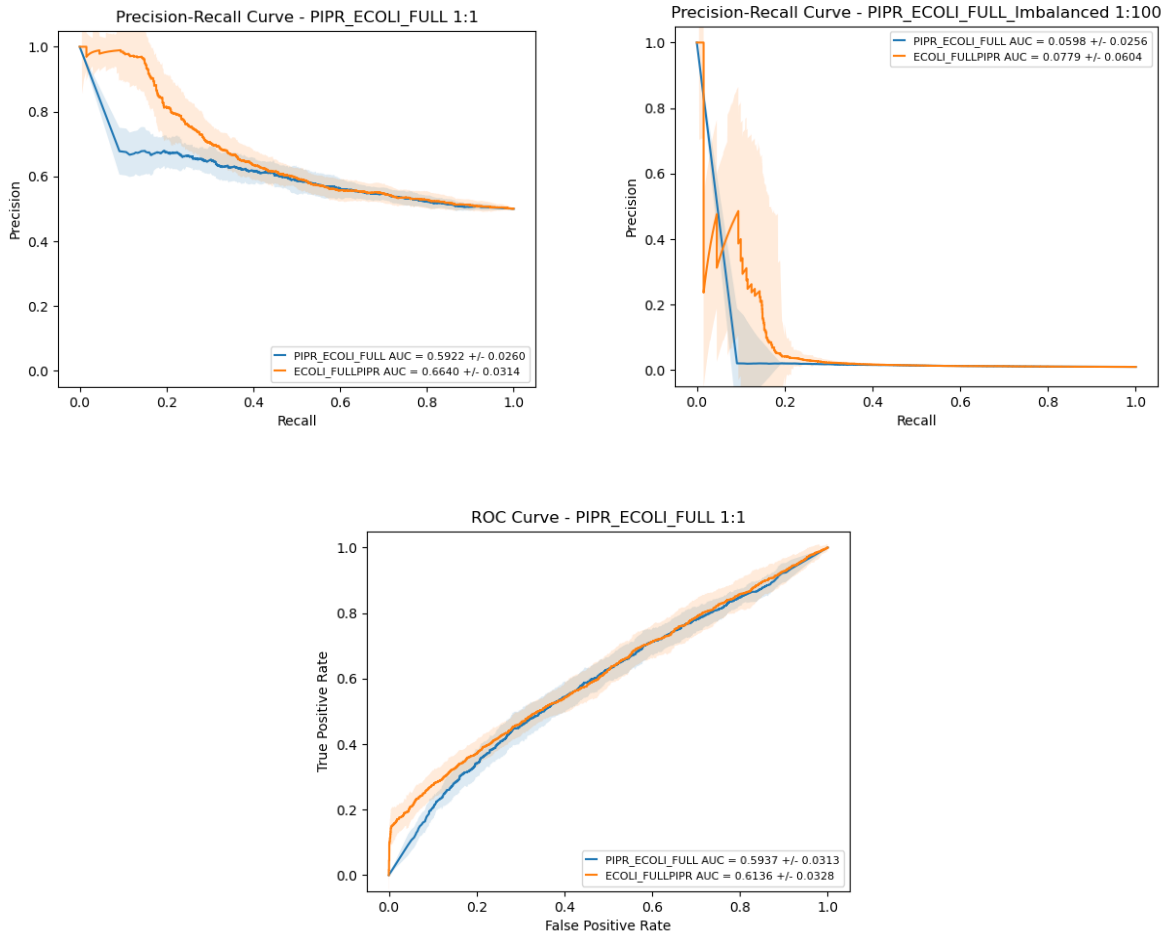

Figure 60: RP-enhancement of PIPR.

Using a 10-CV with the ECOLI_FULL dataset, the auPR for PIPR was 0.5922 ± 0.0260, which increased to 0.6640 ± 0.0314 using RP-enhancement, for balanced evaluations. For class-imbalanced evaluation, the auPR increased from 0.0598 ± 0.0256 to 0.0779 ± 0.0604 after RP-enhancement. The auROC was also increased from 0.5937 ± 0.0313 to 0.6136 ± 0.0328 after RP-enhancement. Of the 27 RP features, 5 features had no contribution to building the model. Again, the additional elbow-based features developed in this thesis can be seen to contribute substantially to PIPR's RP-enhanced model.

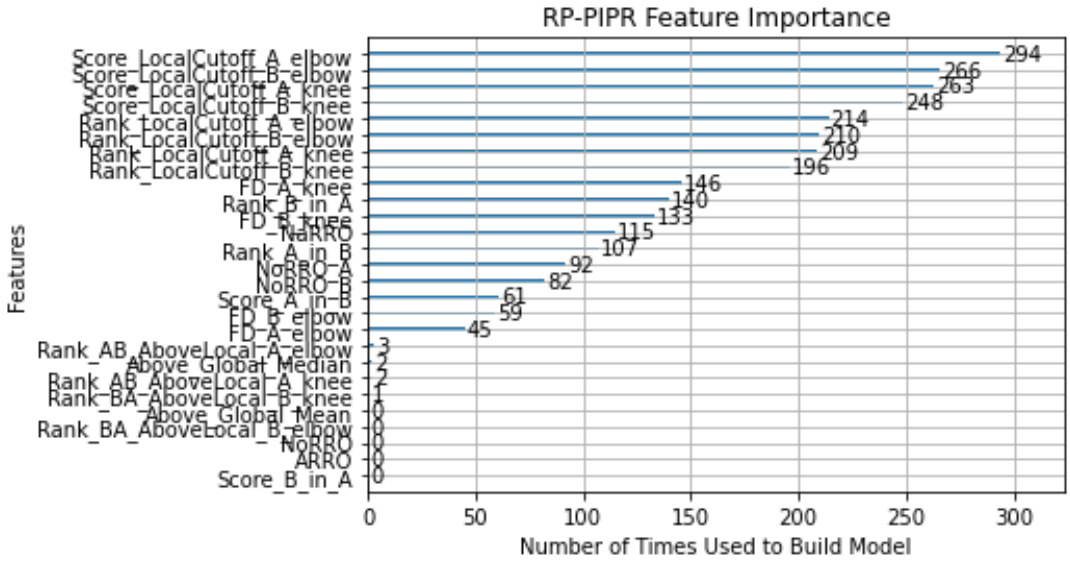

Figure 61: Contribution of RP features to RP-PIPR model.

4.1.3 Discussion and Conclusion

The comparisons above show a substantial improvement in performance for all base classifiers when using an RP approach. This independently validates the RP method as a technique to improve pairwise classifications. Using Kneed to detect knees/elbows and defining new features further proves that applying the concept of RP for enhancement is useful irrespective of how it is implemented. Figure 62 shows that each RP-enhanced method results in similar prediction performance. Subsequent testing with ANOVA confirmed the lack of significant difference among them. It would appear that RP was able to improve the weakest classifiers and make them relatively indistinguishable.

The feature importance graphs displayed above also indicate that RP features from different base classifiers contribute differently to the meta-classifier. This indicates the potential of combining these features into a classification model and is explored in the next section. Some features were found to have no importance across all base classifiers (e.g., “Score_B_in_A”). These features may be unimportant when other features can

provide similar contextual information within the O2A curves. For example, if the protein is ranked with a low value on the O2A curve then it is also implied that the protein is also scored highly among all proteins in that curve.

Furthermore, improvements to RP feature extraction can be investigated such as optimizing knee/elbow detection, defining additional features, feature reduction, and refinements of current features. As an aside, this RP approach could also be extended to higher dimensional spaces (beyond pairwise interactions) for researching interaction prediction for protein complexes that are formed by multiple proteins.

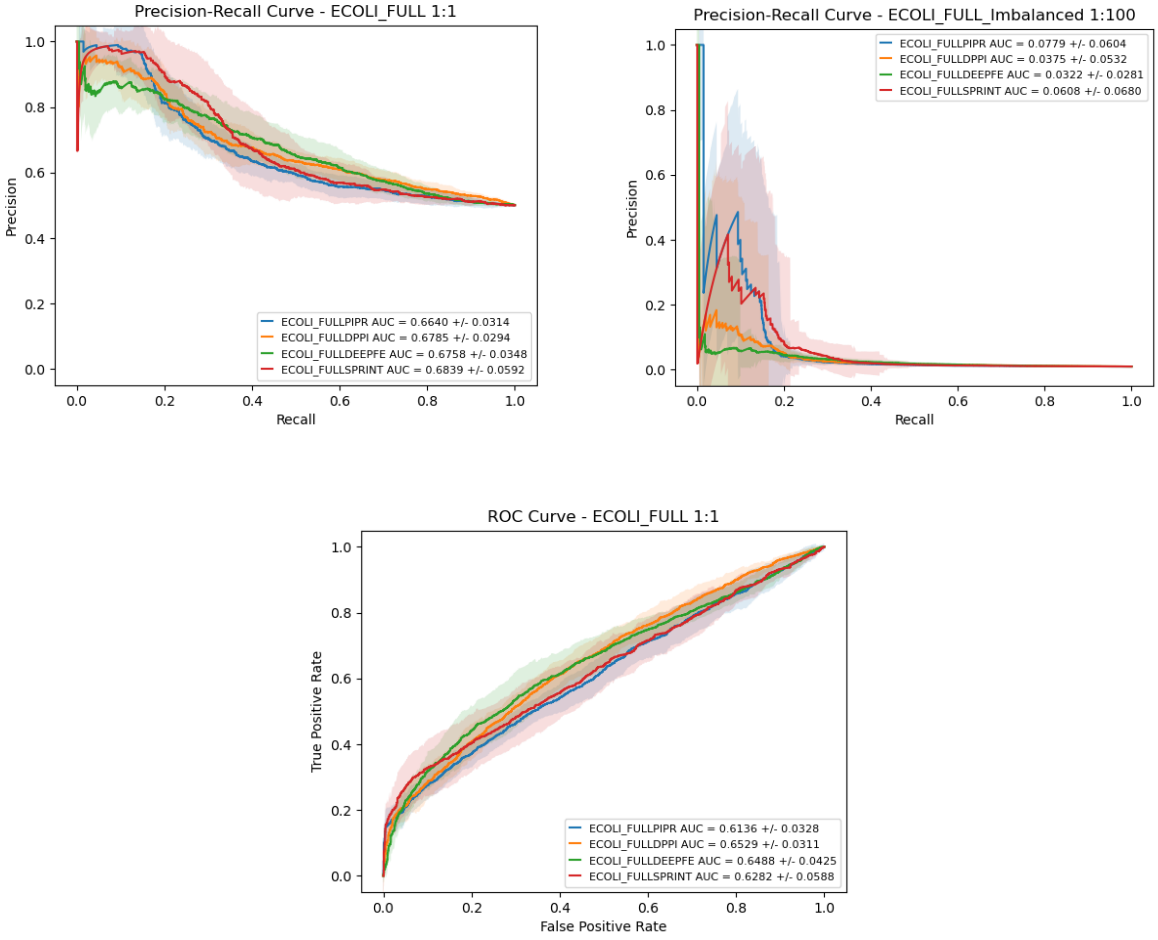

Figure 62: Comparison of RP-enhanced base classifiers.

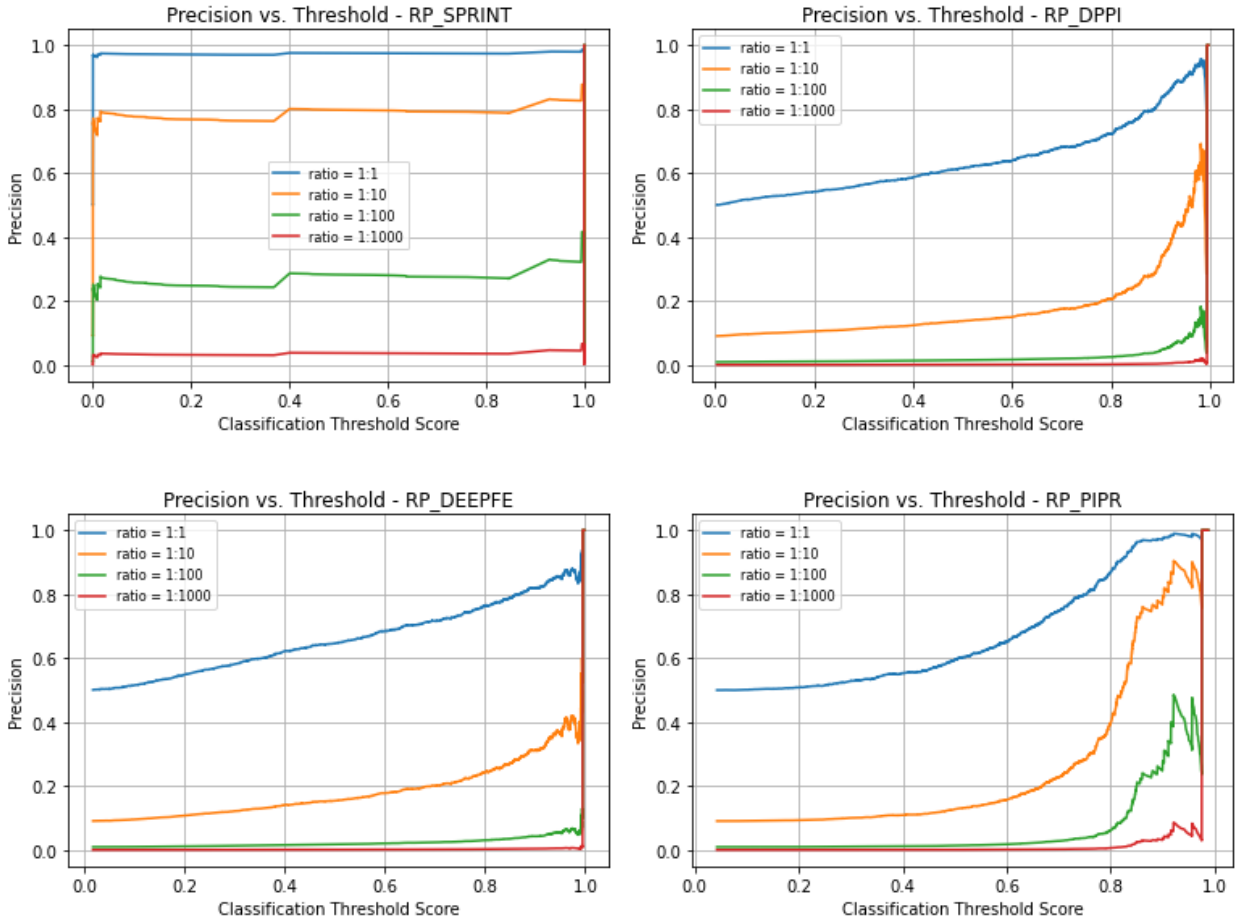

Figure 63: Precision vs. classification threshold using different class imbalance ratios for RP-enhanced classifiers.

