Optimization of 6-Node Graphlet Features for Protein Interaction Predictions

Project Report submitted in partial fulfillment of requirements
For the degree of
Bachelor of Computer Science and Engineering
of
Computer Science and Engineering Department
of
Jadavpur University

by

Debadrita Roy
Regn. No.- 148841 of 2019 - 2020
Exam Roll No. - CSE238001

under the supervision of

Dr. Subhadip Basu
Professor

Department of Computer Science and Engineering
Jadavpur University
Kolkata, West Bengal, India
2023

Abstract

Proteins rarely carry out their tasks in isolation but function largely as complexes that are involved in performing and regulating a variety of biological activities. Hence predicting protein complexes is of extensive research interest. This work focuses on predicting protein complexes using graphlet-based features constructed from a protein-protein interaction network. We show that one can accurately predict protein complexes based only on the structural properties gleaned from the PPI network and find the appropriate number of graphlet-based features needed to achieve the highest performance. We find that this performance is higher than those achieved by existing studies on protein complex predictions.

1 Introduction

Graphlets are small, connected, non-isomorphic, induced subgraphs in a network. N. Pržulj [6] in 2007 proposed the usage of graphlets in determining the structural similarity between large networks. Pržulj defined 30 graphlets having 2 to 5 nodes. The nodes were divided into automorphism orbits, which define the roles of the nodes within the graphlet. There were 73 orbits defined for 5 node graphlets (15 for 4 node graphlets) and the number of times a particular node appears in each of these orbits gives a finer description of the node's neighbourhood. Thus, we can characterise each node by a 73-dimensional vector if we consider all 5 node graphlets and a 15-dimensional vector considering all 4 node graphlets. Melckenbeeck et al (2016) [4] proposed a new way to give all graphlets a unique ordering so that it can be easily extended to name graphlets of any size. An algorithm extending the previous orbit counting algorithms was also proposed which could be used to count larger graphlets (previously, only graphlets having upto 5 nodes could be counted efficiently).

This work considers orbit counts upto those for graphlets having 6 nodes and focuses on finding the most significant orbits when trying to use their counts as features for protein complex prediction. A protein complex is defined as a group of two or more proteins that physically interact and form a quaternary structure. The DIP (Database of Interacting Proteins) dataset is used to construct graphlet orbit count vectors for every protein, i.e., how many times the protein node acts as the specified automorphism orbit in the network. After constructing feature vectors for every protein, the CORUM dataset is used to find the positive examples of protein complexes whereas negative examples are constructed using a combination of protein complexes from CORUM, MCODE and MCL. Random Forest algorithm is used to develop a prediction model. The performance is compared when using different subsets of all the orbit counts as features and the best feature vector (subset of graphlet orbit counts) is noted.

The main contributions of this work are:

1. We extend previous work on protein complex predictions by considering graphlet degree vectors with 6 node orbits instead of 4 or 5 node orbits.

2. We experimentally find the subset of graphlet orbit counts that best describe each protein in the protein complex prediction task.

3. We compare the performance of the experimentally found best subset of orbit count vectors against those of existing studies and find that the performance of the proposed feature vector is superior.