The RP-enhanced precision versus threshold curves shown in Figure 63 illustrate improvement compared to their respective base classifier curves in Figure 49 above. Notably, RP-enhanced predictions become more practical to test for experimental confirmation and accept as likely PPI. However, only RP-SPRINT and RP-PIPR would produce high scoring predictions that would be about 30-40% true if a 1:100 class imbalance is hypothesized to exist.

4.2 Multiple Classifier System

To further explore improvements in PPI prediction, a combination of multiple experts (CME) approach was investigated. Ensemble approaches such as hard voting, soft voting/averaging, bagging, boosting, and stacking are popular approaches to consolidate weak learners into a high performing classifier. Typically, effective CME only requires that the "weak learners" perform better than average. Here, we have four such weak learners.

In particular, stacking and soft voting can be easily implemented without requiring extensive time and computational resources to run and evaluate each classifier approach. A third novel approach is to average the RP features produced by each base classifier to obtain a combined reciprocal perspective from each classifier. Therefore, these three CME techniques have been explored in this thesis.

4.2.1 Implementing a Combination of Multiple Experts Approach

In soft voting, the predictions from each model are averaged to make a final prediction. The use of averaging this way improves prediction stability by accounting for each RP model's predictions before making the final PPI classification. The overall scheme for applying soft voting using the four RP model predictions is in Figure 64.

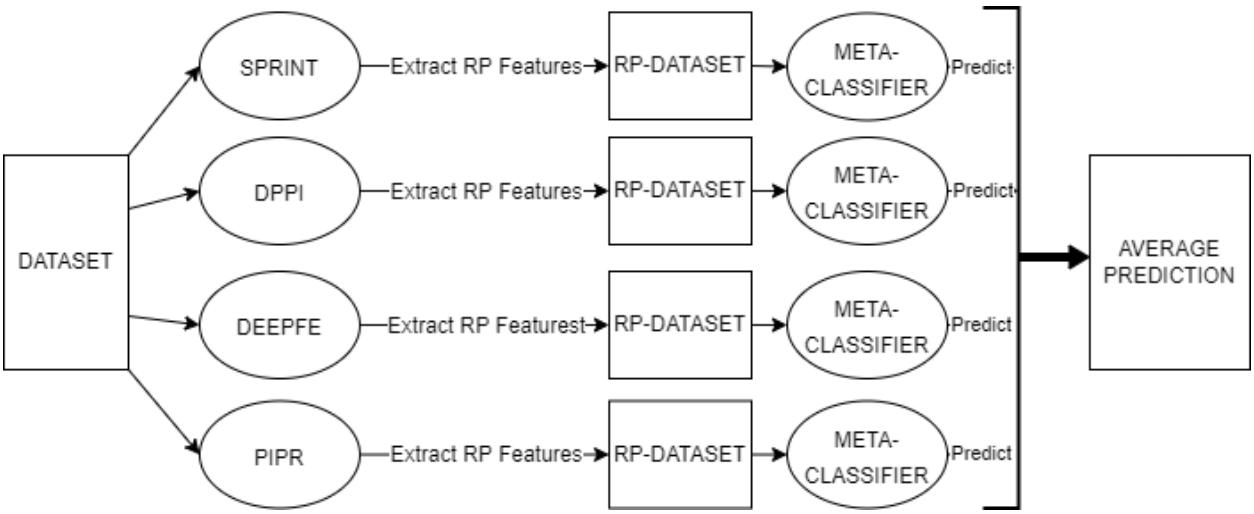

Figure 64: Architecture for soft-voting multi-classifier system.

Stacking classifiers in this work involves concatenating the RP features for each PPI from each RP model as in Figure 65. It is then expected that better predictions can be made by incorporating each model's RP representation for each PPI before training the meta-classifier. Therefore, instead of providing 27 RP features for each PPI, stacking these RP models will provide the meta-classifier with 108 features.

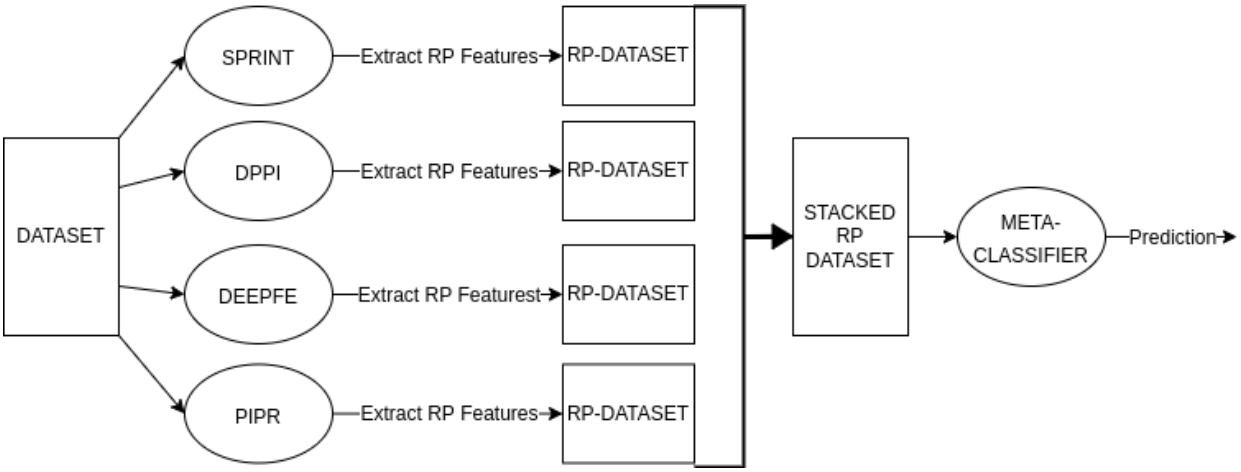

Figure 65: Architecture of stacked multi-classifier system.

The third CME approach presented in Figure 66 takes the average of RP feature values to account for differences in each model's RP representation of protein pairs. As

seen in Section 4.1 above, different base classifiers can produce different O2A curves. In some instances, such curves may exhibit weakly defined knees/elbows. Thus, averaging RP features should reduce variance in the RP feature space and provide more defined values for the meta-classifier to learn from and make predictions.

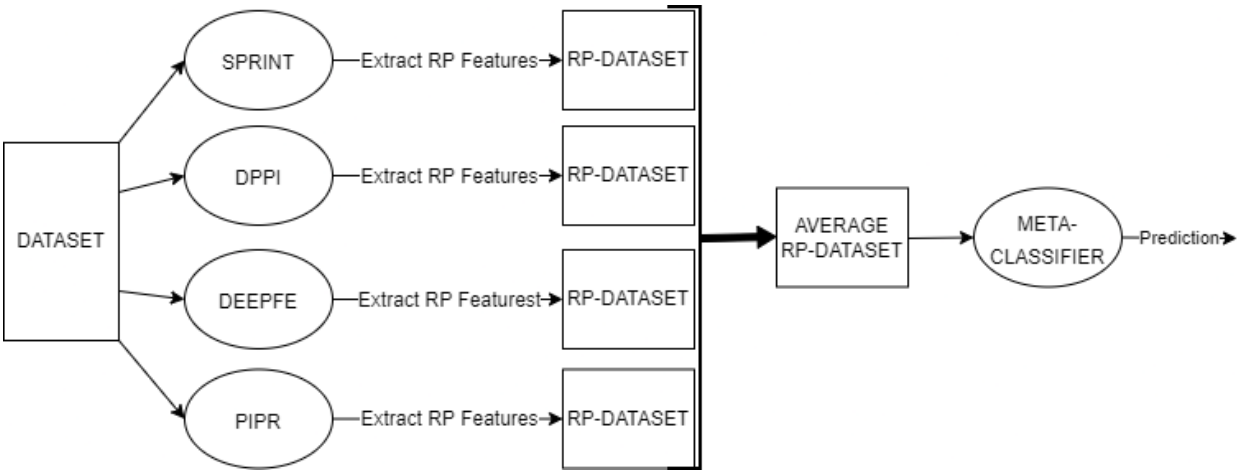

Figure 66: Architecture for RP-averaged multi-classifier system.

4.2.2 Results for CME Methods

Each of the CME schemes presented above were implemented using the same LightGBM classifier used in the RP-enhancements from the previous section. Figure 67 below shows the resulting performance of each CME scheme using a 10-CV evaluation on the ECOLI_FULL dataset. There was no statistically significant difference found among the three CME schemes using LightGBM. The auPR for stacked, average RP, and soft vote averaging schemes were 0.7117 ± 0.0446, 0.7037 ± 0.0492, and 0.7207 ± 0.0349, respectively, for balanced evaluations. For class-imbalanced evaluation, the auPR were 0.0753 ± 0.0551, 0.0671 ± 0.0493, and 0.0980 ± 0.0623, respectively. The auROC for each was 0.6776 ± 0.0387, 0.6604 ± 0.0473, and 0.6804 ± 0.0310, respectively.

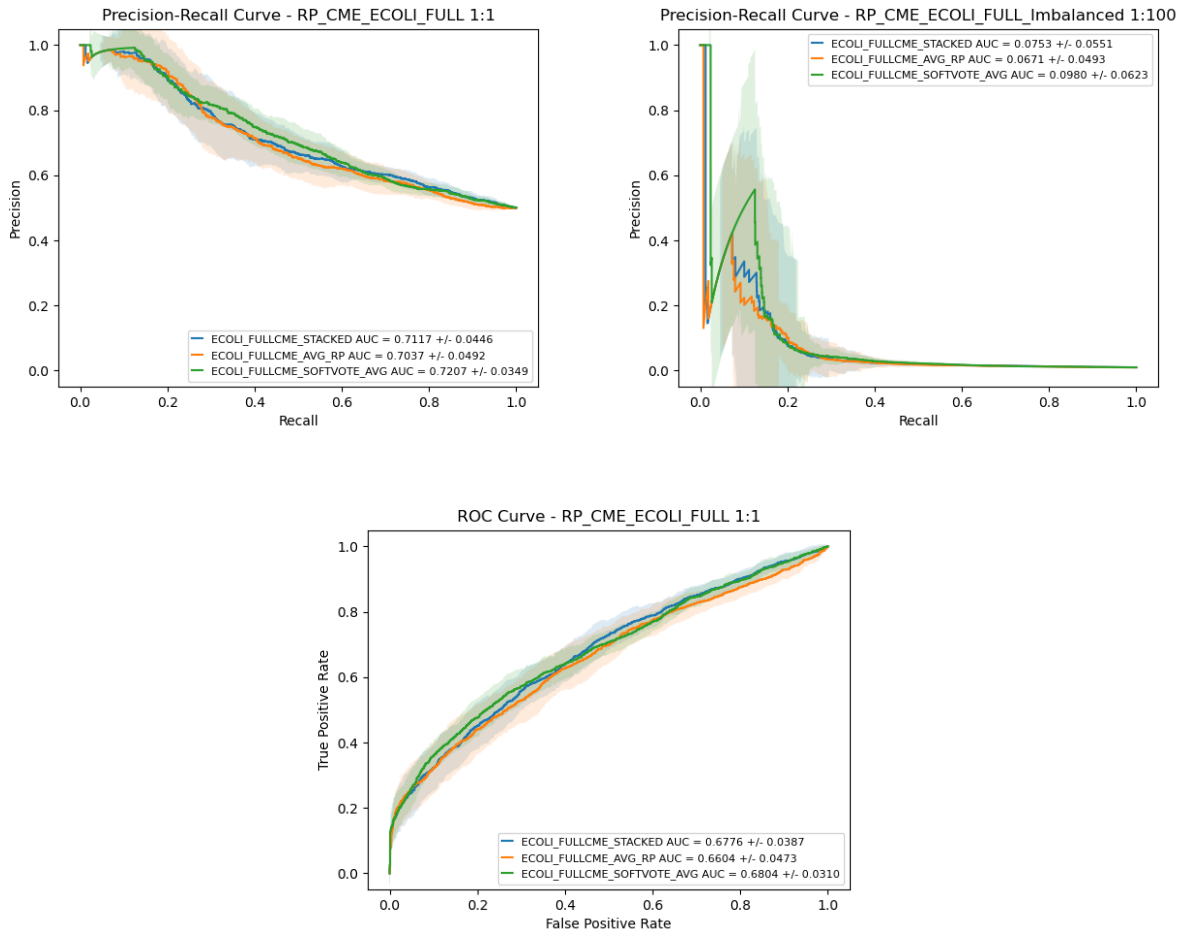

Figure 67: Comparison of the three CME schemes using a LightGBM meta-classifier.

A support vector classifier (SVC) model was also explored for use as the meta-classifier in each CME scheme. A grid search was performed for the SVC model (see Appendix B, Table 17) which found best auPR results from 10-CV using a sigmoid kernel, gamma set to ‘scale’, and a C value of 0.6 for parameters. A 10-CV evaluation using ECOLI_FULL is shown by the performance curves in Figure 68. The auPR for stacked, average RP, and soft vote averaging schemes were 0.7320 ± 0.0393, 0.7104 ± 0.0289, and 0.6780 ± 0.0298, respectively, for balanced evaluations. For class-imbalanced evaluation, the auPR were 0.1142 ± 0.0675, 0.0843 ± 0.0721, and 0.1109 ± 0.0421,

respectively. The auROC for each was 0.6886 ± 0.0325, 0.6640 ± 0.0289, and 0.6318 ± 0.0333, respectively.

Shown in Table 12, soft vote averaging was significantly overperformed by the RP-stacking and averaging RP CME approaches when used in a balanced evaluation. However, no CME scheme was significantly different from one another when evaluated under class-imbalance conditions.

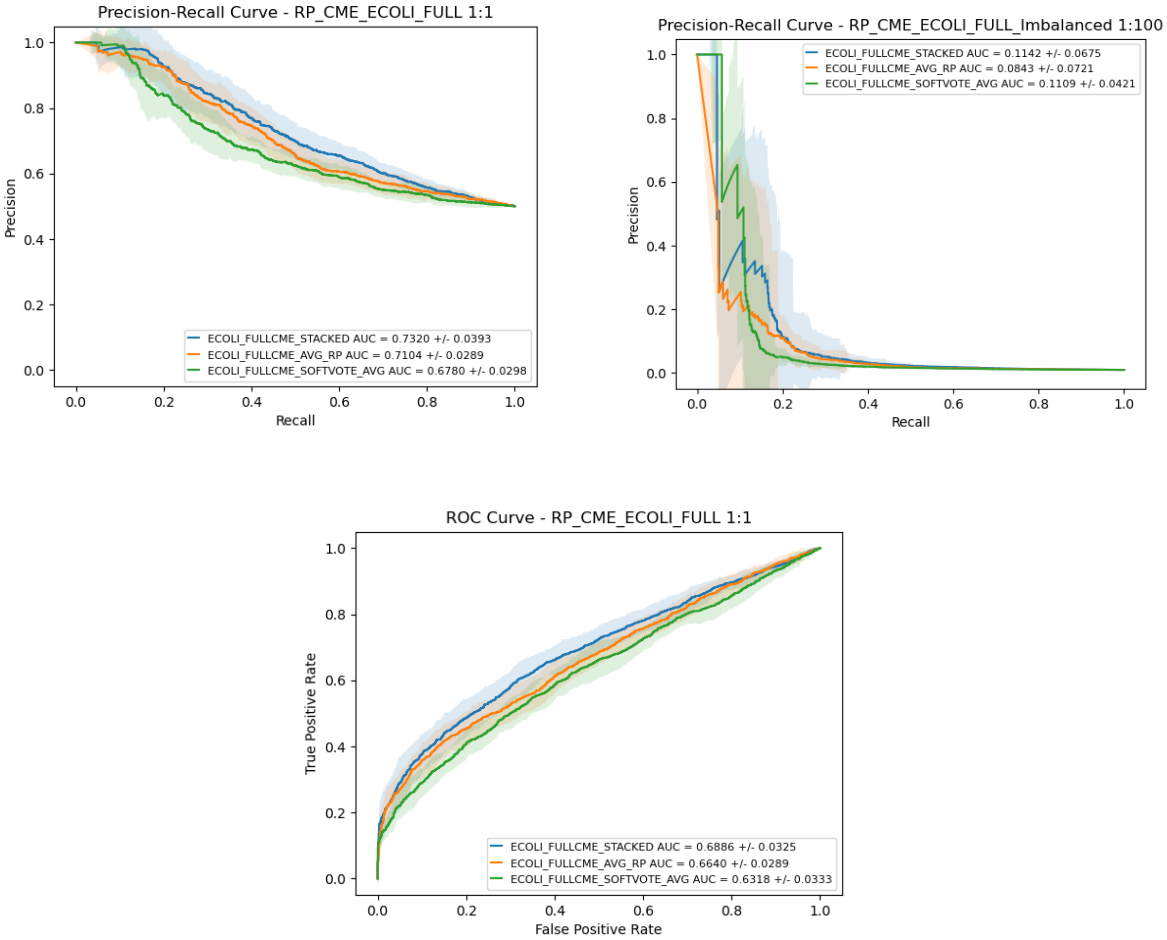

Figure 68: Comparison of CME schemes using an SVC meta-classifier.

Table 12: T-test statistics for SVC CME schemes using balanced class evaluation.

	Soft-Vote Averaged	RP-Stacked
RP-Stacked	F = 4.427 **p = 0.002**	
RP-Averaged	F = 4.114 **p = 0.003**	F = -1.380 P = 0.201

To further investigate the benefits of RP for use in a CME approach, a performance comparison of two CME schemes was conducted. In the first scheme, predictions made by all four base classifiers were combined by soft-vote averaging. In the second scheme, predictions made by all four RP-enhanced classifiers were combined by soft-vote averaging. The results shown in Figure 69 further exemplify the advantage of using RP enhancement in a CME approach over only using base classifiers. The auPR for the base CME and RP CME was 0.6498 ± 0.0180 and 0.6780 ± 0.0298, respectively, for balanced evaluations. For class-imbalanced evaluation, the auPR was 0.0281 ± 0.0128 and 0.1109 ± 0.0421, respectively. The auROC for each was 0.6495 ± 0.0184 and 0.6318 ± 0.0333, respectively.

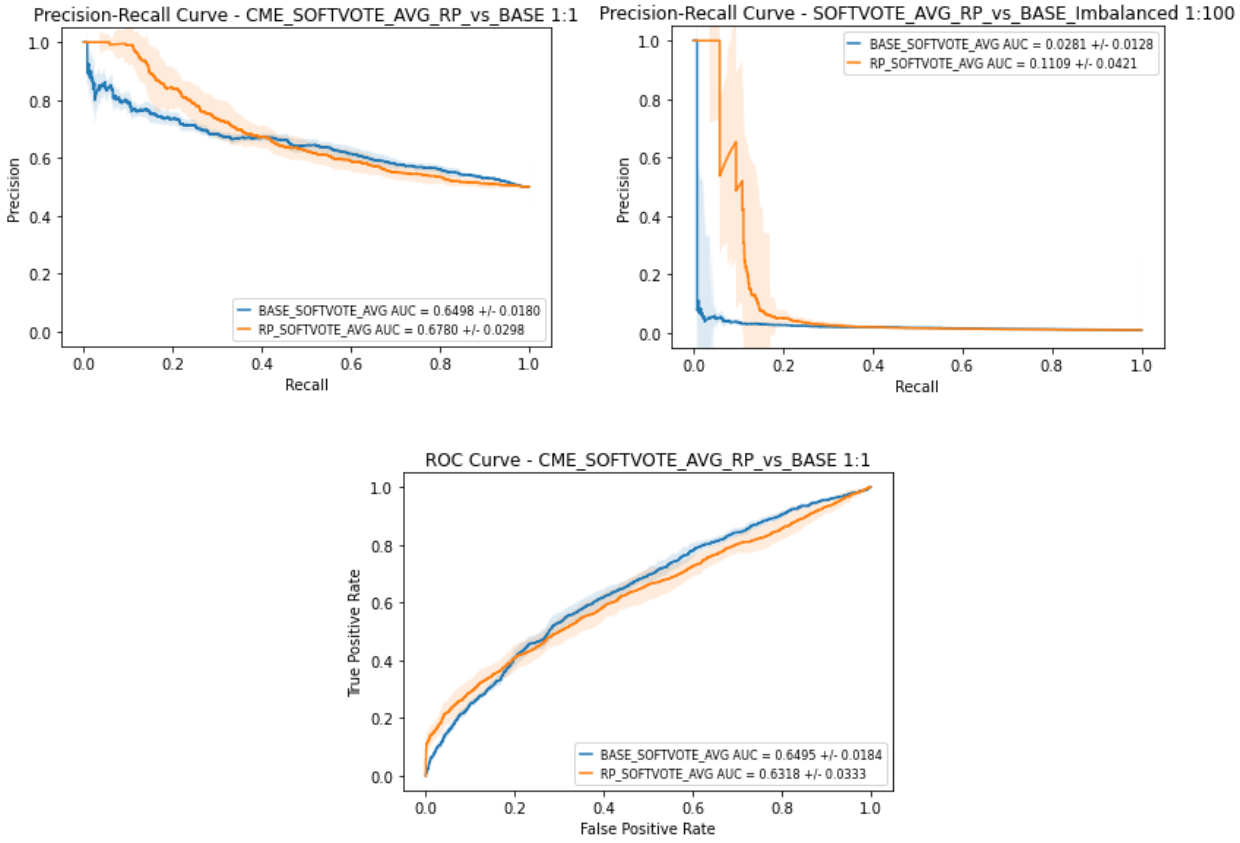

Figure 69: Comparison of soft-vote predictions by combining base classifiers vs. combining RP-enhanced classifiers.

A comparison of the highest auPR results between meta-classifiers, "LightGBM soft vote averaging" and "RP-stacked SVC", showed no significant differences. However, the SVC used in CME has been selected as the highest-performing PPI classifier for *E. coli* and is henceforth referred to as RP-CME. This CME approach further improved performance as seen in Figure 70 over single RP-enhanced classifiers. Statistical differences are shown in Table 13 with significant differences shown in bold.

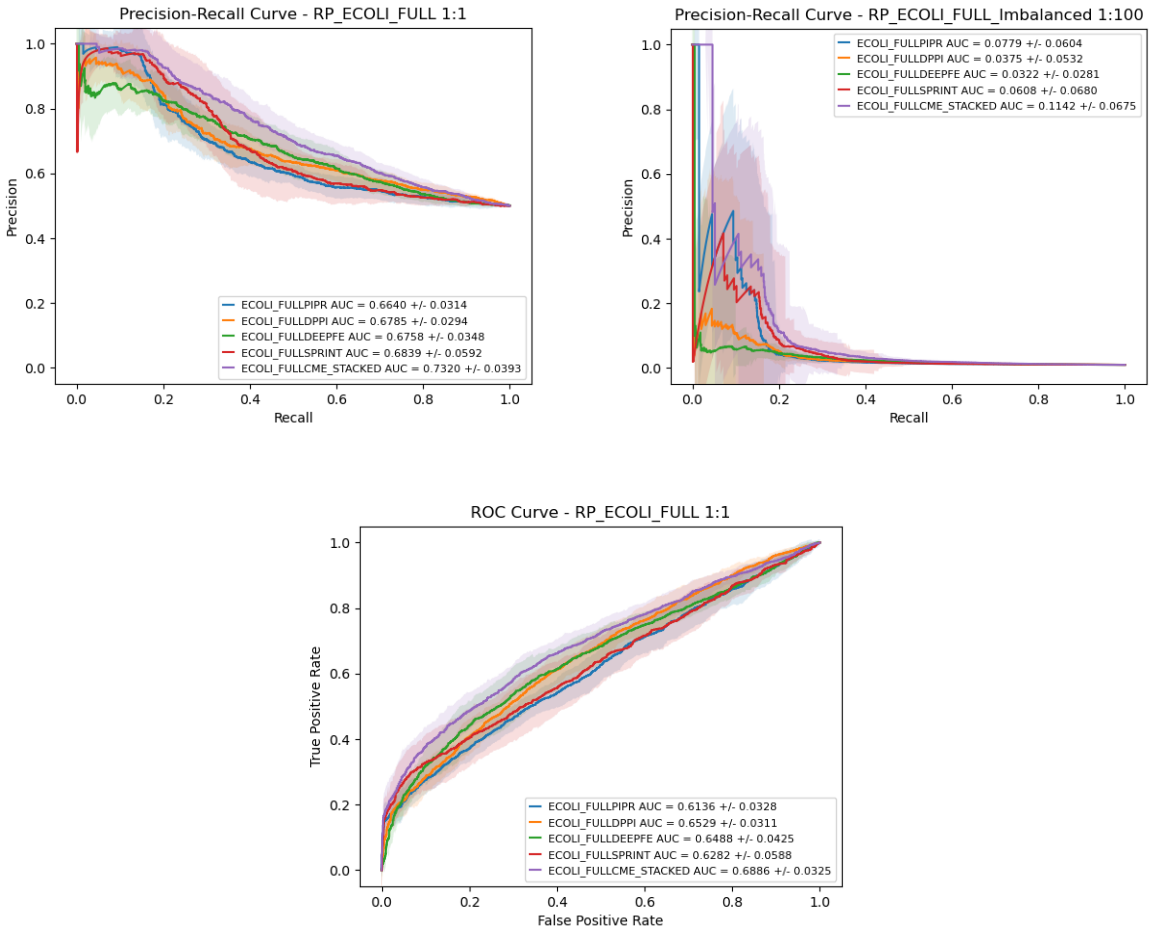

Figure 70: Enhancement of RP models using a RP-stacked CME scheme.

Table 13: T-test statistics for RP-CME comparison to RP-enhanced base classifiers.

BALANCED	RP-SPRINT	RP-DPPI	RP-DEEPFE	RP-PIPR
RP-CME	F = 2.926 p = 0.0169	F = 5.059 **p = 0.0007**	F = 6.616 **p = 0.0001**	F = 5.753 **p = 0.0003**
Imbalanced 1:100	RP-SPRINT	RP-DPPI	RP-DEEPFE	RP-PIPR
RP-CME	F = 2.172 p = 0.0579	F = 3.643 **p = 0.0054**	F = 5.306 **p = 0.0005**	F = 1.604 p = 0.1431

An investigation of each RP-enhanced classifier contribution to the CME performance improvement was also conducted. This was done by stacking RP features of each model but leaving one model's feature set out. Stacking in this way provides a diagnostic as to which base classifiers produced the best RP features. These results are shown in Figure 71 with statistics in the following Table 14.

An ANOVA test suggested no significant difference among each stacked model. However, t-tests shown below indicate possible strength of contributions from each classifier's RP features to the stacked RP-CME model. Noticeably, SPRINT and DEEPFE provide the most useful RP information for the stacked model to improve predictions, with SPRINT significantly contributing the most for imbalanced evaluations. Although, an additional test stacking only SPRINT and DEEPFE RP features did not produce as high performance as the stacked model using all four RP sets suggesting that for stacked CME schemes, more base classifiers may be better.

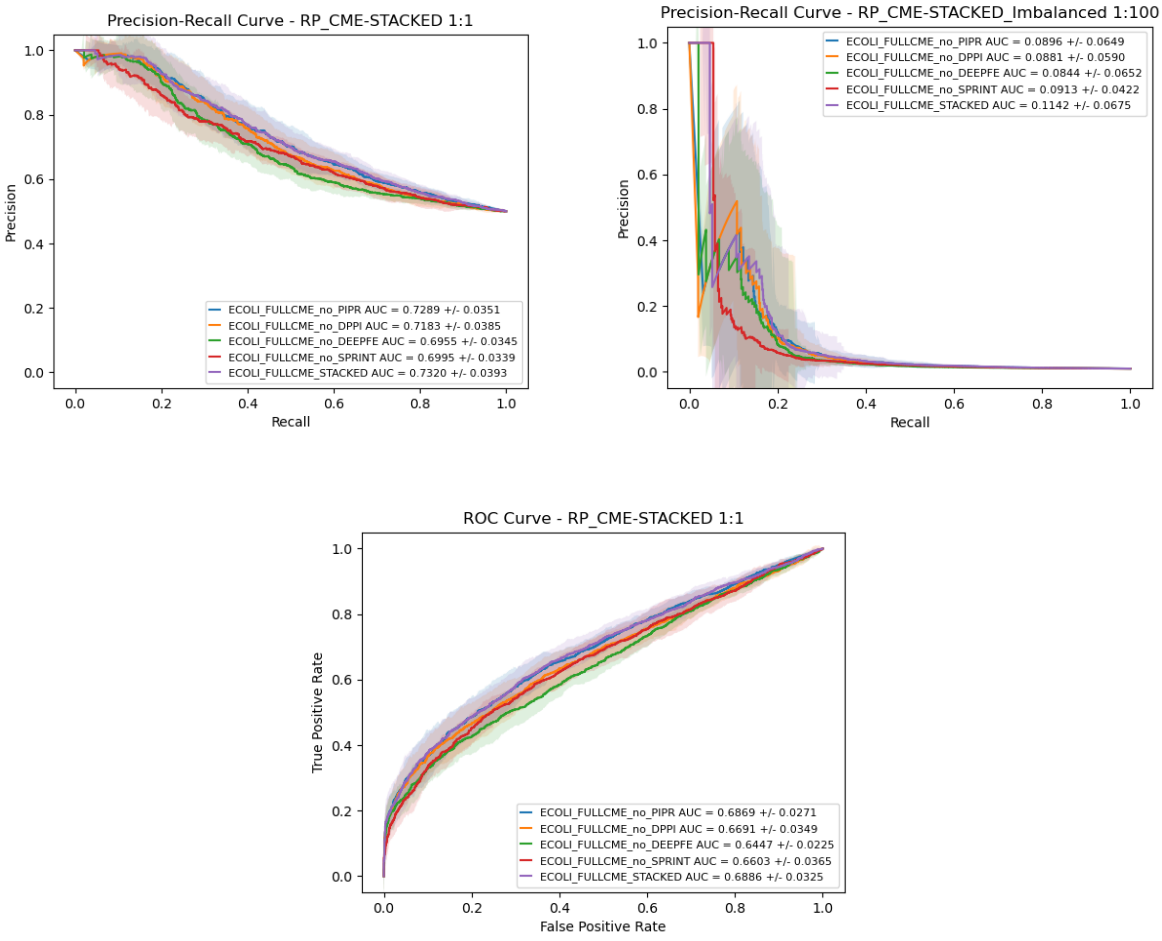

Figure 71: Contribution of RP enhanced models to RP-stacked CME performance.