2 Related Work

Graphs have been extensively used in the field of bioinformatics, especially in modelling of protein-protein interactions and genome interactions, with nodes representing biomolecules such as genes, proteins, etc., and edges representing physical, chemical, or functional interactions between them. Protein networks are large, so comparing them is computationally intensive and infeasible, especially exhaustive-search or brute force algorithms. So, efficient heuristic algorithms (Kashtan et al., 2004[2]; Przulj et al., 2006[8]) have been proposed for protein network comparison. Although global properties of large networks are easy to compute, they are inappropriate for use on incomplete networks because they can at best describe the structure produced by the biological sampling techniques used to obtain the partial networks. Therefore, bottom-up or local heuristic approaches for studying network structure have been proposed (Milo et al., 2002[5]; Shen-Orr et al., 2002[9]; Przulj et al., 2004[7]). N. Pržulj in 2007[6] proposed the usage of graphlets (their orbit counts) in determining the structural similarity between large networks. It was proposed that one can characterise each node (protein) in the network by a 73-dimensional vector if one considers 5 node graphlets and a 15-dimensional vector considering 4 node graphlets. This vector was called the graphlet degree vector. A naive approach to counting this vector for each node by directly enumerating each graphlet had time complexity of $O(nd^{k-1})$, n being the number of nodes (usually 1000 to $10,000$), d the maximal node degree (usually 100) and k the graphlet size (4 or 5). T Hočevar and J Demšar[1] proposed a combinatorial approach to counting the orbits. For computing node orbit counts of graphlet size k, they enumerated graphlets of size k-1 and a single graphlet of size k. They then used these relations to calculate the desired orbit counts. ORCA (ORbit Counting Algorithm) was optimised for 4-node and 5-node graphlets and sparse undirected graphs, and decreased the time complexity by a factor of d when compared with the direct enumeration algorithms. Extensions to the ORCA algorithm were proposed by Melckenbeeck et al (2015)[4][3]. They proposed two techniques. The first allows to generate the equations needed in an automatic way, eliminating the tedious work needed to do so manually each time an extra node is added to the graphlets. It is independent on the number of nodes in the graphlets and can thus be used to count larger graphlets than previously possible. The second technique gives all graphlets a unique ordering which is easily extended to name graphlets of any size. Yaveroglu et al (2014)[10] designed a superior graphlet-based measure by identifying and eliminating redundancies and exploiting dependencies between orbit counts in a network and compared the performance of the resulting graphlet degree vector with that of the original graphlet degree vector.

3 Datasets and Data Preparation

3.1 Building Interaction Graph for Feature Extraction for Protein Complexes

DIP (Database for Interacting Proteins) is used to generate graphlet orbit count vectors for each protein. It consists of a list of edges between two nodes where each edge represents that the corresponding node (protein) interacts with the other. We consider all the proteins present in the protein complexes and filter out the edges containing them from the DIP network. Then we build an interaction network using these selected edges. This network is used to compute graphlet orbit count vectors for each protein (node).

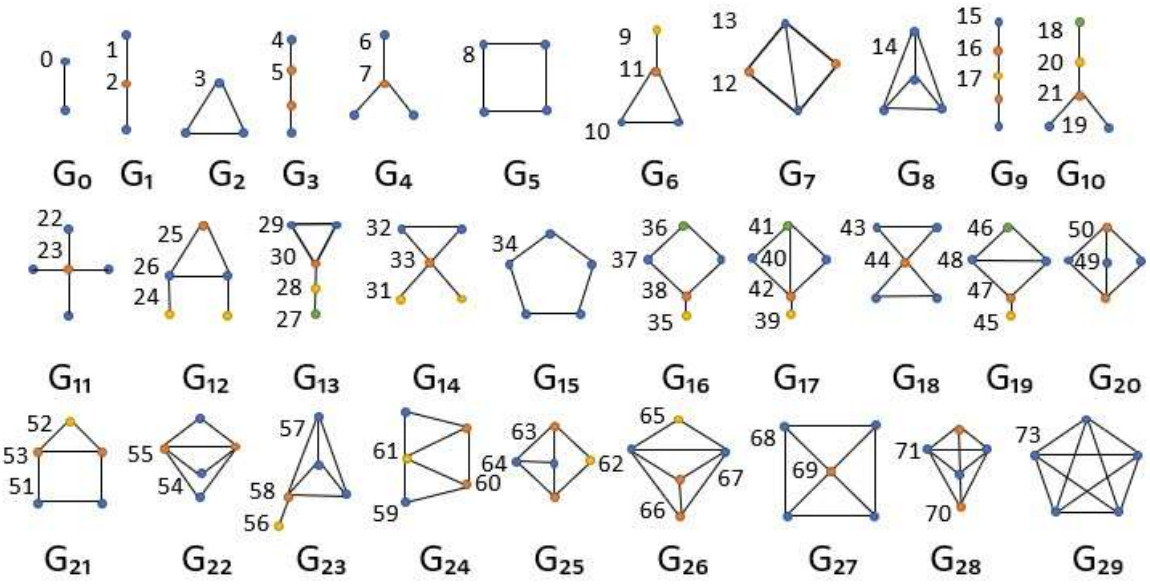

Figure 1: Orbits 0 to 72 for Graphlets having upto 5 nodes. Nodes with the same colour in a particular graphlet belong to the same orbit.