Table 14: T-test statistics for contributions of classifier RP features to stacked CME model.

BALANCED	RP-CME (no SPRINT)	RP-CME (no DPPI)	RP-CME (no DEEPFE)	RP-CME (no PIPR)
RP-CME	F = 3.515 **p = 0.0066**	F = 1.808 p = 0.1041	F = 4.660 **p = 0.0012**	F = -0.238 p = 0.8169
Imbalanced 1:100	RP-CME (no SPRINT)	RP-CME (no DPPI)	RP-CME (no DEEPFE)	RP-CME (no PIPR)
RP-CME	F = 4.489 **p = 0.0015**	F = 0.135 p = 0.8955	F = 0.645 p = 0.5351	F = 0.892 p = 0.3956

Finally, an investigation of the effect of the RP feature set on performance was conducted. This was done to further demonstrate the advantage of defining additional RP features described in Table 11 to supplement the original RP features shown in Table 3. The final stacked RP-CME model was trained and tested using the original RP feature set and the appended RP feature set on the ECOLI_FULL dataset. Performance curves from a 10-CV is shown in Figure 72. These results indicate that an RP approach can benefit from additional features.

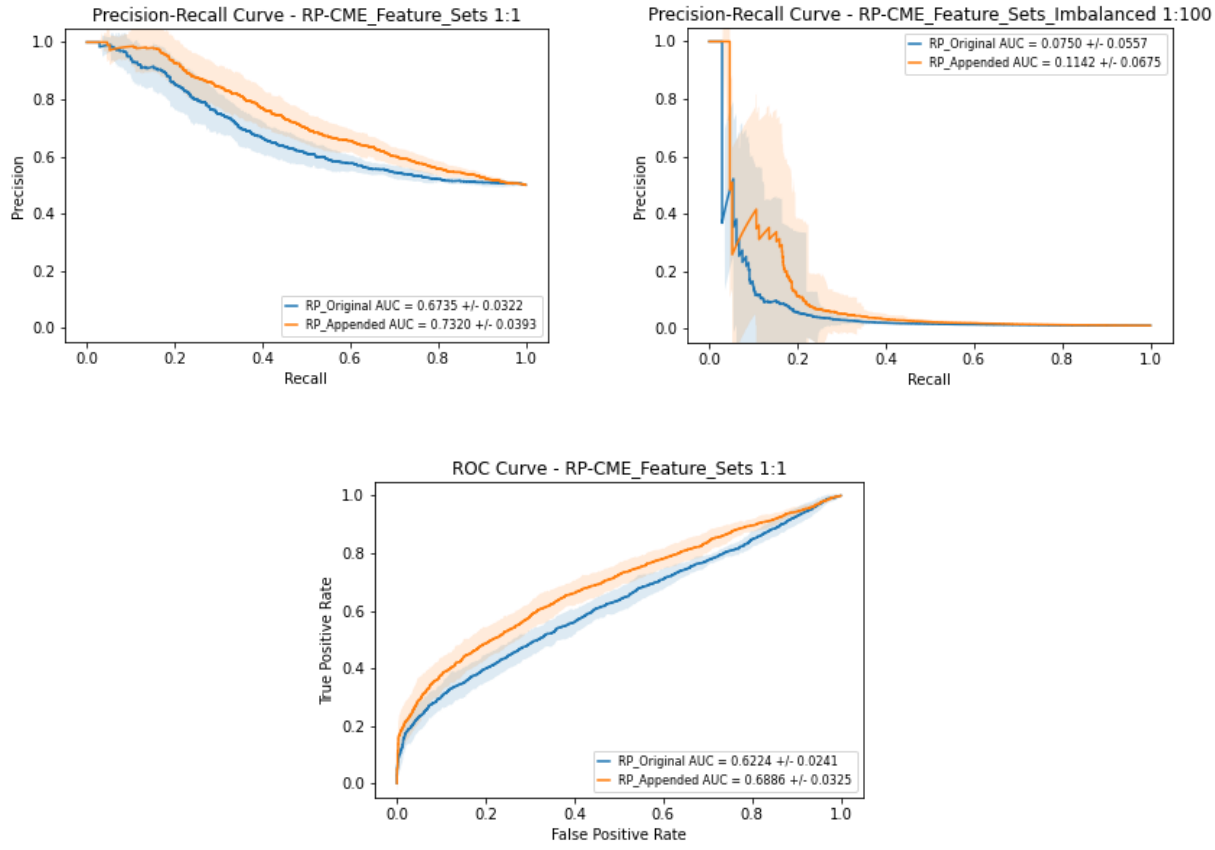

Figure 72: RP-CME performance using the original RP feature set vs. the appended RP feature set.

4.2.3 Discussion and Concluding Remarks

Applying an RP feature approach as an ensemble CME method has been shown to improve PPI classification performance over base classifiers and over single RP enhanced classifiers. Only four base classifiers have been shown here, but it is presumed that any model used for binary classification tasks can benefit from this method since these RP features are insensitive to the type of scoring in a model's prediction values. Additional stacking of RP features from base classifiers may further enhance prediction performance.

In terms of producing confident predictions, RP features from SPRINT had the greatest effect on increasing auPR of the stacked CME model for class imbalanced evaluations. Due its contribution to the CME model and the speed and simplicity of running SPRINT, it is recommended to include this base model in future RP-CME stacking methods.

Despite this CME method being shown to improve PPI classification for bacteria, limitations remain. Implementing this multi-step process requires multi-core computing systems to run SPRINT and extract RP features within a reasonable timeframe. The other base classifiers also require specific GPUs to train their neural network models; however, this could be overcome by reimplementing their NN designs with new API. Consequently, a lack of access to heterogeneous high-performance computing platforms is a limitation of implementing this stacked CME method using RP features.

Besides researching improvements to RP mentioned in Section 4.1.3, other approaches can be explored to improve PPI prediction in bacteria. The results presented in this thesis suggest that dataset size is one factor limiting accurate PPI prediction in

bacteria. Thus, greater effort by wet lab researchers is required to provide prediction models with more training data. Also, dataset size may only be one factor for weak PPI prediction observed in *E. coli*. For instance, there can be underlying differences in bacterial protein sequences compared to other species that can be examined further to produce a species-specific model that captures this biological context. In this direction, collaboration with molecular biologists that specialize in bacteria is being pursued to identify biologically motivated strategies for bacteria-specific PPI prediction.

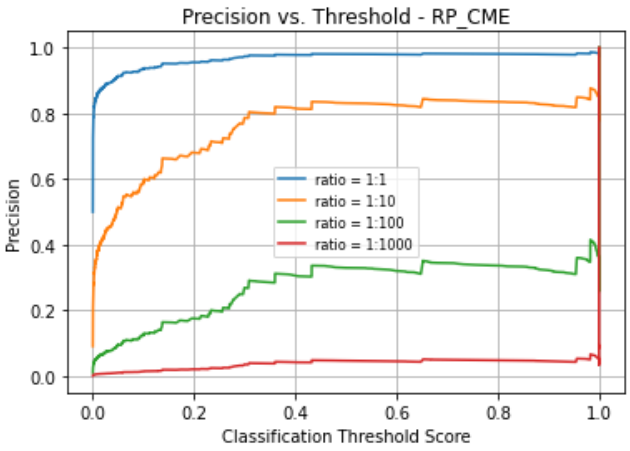

Figure 73: Precision vs. classification threshold using different class imbalance ratios for RP-CME.

The final RP-CME model can produce predictions which can be feasible for experimental confirmation. As illustrated in Figure 73, high scoring predictions are likely to be true almost 100%, 80%, 35%, and 5% of the time when considering class imbalances of 1 to 1, 10, 100, and 1000, respectively. These precision-threshold curves are similar to those for RP-SPRINT in Figure 63 which is also consistent with their respective PR performance curves. Overall, RP-CME produces the best predictions, but if computational resources and time are limited then RP-SPRINT alone may produce adequately similar predictions.

5 Thesis Summary and Future Recommendations

5.1 Final Conclusions

This thesis has presented a way to improve computational PPI prediction for bacteria using *E. coli* as an example and applied to sequence-based prediction models. The current PPI knowledgebase for *E. coli* is limited relative to other model organisms such as human and yeast. More data are required to further improve PPI prediction while high-quality data are required to produce confident predictions. Stacking an extended set of RP features as input to a meta-classifier significantly enhances prediction performance over four state-of-the-art base classifiers which can be seen in Appendix C.

Improvements to PPI prediction for bacteria have been demonstrated in this thesis. The methods presented here for improvement are influenced by the quality of base classifier predictions which rely on the datasets used for training. Overall, the following can be stated:

1. There is a lack of available PPI data for bacteria compared to yeast and human.
2. PPI prediction tends to improve with more data.
3. Benchmarks should be periodically updated to reflect the most accurate and comprehensive PPI data available, and models should be re-trained as new data are released.
4. State-of-the-art predictors perform poorly for bacterial PPI, relative to other species.
5. State-of-the-art predictors can be improved using RP-enhancement.

6. Stacking RP features from multiple methods further improves classification performance.
7. Evaluation of classifiers should be more comprehensive and consistent.

5.2 Summary of Contributions

Throughout the investigation of this thesis, several contributions have been made. Firstly, problems have been presented in data collection and benchmark datasets reported in the literature that have been ubiquitously used in PPI predictor evaluations. A solution to this problem is provided with an algorithm for constructing high-quality datasets using current protein and PPI information. Secondly, publications in binary PPI classification have often not provided a consistent and comprehensive evaluation on bacteria which limits one's ability to accurately interpret and compare methods. This thesis establishes a systematic, impartial, and comprehensive evaluation of such classifiers and demonstrates that these models do not perform well at predicting PPIs for *E. coli* or for new PPI data in general. To add, an extension of RP features has been shown to provide useful information for applying RP. Lastly, using a combination of RP features extracted from multiple classifiers has been shown to significantly enhance PPI prediction for *E. coli*.

In the effort to identify a computational method for producing reliable bacterial PPI predictions, the stacked RP-CME model developed here currently outperforms state-of-the-art predictors. This model can be used to assist PPI discovery for anti-bacterial drugs and microbiome research. Experimental validation of the highest scoring PPIs produced by this model is currently being pursued through collaboration with Dr. Alex Wong (Carleton University).

5.3 Recommendations for future work

The research presented in this thesis provides insight into potential future work to be explored. Training models in different ways can be explored to improve performance. For example, training with an imbalanced dataset and weighting classes may be suitable to overcome the issue of a lack of NIP data while providing representative examples of positive and negative protein pairs. Changing data used for pre-trained embeddings may also contribute to improving prediction performance. Additionally, overcoming the small dataset size for *E. coli*-trained models could be explored using transfer learning by taking, for example, and human-trained classifier then fine-tuning the model with *E. coli* data. Finally, a more exhaustive exploration of hyperparameter tuning can be conducted to seek performance improvements.

Additionally, improving RP feature design and an in-depth analysis can be performed for optimal feature selection. Differences in datasets between species can also be analyzed further to determine causes for performance differences. Furthermore, development of a novel base classifier can be investigated to produce better predictions for *E. coli* prior to enhancement with RP. Future studies can also apply this RP-CME method to other bacterial datasets and determine usefulness for cross-species predictions.

References

[1] Y. Fan and O. Pedersen, “Gut microbiota in human metabolic health and disease,” *Nature Reviews Microbiology*, vol. 19, no. 1. 2021. doi: 10.1038/s41579-020-0433-9.

[2] M. Schultz, “Clinical use of E. coli Nissle 1917 in inflammatory bowel disease,” *Inflammatory Bowel Diseases*, vol. 14, no. 7. 2008. doi: 10.1002/ibd.20377.

[3] Z. Zhou *et al.*, “Engineering probiotics as living diagnostics and therapeutics for improving human health,” *Microbial Cell Factories*, vol. 19, no. 1. 2020. doi: 10.1186/s12934-020-01318-z.

[4] L. Carro, “Protein-protein interactions in bacteria: A promising and challenging avenue towards the discovery of new antibiotics,” *Beilstein Journal of Organic Chemistry*, vol. 14. 2018. doi: 10.3762/bjoc.14.267.

[5] *Model Organisms for Microbial Pathogenesis, Biofilm Formation and Antimicrobial Drug Discovery*. 2020. doi: 10.1007/978-981-15-1695-5.

[6] J. Vila *et al.*, “Escherichia coli: An old friend with new tidings,” *FEMS Microbiology Reviews*, vol. 40, no. 4. Oxford University Press, pp. 437–463, Jul. 01, 2016. doi: 10.1093/femsre/fuw005.

[7] D. K. Maheshwari, *Bacteria in agrobiology: Disease management*. 2012. doi: 10.1007/978-3-642-33639-3.

[8] G. Walsh, *Proteins: Biochemistry and Biotechnology: Second Edition*. Wiley Blackwell, 2015. doi: 10.1002/9781119117599.

[9] K. M. Poluri, K. Gulati, and S. Sarkar, “Structural and Functional Properties of Proteins,” in *Protein-Protein Interactions*, Singapore: Springer Singapore, 2021, pp. 1–60. doi: 10.1007/978-981-16-1594-8_1.

[10] X. Peng, J. Wang, W. Peng, F. X. Wu, and Y. Pan, “Protein-protein interactions: detection, reliability assessment and applications,” *Briefings in bioinformatics*, vol. 18, no. 5. 2017. doi: 10.1093/bib/bbw066.

[11] A. H. Smits and M. Vermeulen, “Characterizing Protein–Protein Interactions Using Mass Spectrometry: Challenges and Opportunities,” *Trends in Biotechnology*, vol. 34, no. 10. 2016. doi: 10.1016/j.tibtech.2016.02.014.

[12] K. M. Poluri, K. Gulati, and S. Sarkar, “Prediction, Analysis, Visualization, and Storage of Protein–Protein Interactions Using Computational Approaches,” in *Protein-Protein Interactions*, Singapore: Springer Singapore, 2021, pp. 265–346. doi: 10.1007/978-981-16-1594-8_6.

[13] K. Dick and J. R. Green, “Reciprocal Perspective for Improved Protein-Protein Interaction Prediction,” *Scientific Reports*, vol. 8, no. 1, 2018, doi: 10.1038/s41598-018-30044-1.

[14] A. Bateman *et al.*, “UniProt: The universal protein knowledgebase in 2021,” *Nucleic Acids Research*, vol. 49, no. D1, 2021, doi: 10.1093/nar/gkaa1100.

[15] F. R. Blattner *et al.*, “The complete genome sequence of Escherichia coli K-12,” *Science*, vol. 277, no. 5331. 1997. doi: 10.1126/science.277.5331.1453.

[16] C. Kanz *et al.*, “The EMBL nucleotide sequence database,” *Nucleic Acids Research*, vol. 33, no. DATABASE ISS., 2005, doi: 10.1093/nar/gki098.

[17] I. M. Keseler *et al.*, “The EcoCyc database: Reflecting new knowledge about Escherichia coli K-12,” *Nucleic Acids Research*, vol. 45, no. D1, 2017, doi: 10.1093/nar/gkw1003.

[18] R. v. Misra, R. S. P. Horler, W. Reindl, I. I. Goryanin, and G. H. Thomas, “EchoBASE: An integrated post-genomic database for Escherichia coli,” *Nucleic Acids Research*, vol. 33, no. DATABASE ISS., 2005, doi: 10.1093/nar/gki028.

[19] H. Yun *et al.*, “EcoProDB: The Escherichia coli protein database,” *Bioinformatics*, vol. 23, no. 18, 2007, doi: 10.1093/bioinformatics/btm351.

[20] A. R. Wattam *et al.*, “PATRIC, the bacterial bioinformatics database and analysis resource,” *Nucleic Acids Research*, vol. 42, no. D1, 2014, doi: 10.1093/nar/gkt1099.

[21] J. C. Silva *et al.*, “Simultaneous qualitative and quantitative analysis of the Escherichia coli proteome: A sweet tale,” *Molecular and Cellular Proteomics*, vol. 5, no. 4, 2006, doi: 10.1074/mcp.M500321-MCP200.

[22] A. Schmidt *et al.*, “The quantitative and condition-dependent Escherichia coli proteome,” *Nature Biotechnology*, vol. 34, no. 1, 2016, doi: 10.1038/nbt.3418.

[23] A. Mateus *et al.*, “The functional proteome landscape of Escherichia coli,” *Nature*, vol. 588, no. 7838, 2020, doi: 10.1038/s41586-020-3002-5.

[24] P. Hu *et al.*, “Global functional atlas of Escherichia coli encompassing previously uncharacterized proteins,” *PLoS Biology*, vol. 7, no. 4, 2009, doi: 10.1371/journal.pbio.1000096.

[25] M. Shatsky *et al.*, “Bacterial interactomes: Interacting protein partners share similar function and are validated in independent assays more frequently than previously

reported," *Molecular and Cellular Proteomics*, vol. 15, no. 5, 2016, doi: 10.1074/mcp.M115.054692.

[26] R. Velasco-García and R. Vargas-Martínez, "The study of protein-protein interactions in bacteria," *Canadian Journal of Microbiology*, vol. 58, no. 11, 2012, doi: 10.1139/w2012-104.

[27] K. M. Poluri, K. Gulati, and S. Sarkar, "Experimental Methods for Determination of Protein–Protein Interactions," in *Protein-Protein Interactions*, Singapore: Springer Singapore, 2021, pp. 197–264. doi: 10.1007/978-981-16-1594-8_5.

[28] L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie, and D. Eisenberg, "The Database of Interacting Proteins: 2004 update," *Nucleic Acids Research*, vol. 32, no. DATABASE ISS., 2004, doi: 10.1093/nar/gkh086.

[29] R. Oughtred *et al.*, "The BioGRID interaction database: 2019 update," *Nucleic Acids Research*, vol. 47, no. D1, 2019, doi: 10.1093/nar/gky1079.

[30] L. Licata *et al.*, "MINT, the molecular interaction database: 2012 Update," *Nucleic Acids Research*, vol. 40, no. D1, 2012, doi: 10.1093/nar/gkr930.

[31] S. Orchard *et al.*, "The MIntAct project - IntAct as a common curation platform for 11 molecular interaction databases," *Nucleic Acids Research*, vol. 42, no. D1, 2014, doi: 10.1093/nar/gkt1115.

[32] T. S. Keshava Prasad *et al.*, "Human Protein Reference Database - 2009 update," *Nucleic Acids Research*, vol. 37, no. SUPPL. 1, 2009, doi: 10.1093/nar/gkn892.

[33] D. Szklarczyk *et al.*, "STRING v11: Protein-protein association networks with increased coverage, supporting functional discovery in genome-wide experimental

datasets," *Nucleic Acids Research*, vol. 47, no. D1, 2019, doi: 10.1093/nar/gky1131.

[34] B. Lehne and T. Schlitt, "Protein-protein interaction databases: keeping up with growing interactomes.," *Hum Genomics*, vol. 3, no. 3, 2009, doi: 10.1186/1479-7364-3-3-291.

[35] V. S. Rao, K. Srinivas, G. N. Sujini, and G. N. S. Kumar, "Protein-Protein Interaction Detection: Methods and Analysis," *International Journal of Proteomics*, vol. 2014, 2014, doi: 10.1155/2014/147648.

[36] B. Turner *et al.*, "iRefWeb: interactive analysis of consolidated protein interaction data and their supporting evidence.," *Database (Oxford)*, vol. 2010, 2010, doi: 10.1093/database/baq023.

[37] C. Prieto and J. de Las Rivas, "APID: Agile protein interaction DataAnalyzer," *Nucleic Acids Research*, vol. 34, no. WEB. SERV. ISS., 2006, doi: 10.1093/nar/gkl128.

[38] A. Patil, K. Nakai, and H. Nakamura, "HitPredict: A database of quality assessed protein-protein interactions in nine species," *Nucleic Acids Research*, vol. 39, no. SUPPL. 1, 2011, doi: 10.1093/nar/gkq897.

[39] E. A. Leon, L. Ezkurdia, B. García, A. Valencia, and D. Juan, "EcID. A database for the inference of functional interactions in E. coli," *Nucleic Acids Research*, vol. 37, no. SUPPL. 1, 2009, doi: 10.1093/nar/gkn853.

[40] C. Su *et al.*, "Bacteriome.org - An integrated protein interaction database for E. coli," *Nucleic Acids Research*, vol. 36, no. SUPPL. 1, 2008, doi: 10.1093/nar/gkm807.

[41] J. Goll, S. v. Rajagopala, S. C. Shiau, H. Wu, B. T. Lamb, and P. Uetz, “MPIDB: The microbial protein interaction database,” *Bioinformatics*, vol. 24, no. 15, 2008, doi: 10.1093/bioinformatics/btn285.

[42] S. v. Rajagopala *et al.*, “MPI-LIT: A literature-curated dataset of microbial binary protein-protein interactions,” *Bioinformatics*, vol. 24, no. 22, 2008, doi: 10.1093/bioinformatics/btn481.

[43] N. Singh, V. Bhatia, S. Singh, and S. Bhatnagar, “MorCVD: A Unified Database for Host-Pathogen Protein-Protein Interactions of Cardiovascular Diseases Related to Microbes,” *Scientific Reports*, vol. 9, no. 1, 2019, doi: 10.1038/s41598-019-40704-5.

[44] R. Kumar and B. Nanduri, “HPIDB - a unified resource for host-pathogen interactions,” *BMC Bioinformatics*, vol. 11, no. SUPPL. 6, 2010, doi: 10.1186/1471-2105-11-16.

[45] M. G. Ammari, C. R. Gresham, F. M. McCarthy, and B. Nanduri, “HPIDB 2.0: a curated database for host-pathogen interactions,” *Database (Oxford)*, vol. 2016, 2016, doi: 10.1093/database/baw103.

[46] Y. Z. Zhou, Y. Gao, and Y. Y. Zheng, “Prediction of protein-protein interactions using local description of amino acid sequence,” in *Communications in Computer and Information Science*, 2011, vol. 202 CCIS, no. PART 2. doi: 10.1007/978-3-642-22456-0_37.

[47] K. Dick, F. Dehne, A. Golshani, and J. R. Green, “Positome: A method for improving protein-protein interaction quality and prediction accuracy,” 2017. doi: 10.1109/CIBCB.2017.8058545.

[48] S. Pitre *et al.*, “PIPE: A protein-protein interaction prediction engine based on the re-occurring short polypeptide sequences between known interacting protein pairs,” *BMC Bioinformatics*, vol. 7, 2006, doi: 10.1186/1471-2105-7-365.

[49] M. Babu *et al.*, “Global landscape of cell envelope protein complexes in Escherichia coli,” *Nature Biotechnology*, vol. 36, no. 1, 2018, doi: 10.1038/nbt.4024.

[50] S. v. Rajagopala *et al.*, “The binary protein-protein interaction landscape of escherichia coli,” *Nature Biotechnology*, vol. 32, no. 3, 2014, doi: 10.1038/nbt.2831.

[51] W. Li and A. Godzik, “Cd-hit: A fast program for clustering and comparing large sets of protein or nucleotide sequences,” *Bioinformatics*, vol. 22, no. 13, 2006, doi: 10.1093/bioinformatics/btl158.

[52] P. Smialowski *et al.*, “The Negatome database: A reference set of non-interacting protein pairs,” *Nucleic Acids Research*, vol. 38, no. SUPPL.1, 2009, doi: 10.1093/nar/gkp1026.

[53] P. Blohm *et al.*, “Negatome 2.0: A database of non-interacting proteins derived by literature mining, manual annotation and protein structure analysis,” *Nucleic Acids Research*, vol. 42, no. D1, 2014, doi: 10.1093/nar/gkt1079.

[54] A. Moscatelli, “Modeling of negative protein-protein interactions: methods and experiments,” Oct. 2019.

[55] J. Yu, M. Guo, C. J. Needham, Y. Huang, L. Cai, and D. R. Westhead, “Simple sequence-based kernels do not predict protein-protein interactions,” *Bioinformatics*, vol. 26, no. 20, 2010, doi: 10.1093/bioinformatics/btq483.

[56] Y. Park and E. M. Marcotte, “Revisiting the negative example sampling problem for predicting protein-protein interactions,” *Bioinformatics*, vol. 27, no. 21, 2011, doi: 10.1093/bioinformatics/btr514.

[57] A. Ben-Hur and W. S. Noble, “Choosing negative examples for the prediction of protein-protein interactions,” *BMC Bioinformatics*, vol. 7, no. SUPPL.1, 2006, doi: 10.1186/1471-2105-7-S1-S2.

[58] L. Zhang, G. Yu, M. Guo, and J. Wang, “Predicting protein-protein interactions using high-quality non-interacting pairs,” *BMC Bioinformatics*, vol. 19, 2018, doi: 10.1186/s12859-018-2525-3.

[59] S. Mei and K. Zhang, “Neglog: Homology-based negative data sampling method for genome-scale reconstruction of human protein–protein interaction networks,” *International Journal of Molecular Sciences*, vol. 20, no. 20, 2019, doi: 10.3390/ijms20205075.

[60] O. Keskin, N. Tuncbag, and A. Gursoy, “Predicting Protein-Protein Interactions from the Molecular to the Proteome Level,” *Chemical Reviews*, vol. 116, no. 8. 2016. doi: 10.1021/acs.chemrev.5b00683.

[61] D. Sarkar and S. Saha, “Machine-learning techniques for the prediction of protein–protein interactions,” *Journal of Biosciences*, vol. 44, no. 4. 2019. doi: 10.1007/s12038-019-9909-z.

[62] L. Skrabanek, H. K. Saini, G. D. Bader, and A. J. Enright, “Computational Prediction of Protein–Protein Interactions,” *Molecular Biotechnology*, vol. 38, no. 1, pp. 1–17, Jan. 2008, doi: 10.1007/s12033-007-0069-2.

[63] J. Jumper *et al.*, “Highly accurate protein structure prediction with AlphaFold,” *Nature*, vol. 596, no. 7873, 2021, doi: 10.1038/s41586-021-03819-2.

[64] M. Baek *et al.*, “Accurate prediction of protein structures and interactions using a three-track neural network,” *Science (1979)*, vol. 373, no. 6557, 2021, doi: 10.1126/science.abj8754.

[65] Y. Li and L. Ilie, “SPRINT: Ultrafast protein-protein interaction prediction of the entire human interactome,” *BMC Bioinformatics*, vol. 18, no. 1, 2017, doi: 10.1186/s12859-017-1871-x.

[66] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman, “Basic local alignment search tool,” *Journal of Molecular Biology*, vol. 215, no. 3, 1990, doi: 10.1016/S0022-2836(05)80360-2.

[67] S. F. Altschul *et al.*, “Gapped BLAST and PSI-BLAST: A new generation of protein database search programs,” *Nucleic Acids Research*, vol. 25, no. 17. 1997. doi: 10.1093/nar/25.17.3389.

[68] J. R. Bock and D. A. Gough, “Predicting protein-protein interactions from primary structure,” *Bioinformatics*, vol. 17, no. 5, 2001, doi: 10.1093/bioinformatics/17.5.455.

[69] S. Martin, D. Roe, and J. L. Faulon, “Predicting protein-protein interactions using signature products,” *Bioinformatics*, vol. 21, no. 2, 2005, doi: 10.1093/bioinformatics/bth483.

[70] B. K. Sriwastava, S. Basu, and U. Maulik, “Predicting Protein-Protein Interaction Sites with a Novel Membership Based Fuzzy SVM Classifier,” in *IEEE/ACM*

Transactions on Computational Biology and Bioinformatics, 2015, vol. 12, no. 6. doi: 10.1109/TCBB.2015.2401018.

[71] R. K. Barman, T. Jana, S. Das, and S. Saha, “Prediction of intra-species protein-protein interactions in enteropathogens facilitating systems biology study,” *PLoS ONE*, vol. 10, no. 12, 2015, doi: 10.1371/journal.pone.0145648.

[72] S. Hashemifar, B. Neyshabur, A. A. Khan, and J. Xu, “Predicting protein-protein interactions through sequence-based deep learning,” in *Bioinformatics*, 2018, vol. 34, no. 17. doi: 10.1093/bioinformatics/bty573.