3.2 Preparation of Positive Examples for Protein Complexes

The CORUM (Comprehensive Resource Of Mammalian Protein Complexes) dataset is used for obtaining positive examples for the classification task. CORUM contains a list of protein complexes, each having 3 or more proteins. Only those protein complexes are considered whose constituent proteins are in DIP, so that features can be extracted for each of them. To maintain uniformity, 3 proteins are randomly chosen from each protein complex to make a new positive example. The assumption here is that because all the proteins in the complex interact with each other in some way, all 3-protein combinations in the complex will also interact with each other. Of the 1118 CORUM complexes considered (those with all constituent proteins in DIP), 1118 random 3-protein combinations are taken. 118 are kept as holdout set while 1000 are kept for performing 10-fold cross-validation. The features for each individual protein are extracted and the features for the 3 proteins in a complex are concatenated to form the positive example.

3.3 Preparation of Negative Examples for Protein Complexes

CORUM, MCODE and MCL clusters are used to form the negative data instances. First, the proteins from all the clusters and complexes are merged into a dictionary, with the protein as key and the frequency of occurrence as value. The proteins which were encountered only once were selected and all possible 3- protein combinations made out of them. Out of these combinations, 1118 are randomly chosen and divided into holdout set and set for cross-validation as for the positive examples. The features for each individual protein are extracted and the features for the 3 proteins in a complex are concatenated to form the negative example.

4 Methodology

The orbit counts for all graphlets upto 6 nodes were calculated. Then a number of different methods were considered in order to find the most significant features for prediction. There are 480 orbit counts considering all the graphlets upto 6 nodes, most of which are not of much importance for prediction. The results of the different methods were compared in order to figure out which is the best subset of features for this problem. The results were also compared to results achieved on the same dataset using related methods suggested by other contemporary research.

The proposed Protein Complex Prediction methodology consists of modules: Protein graph construction, feature extraction, feature selection and a classifier that predicts whether the given proteins form a complex or not. This section details each module and their implementation. All experiments were performed in accordance with relevant guidelines and regulations.

4.1 Graph Representation of Proteins

In this work, a file obtained from DIP was used to build a protein molecular graph, also known as a protein network. The file contains border information using two corresponding columns. Let $G(V, E)$ be a graph representing proteins. Here, each vertex ($v \in V$) is a residue (protein), and interactions between residues are described by edges ($e \in E$). Existence of an edge in the file indicates that the corresponding proteins (nodes or vertices) interact with each other.

4.2 Mathematical Formulation

An orbit is a set of vertices from a graphlet which can be mapped onto each other by an automorphism of the graphlet. The more vertices are allowed in a graphlet, the more different graphlets there are. Within a simple graph with n vertices, there are $n * (n-1)/2$ possible edges, which can independently be present or absent. So, a loose upper bound for the number of possible graphlets is $2^{n/2}$. However, not all these graphs will be connected and there will be some graphs which are isomorphic to others. So, the actual number of graphlets will be less.

Graphlet degree vector is formed by computing the counts for all the graphlet orbits, i.e., the number of vertices in the explored graph that touch orbit o for all orbits considered. For example, we count how many nodes touch one graphlet G_1 at an end-node(i.e., at orbit 1), how many nodes touch G_1 at a mid-node (i.e., at orbit 2), how many nodes touch G_3 at an end-node (i.e., at orbit 4) etc. The Orbit Counting Algorithm (ORCA) proposed by T Hočevar and J Demšar (2013)[1] takes a combinatorial approach to counting the above. A system of linear equations demonstrating the relationship between the different node orbits is set up and solved using backward substitution to get the corresponding counts (i.e., we have to directly enumerate one orbit count, usually the one representing nodes of the complete graphlet having n nodes as it is assumed that a complete graphlet will occur less frequently in a network than other graphlets). Thus, for computing node orbit counts of graphlet size n, one enumerates graphlets of size n-1 and a single graphlet of size n.