[73] Y. Yao, X. Du, Y. Diao, and H. Zhu, “An integration of deep learning with feature embedding for protein–protein interaction prediction,” *PeerJ*, vol. 2019, no. 6, 2019, doi: 10.7717/peerj.7126.

[74] M. Chen *et al.*, “Multifaceted protein-protein interaction prediction based on Siamese residual RCNN,” in *Bioinformatics*, 2019, vol. 35, no. 14. doi: 10.1093/bioinformatics/btz328.

[75] D. G. Kyrollos, B. Reid, K. Dick, and J. R. Green, “RPmirDIP: Reciprocal Perspective improves miRNA targeting prediction,” *Scientific Reports*, vol. 10, no. 1, 2020, doi: 10.1038/s41598-020-68251-4.

[76] K. Dick *et al.*, “PIPE4: Fast PPI Predictor for Comprehensive Inter- and Cross-Species Interactomes,” *Scientific Reports*, vol. 10, no. 1, 2020, doi: 10.1038/s41598-019-56895-w.

[77] Y. Park and E. M. Marcotte, “Flaws in evaluation schemes for pair-input computational predictions,” *Nature Methods*, vol. 9, no. 12. 2012. doi: 10.1038/nmeth.2259.

[78] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” 2013. doi: https://doi.org/10.48550/arXiv.1301.3781.

[79] Y. A. Huang, Z. H. You, X. Gao, L. Wong, and L. Wang, “Using Weighted Sparse Representation Model Combined with Discrete Cosine Transformation to Predict Protein-Protein Interactions from Protein Sequence,” *BioMed Research International*, vol. 2015, 2015, doi: 10.1155/2015/902198.

[80] J. Shen *et al.*, “Predicting protein-protein interactions based only on sequences information,” *Proc Natl Acad Sci U S A*, vol. 104, no. 11, 2007, doi: 10.1073/pnas.0607879104.

[81] G. Moreno-Hagelsieb and K. Latimer, “Choosing BLAST options for better detection of orthologs as reciprocal best hits,” *Bioinformatics*, vol. 24, no. 3, 2008, doi: 10.1093/bioinformatics/btm585.

[82] A. Luque, A. Carrasco, A. Martín, and A. de las Heras, “The impact of class imbalance in classification performance metrics based on the binary confusion matrix,” *Pattern Recognition*, vol. 91, 2019, doi: 10.1016/j.patcog.2019.02.023.

[83] Y. Guo, L. Yu, Z. Wen, and M. Li, “Using support vector machine combined with auto covariance to predict protein-protein interactions from protein sequences,” *Nucleic Acids Research*, vol. 36, no. 9, 2008, doi: 10.1093/nar/gkn159.

[84] V. Satopää, J. Albrecht, D. Irwin, and B. Raghavan, “Finding a ‘kneedle’ in a haystack: Detecting knee points in system behavior,” 2011. doi: 10.1109/ICDCSW.2011.20.

[85] G. Ke *et al.*, “LightGBM: A highly efficient gradient boosting decision tree,” in *Advances in Neural Information Processing Systems*, 2017, vol. 2017-December.

Appendix A: Computational Resources

All base classifiers were run on the Compute Canada Advanced Research Computing platform's High Performance Computing systems. Table 15 describes resources used to run models and wall-clock time required to run them. Note that these values can be used as a general guide of estimates for running 10-fold cross validations and that efficiencies can be made to the programs to run models. Also, DPPI required P100 Pascal NVIDIA GPUs and PIPR used T4 Turing NVIDIA GPUs to run.

Table 15: Resources used for running base classifiers.

Model	Dataset	Memory	Time
SPRINT (computing HSPs, predicting interactions)	ECOLI_FULL	12 cores 272 MB, 15 cores <1MB	4.5 mins, 27 seconds
	HUMAN	10 cores 3.87 GB, 16 cores 741 MB	6 hours 22mins, 45 mins
	YEAST	32 cores 901 MB, 16 cores 126 MB	17 mins, 9 minutes
DPPI	ECOLI_FULL	1.6 GB	56 minutes
	HUMAN	8.54 GB	18 hours
	YEAST	3.81 GB	14 hours
DEEPFE	ECOLI_FULL	3.56 GB	4 hours 15 mins
	HUMAN	45.56 GB	77 hours 30 mins
	YEAST	18.33 GB	32 hours 17mins
PIPR	ECOLI_FULL	4.19 GB	1 hour 36 mins
	HUMAN	39.56 GB	30 hours 30 mins
	YEAST	16.4 GB	12 hours 52 mins

Extracting RP features for the 10 train/test subsets in the ECOLI_FULL dataset used about 36 cores, 4 GB memory, and completed in 15 minutes. Extracting RP features for the complete *E. coli* interactome used 36 cores, 23.34 GB, and 36 minutes. Running the meta-classier was performed on a laptop with an Intel i7-8550U processor, 12GB memory, and took about 40 seconds to complete a 10-fold cross-validation.

Appendix B: Additional Details

A.1 Details of Calculations for Evaluations

The average PR and ROC curves from cross-validations were used to produce variance statistics for area under curves. However, the overall PR and ROC curve in each performance evaluation was calculated by combining each hold-out test set from cross-validations to obtain the curve and area under curve of the complete dataset.

A.2 Parameter spaces explored in classifier grid searches

Hyperparameter tuning was explored for base classifiers using grid searches and cross-validations due to differences in data sizes from the original models. For example, DPPI, DEEPFE, and PIPR were run using learning rates of 0.0001, 0.001, 0.01, 0.05 and epochs of 30, 50, 100, 150, 200, 300, and 400. Notably, the best performance was found using the default suggested parameters for each model. Figure 74 shows examples of learning curves by PIPR and DEEPFE models using ECOLI_FULL which may reflect their poor performance.

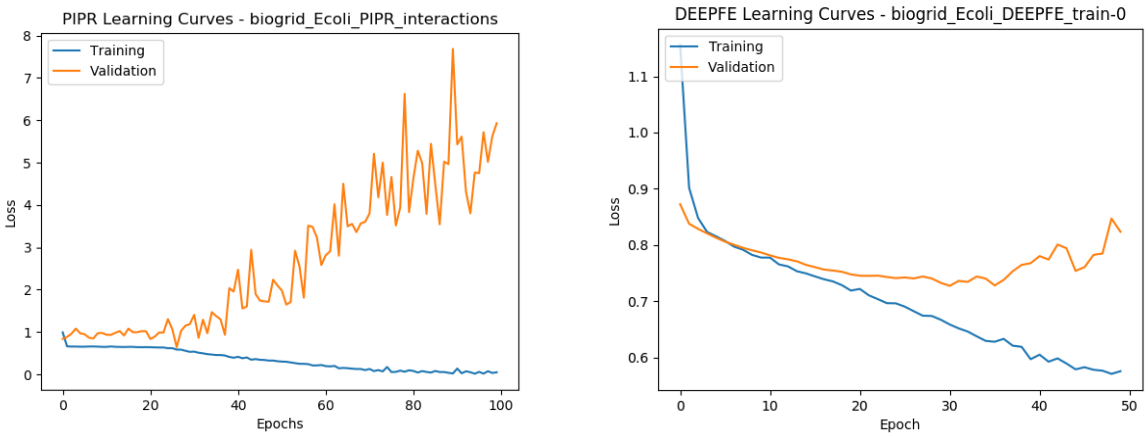

Figure 74: Learning curves for PIPR and DEEPFE using ECOLI_FULL.

A grid search was performed for the LightGBM and SVC meta-classifiers used in RP-enhanced models. The parameter space explored for these can be seen in Table 16 and Table 17.

Table 16: Parameters explored for LightGBM RP-enhanced meta-classifier.

Parameter	Values
boosting_type	'dart', 'gbdt', '**goss**'
learning_rate	0.05, **0.1**, 0.15, 0.2
num_leaves	5, 20, **50**, 100
n_estimators	10, 40, 100, **150**, 200
min_data_in_leaf	10, **50**, 100
max_depth	3, 5, **10**, -1
smoothing	0, **0.1**, 0.2
lambda_l1	**0**, 0.05, 0.1
lambda_l2	**0**, 0.05, 0.1
min_gain_to_split	**0**, 0.1

Table 17: Parameters explored for the SVC stacked meta-classifier.

Parameter	Values
C	0.1, 0.2, 0.5, **0.6**, 0.7, 1, 1.5
kernel	'rbf', 'linear', '**sigmoid**'
gamma	'**scale**', 'auto'

Appendix C: Additional Results

C.1 Performance comparisons for level of proteome resolution for RP

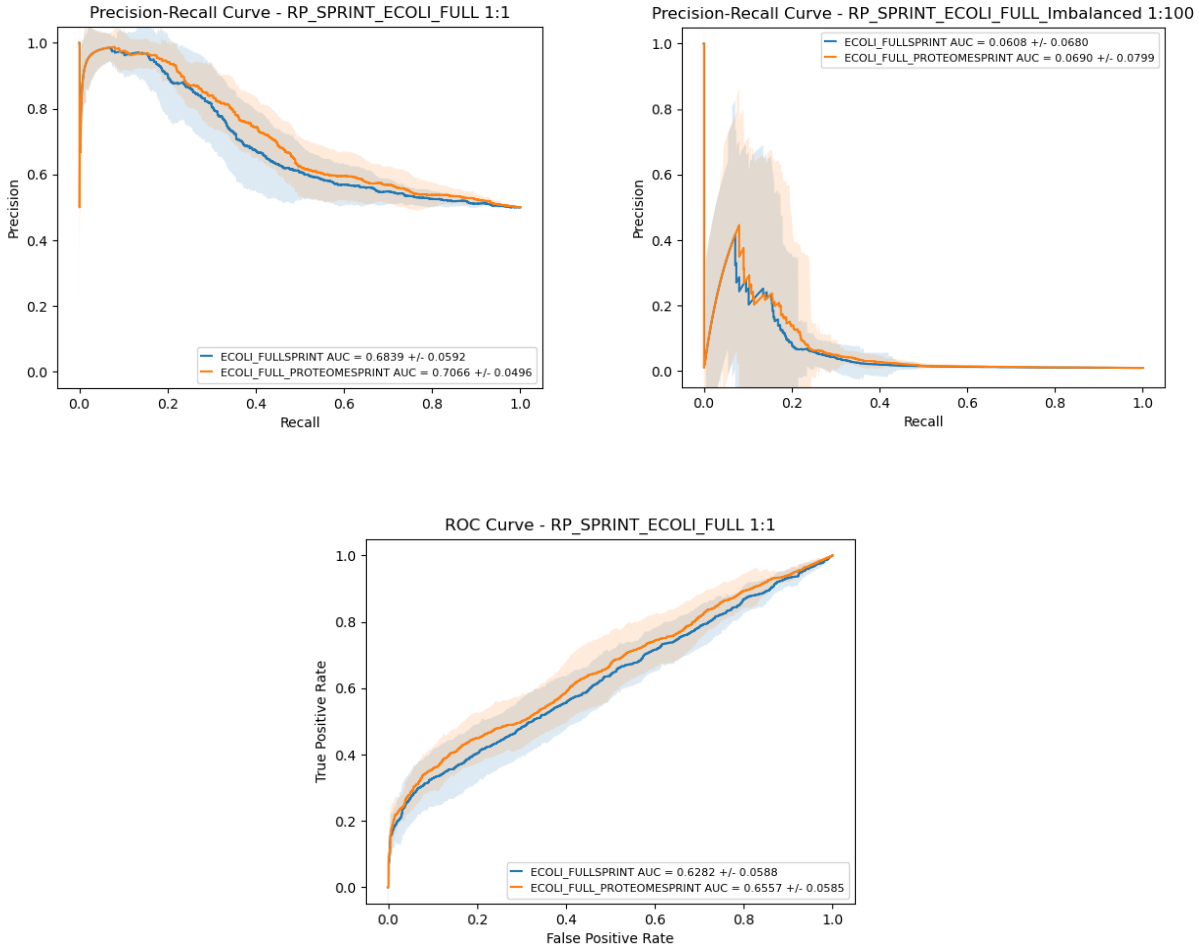

Figure 75: Comparison of RP-SPRINT performance using features extracted by from ECOLI_FULL proteins versus the entire proteome.

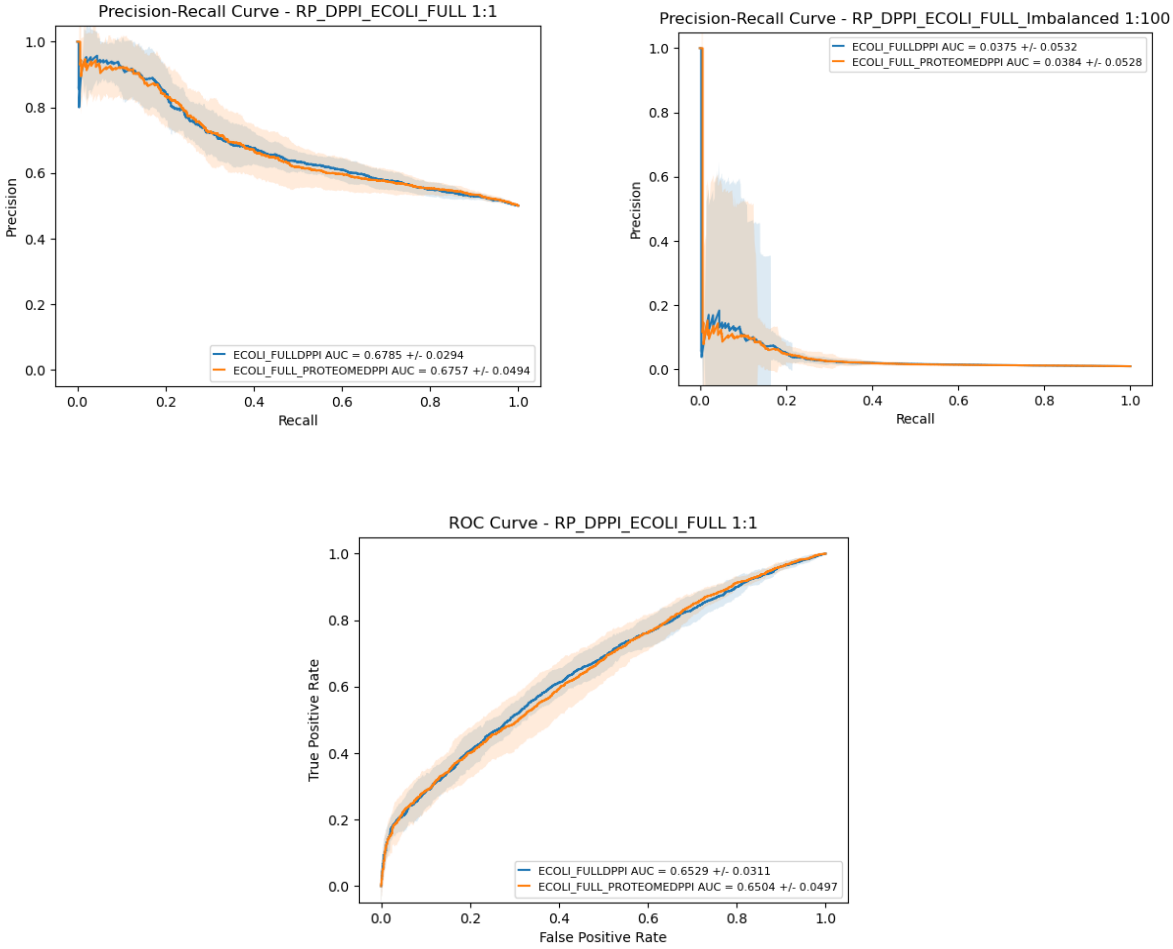

Figure 76: Comparison of RP-DPPI performance using features extracted by from ECOLI_FULL proteins versus the entire proteome.

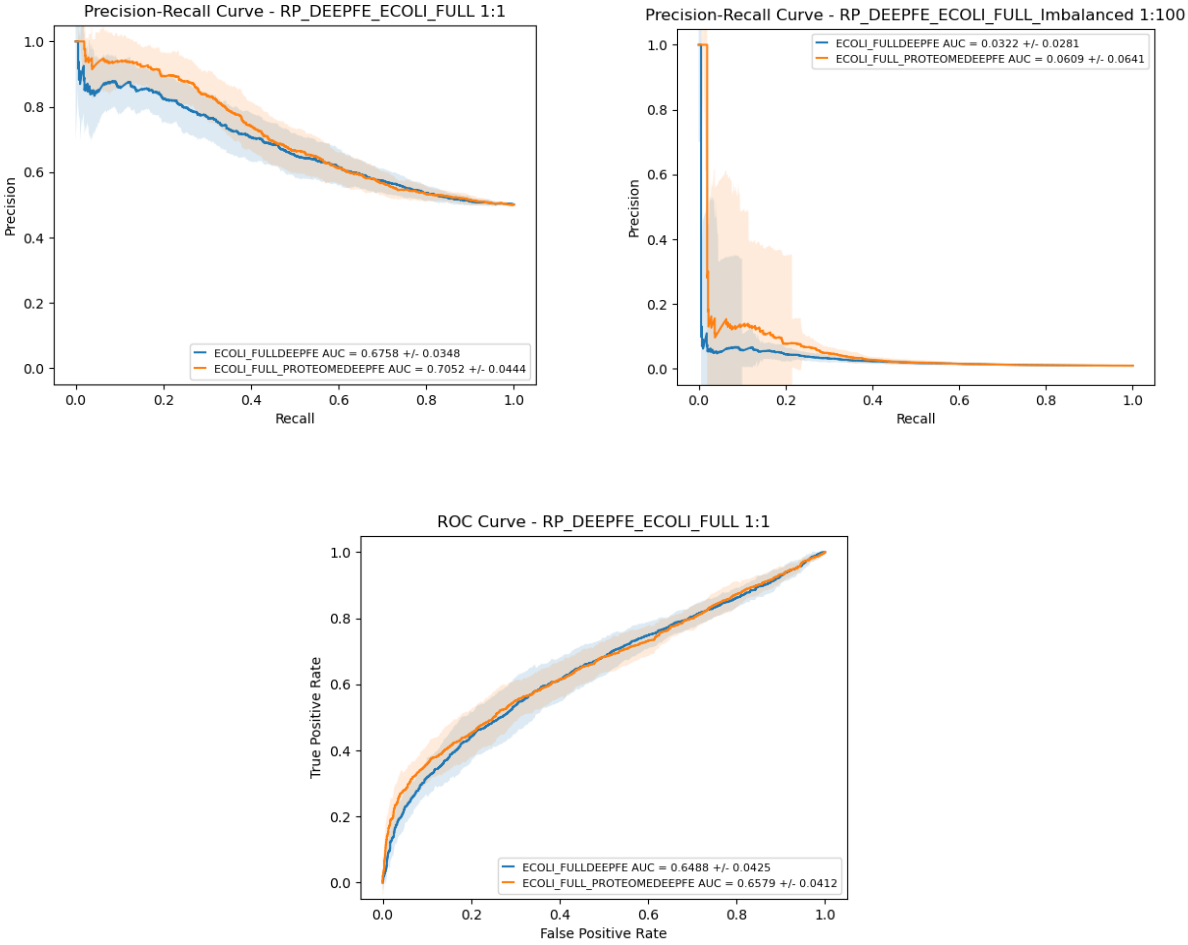

Figure 77: Comparison of RP-DEEPFE performance using features extracted by from ECOLI_FULL proteins versus the entire proteome.

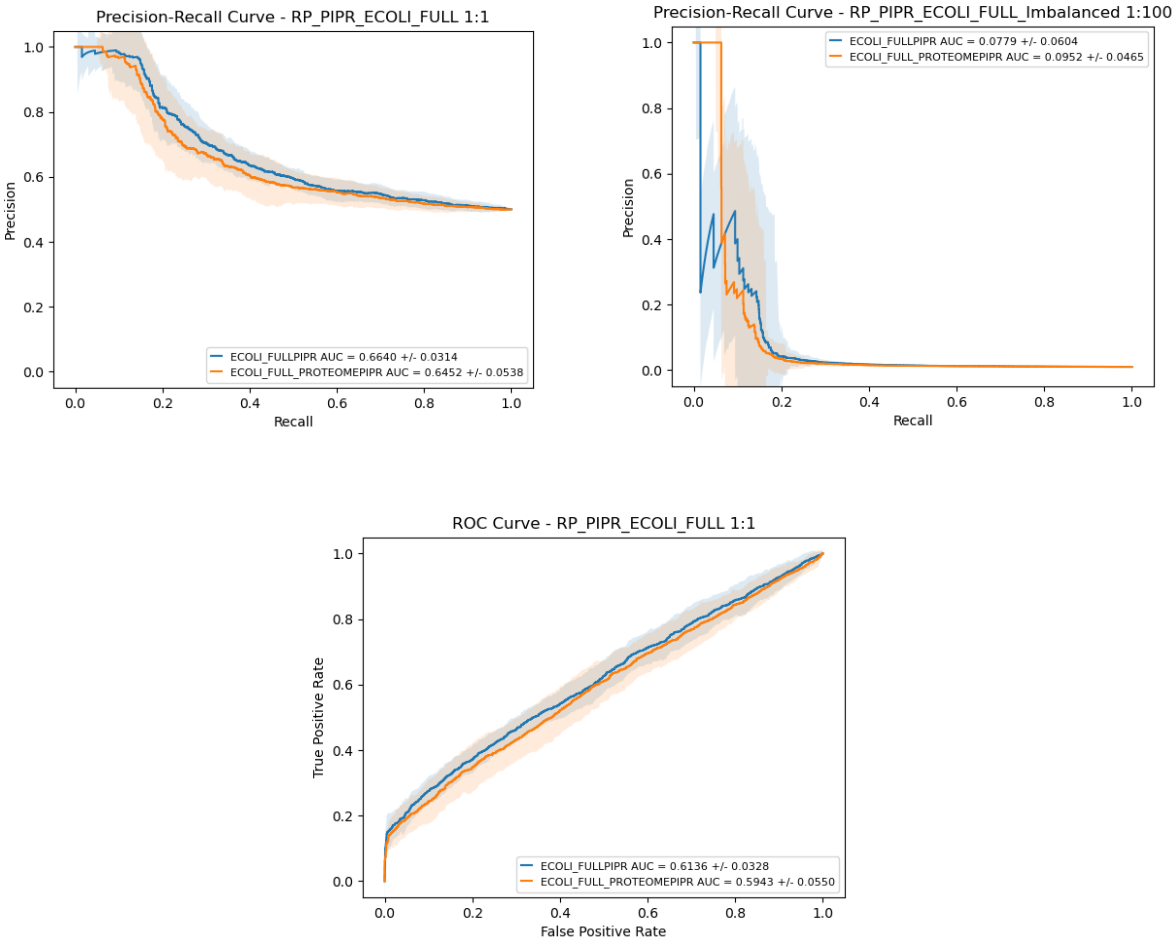

Figure 78: Comparison of RP-PIPR performance using features extracted by from ECOLI_FULL proteins versus the entire proteome.

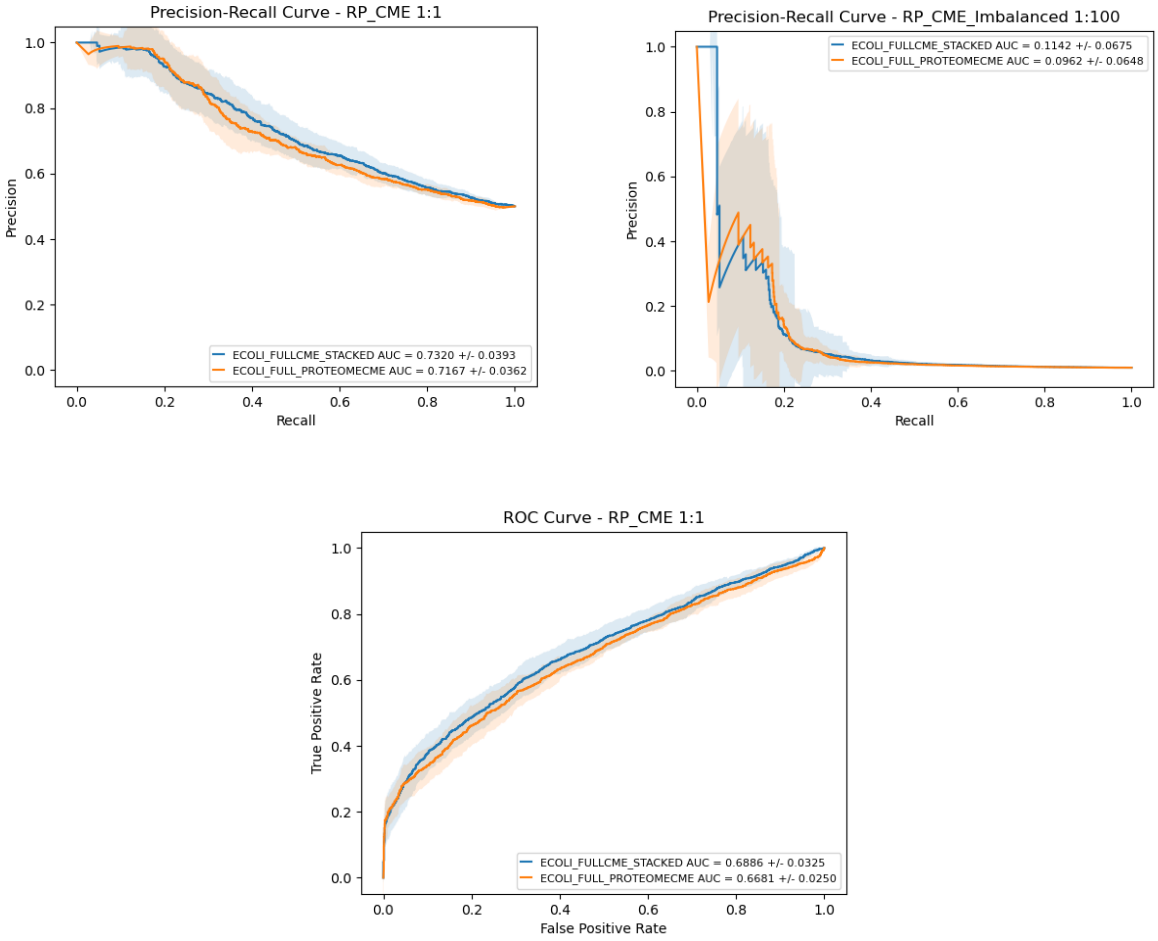

Figure 79: Comparison of stacked RP-CME performance using features extracted by from ECOLI_FULL proteins versus the entire proteome.

C.2 Performance metrics scored at median classification threshold

Table 18: Performance metrics for base classifiers on ECOLI_FULL 10-fold cross-validation, balanced.

Metric	SPRINT	DPPI	DEEPFE	PIPR
Accuracy		0.58682 ± 0.05255	0.54002 ± 0.02468	0.56456 ± 0.02346
Precision		0.59811 ± 0.06250	0.62026 ± 0.08561	0.55872 ± 0.02098
Recall		0.55498 ± 0.05298	0.22974 ± 0.05911	0.61859 ± 0.05944
Specificity		0.61862 ± 0.10168	0.85036 ± 0.06465	0.51056 ± 0.05749
F1-score		0.57347 ± 0.04472	0.32886 ± 0.06227	0.58575 ± 0.03118
MCC		0.17528 ± 0.10674	0.10675 ± 0.06641	0.13071 ± 0.04783
auROC	0.59917 ± 0.01973	0.63720 ± 0.07513	0.53771 ± 0.04667	0.59477 ± 0.02967
auPR	0.63249 ± 0.02656	0.62269 ± 0.05563	0.57284 ± 0.04763	0.59582 ± 0.02467
auROC overall	0.59787	0.63933	0.53788	0.59371
auPR overall	0.62775	0.61893	0.56208	0.59221

Table 19: Performance metrics for base classifiers on ECOLI_FULL 10-fold cross-validation, imbalanced.

Metric	SPRINT	DPPI	DEEPFE	PIPR
Accuracy		0.61799 ± 0.10057	0.84422 ± 0.06361	0.51163 ± 0.05653
Precision		0.01525 ± 0.00395	0.01774 ± 0.00780	0.01255 ± 0.00101
Recall		0.55498 ± 0.05298	0.22974 ± 0.05911	0.61859 ± 0.05944
Specificity		0.61862 ± 0.10168	0.85036 ± 0.06465	0.51056 ± 0.05749
F1-score		0.02964 ± 0.00747	0.03217 ± 0.01239	0.02459 ± 0.00195
MCC		0.03700 ± 0.02352	0.02570 ± 0.01867	0.02572 ± 0.00930
auROC	0.59917 ± 0.01973	0.63720 ± 0.07513	0.53771 ± 0.04667	0.59477 ± 0.02967
auPR	0.06843 ± 0.03690	0.03801 ± 0.01534	0.04240 ± 0.03579	0.06022 ± 0.02430
auROC overall	0.59787	0.63933	0.53788	0.59371
auPR overall	0.02426	0.03147	0.01348	0.05975

Table 20: Performance metrics for RP-enhanced classifiers on ECOLI_FULL 10-fold cross-validation, balanced.