We demonstrate the setting up of the equations using an example.

Consider the orbits 12 and 14 in graphlets G_7 and G_8.

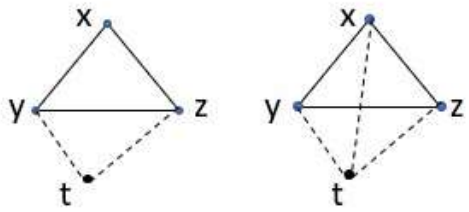

Figure 2: Demonstration of setting up of an equation involving orbits 12 and 14 by using G_2 as base

In Fig. 2, we consider the graphlet G_2 as the base with x, y and z as vertices. Now, t can either be not connected to x or be connected to x. If t is not connected to x, then t appears in o_{12}. Else, t appears on o_{14}. Thus, if we consider the orbit count of t for this particular configuration of x,y,z,o'_{12} + o'_{14} = c(y,z) - 1.Summing up for all configurations x,y,z in G and such that each configuration appears once, the RHS becomes $\sum_{y,z:y<z,G[x,y,z]\cong G_2} c(y,z)-1$. Now, even if $y < z$, o_{14} is counted 3 times.[x,y,z can be permuted in 3!/2 ways (divided by 2 because 3 permutations are prohibited due to $y < z$ condition)]. So, the final equation looks like $o_{12} + 3o_{14} = \sum_{y,z:y<z,G[x,y,z]\cong G_2} c(y,z) - 1$

Similarly, the rest of the equations can be set up. All the orbit counts are then computed using backward substitution by using the enumerated count of o_{72} and are then used as the features for the given node (protein).

Melckenbeeck et al (2016)[4] extended this concept to n nodes (instead of 4 or 5) and proposed an algorithm to automate the generation of the equations, thereby reducing manual work. An automatic generalised naming system for larger graphlets (based on some features from Przulj's naming system) was also proposed by them. These systems[3] have been used in this work for naming orbits for 6-node graphlets and for calculating the orbit counts in the network.

4.3 Selection of Significant Orbits

The orbit counts generated in the previous section represent the local structural properties of a protein in the network. However, the feature space if all the orbit counts are considered will be very high-dimensional as the number of orbits for 6 node graphlets is 480. Moreover, for even a large PPI network, most of the orbit counts (especially for the more complex graphlets) are zero or do not contribute significantly to the protein features. So, selection of the significant orbits is necessary to reduce the dimensions of feature space and to remove redundant features.

The following methods have been used for selecting orbits.

4.3.1 Based On Variances

The variance for each graphlet orbit count is calculated over the values for all the proteins in the interaction network. The orbits are then sorted in descending order and the top k orbits are selected to be the features for each protein. The orbit selection is done using different values of k and the results obtained are compared.

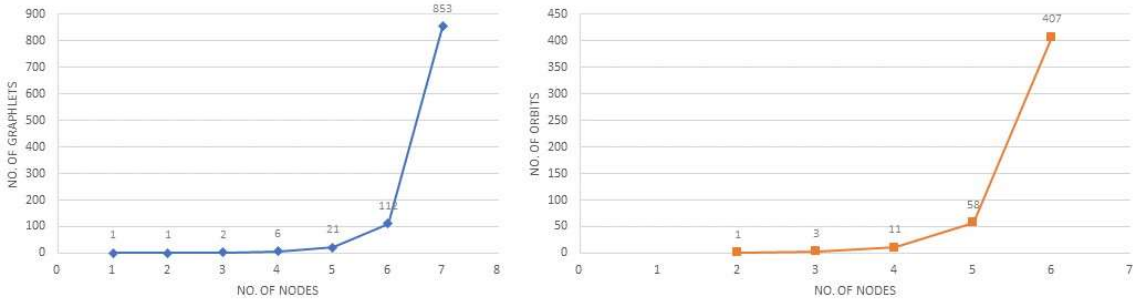

Figure 3: Graphs demonstrating the number of graphlets induced for a given number of nodes and the number of orbits for a given number of nodes: As is evident, there is an exponential rise in the curve after number of nodes becomes greater than 6, so only graphlets having less than or equal to 6 nodes are considered in this work.