Metric	RP-SPRINT	RP-DPPI	RP-DEEPFE	RP-PIPR
Accuracy	0.53828 ± 0.1381	0.60810 ± 0.02062	0.61534 ± 0.03582	0.57958 ± 0.01627
Precision	0.97340 ± 0.04807	0.61865 ± 0.03120	0.64666 ± 0.04571	0.59437 ± 0.01805
Recall	0.07856 ± 0.02690	0.57404 ± 0.05222	0.50997 ± 0.07504	0.50095 ± 0.04941
Specificity	0.99800 ± 0.00332	0.64212 ± 0.06316	0.72073 ± 0.05438	0.65820 ± 0.03432
F1-score	0.14430 ± 0.04567	0.59327 ± 0.02763	0.56760 ± 0.05582	0.54252 ± 0.03189
MCC	0.19189 ± 0.04019	0.21817 ± 0.04228	0.23703 ± 0.07254	0.16152 ± 0.03253
auROC	0.63063 ± 0.05582	0.65316 ± 0.02949	0.64883 ± 0.04035	0.61384 ± 0.03113
auPR	0.68916 ± 0.05614	0.68231 ± 0.02788	0.67824 ± 0.03301	0.66530 ± 0.02975
auROC overall	0.62822	0.65290	0.64884	0.61356
auPR overall	0.68394	0.67847	0.67578	0.66397

Table 21: Performance metrics for RP-enhanced classifiers on ECOLI_FULL 10-fold cross-validation, imbalanced.

Metric	RP-SPRINT	RP-DPPI	RP-DEEPFE	RP-PIPR
Accuracy	0.98890 ± 0.00334	0.64145 ± 0.06214	0.71864 ± 0.05354	0.65664 ± 0.03362
Precision	0.73493 ± 0.40582	0.01614 ± 0.00224	0.01838 ± 0.00329	0.01448 ± 0.00106
Recall	0.07856 ± 0.2690	0.57404 ± 0.05222	0.50997 ± 0.07504	0.50095 ± 0.04941
Specificity	0.99880 ± 0.00332	0.64212 ± 0.06316	0.72073 ± 0.05438	0.65820 ± 0.03432
F1-score	0.12934 ± 0.05114	0.03136 ± 0.00416	0.03543 ± 0.00619	0.02815 ± 0.00203
MCC	0.21883 ± 0.09849	0.04536 ± 0.01005	0.05169 ± 0.01681	0.03330 ± 0.00690
auROC	0.63063 ± 0.05582	0.65316 ± 0.02949	0.64883 ± 0.04035	0.61384 ± 0.03113
auPR	0.15222 ± 0.06447	0.09675 ± 0.05051	0.06205 ± 0.02661	0.14773 ± 0.05727
auROC overall	0.62822	0.65290	0.64884	0.61356
auPR overall	0.06078	0.03753	0.03220	0.07793

Table 22: Performance metrics for stacked RP-CME classifier on ECOLI_FULL 10-fold cross-validation, balanced.

Metric	RP-CME
Accuracy	0.57230 ± 0.02303
Precision	0.97652 ± 0.03447
Recall	0.14760 ± 0.04564
Specificity	0.99700 ± 0.00332
F1-score	0.25383 ± 0.06984
MCC	0.27043 ± 0.05317
auROC	0.69001 ± 0.03083
auPR	0.73272 ± 0.03726
auROC overall	0.68856
auPR overall	0.73198

Table 23: Performance metrics for stacked RP-CME classifier on ECOLI_FULL 10-fold cross-validation, imbalanced.

Metric	RP-CME
Accuracy	0.98859 ± 0.00339
Precision	0.60854 ± 0.39546
Recall	0.14760 ± 0.04564
Specificity	0.99700 ± 0.00332
F1-score	0.21289 ± 0.07253
MCC	0.27315 ± 0.11588
auROC	0.69001 ± 0.03083
auPR	0.17493 ± 0.06407
auROC overall	0.68856
auPR overall	0.11416

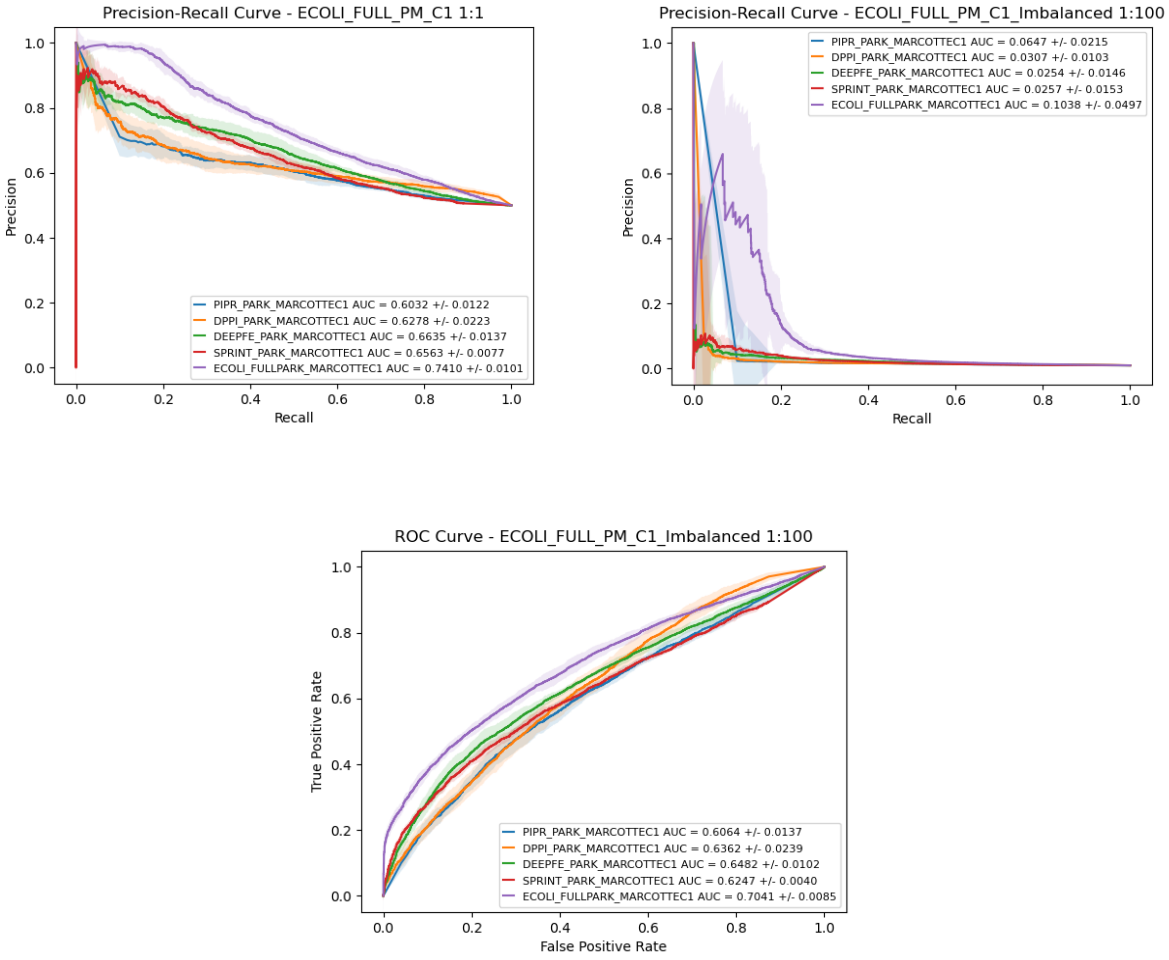

Figure 80: Park and Marcotte evaluation of stacked RP-CME versus base classifiers on C1 test set.

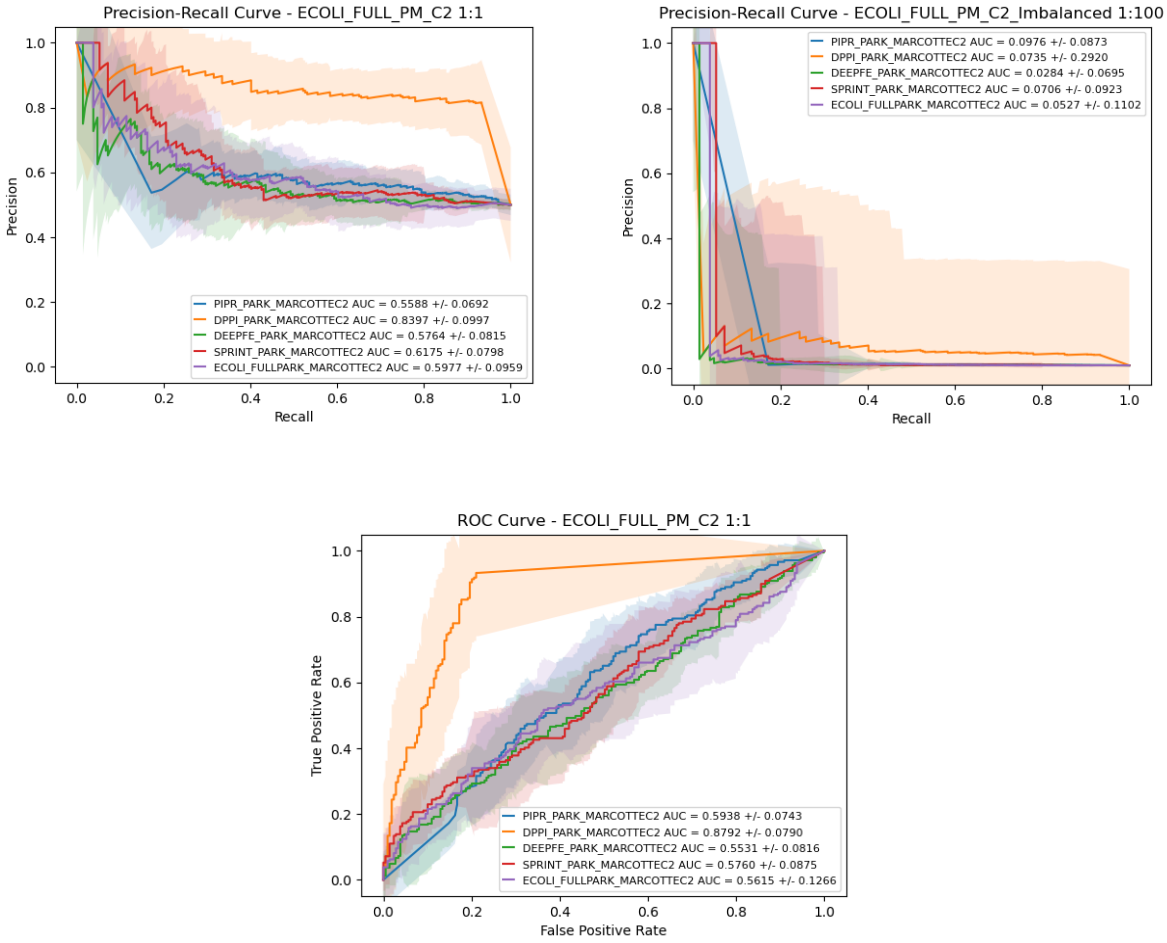

Figure 81: Park and Marcotte evaluation of stacked RP-CME versus base classifiers on C2 test set.

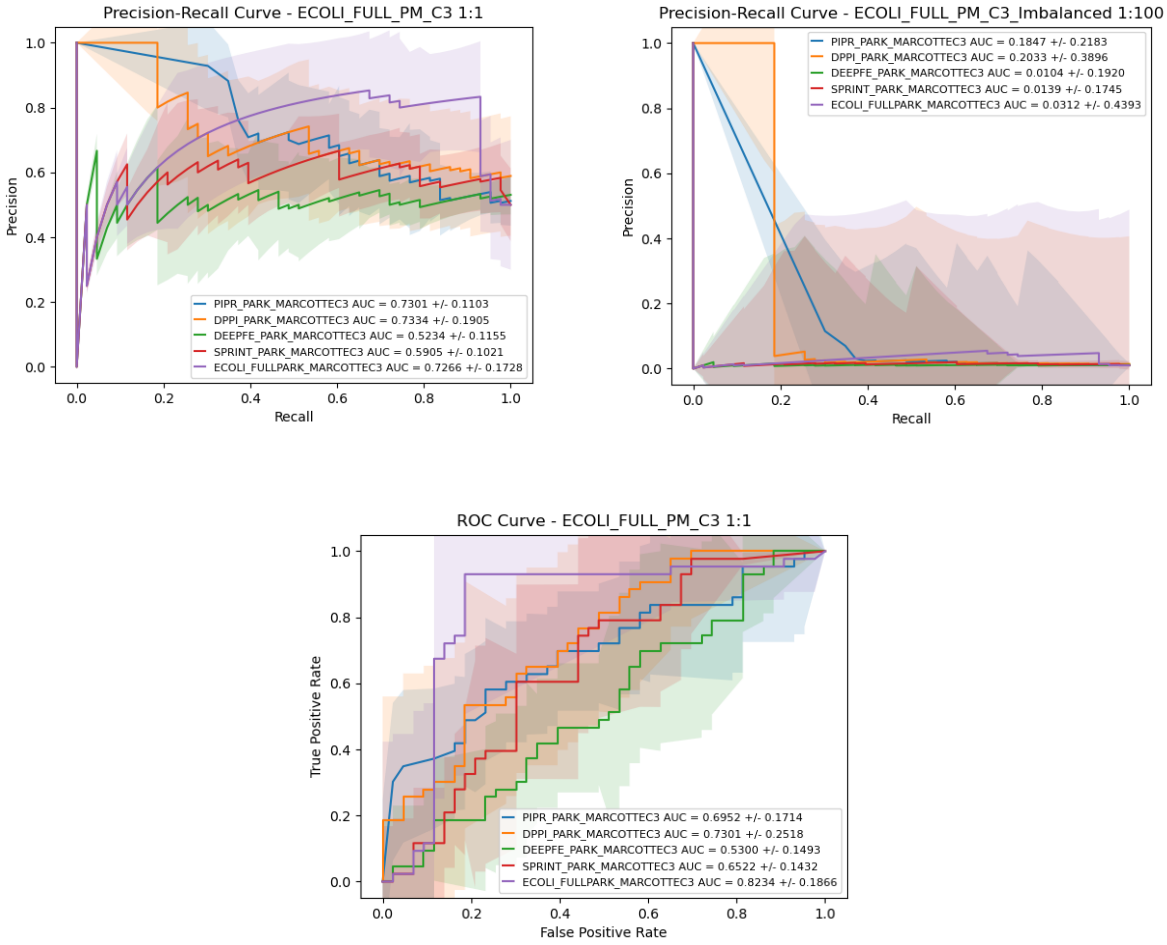

Figure 82: Park and Marcotte evaluation of stacked RP-CME versus base classifiers on C3 test set.