4.3.2 Based on Complexity of Graphlets

As the number of nodes in the graphlets increase, the structure becomes more complex. So, the performance is measured while only considering graphlets of less than or equal to 4 nodes, 5 nodes or 6 nodes.

4.3.3 Based on Elimination of redundant orbits (proposed by Yaverouglu et al)[10]

Graphlet degrees coming from large graphlets are dependent on the graphlet degrees coming from the smaller ones. So, Some orbits are redundant (their counts in a network can be derived from the counts of other orbits). For example, if two edges (a, b) and (a, c), i.e., orbit 0, are combined, then the orbit touching a from the graphlet induced by a,b,c is either 2 (if b and c are not connected) or 3 (if b and c are connected). Thus, the count of orbit 0 is dependent on the counts of orbits 2 and 3. Yaverouglu et al considered all graphlets of size less than or equal to 5 nodes and came up with a subset of orbits where there are no redundancies, i.e., the count of an orbit in the subset cannot be computed using solely the counts of the other orbits in the subset. The following list of orbits is considered to form a non-redundant subset: 0, 1, 2, 4, 5, 6, 7, 8, 9, 10, 11, 15, 18, 19, 22, 24, 25, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 40, 41, 42, 43, 45, 46, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 70. So, only the above orbits have been selected and performance measured. It has been done in two ways: considering the orbits only upto 4 node graphlets and considering the orbits upto 5 node graphlets.

4.3.4 Based on a Combination of Graphlet Complexity and Cumulative Variance of the features

The variance for each graphlet orbit count is calculated over the values for all the proteins in the interaction network. Cumulative variance for each graphlet count is calculated by summing up the variances for all previous orbits and the current orbit. Smaller graphlets are less complex than the larger ones and thus are of more significance in PPI networks. So, the orbits are selected as the first m orbits where the cumulative variance of the mth orbit is 0.90 or 0.95 or 0.99, etc.

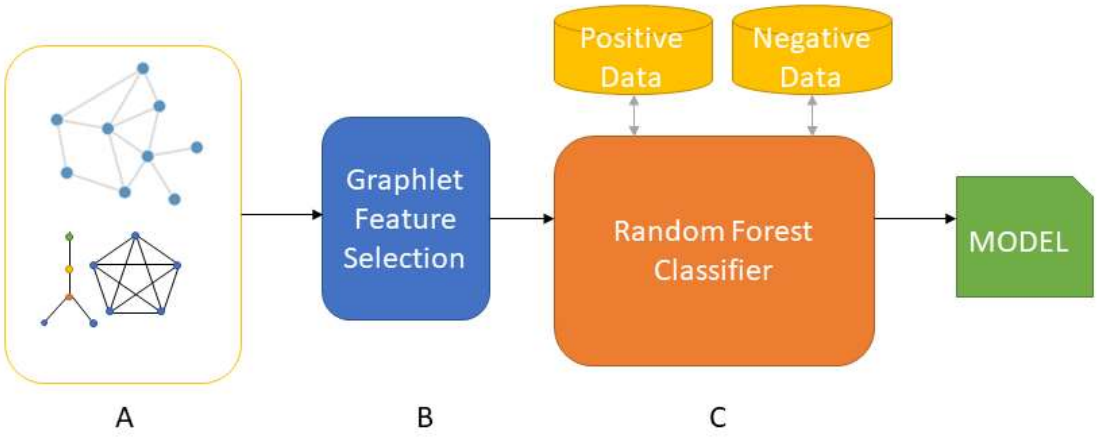

Figure 4: Different Modules used to perform the classification task. Module A calculates the graphlet orbit counts for all the proteins in the interaction network. The graphlet-based features are then fed into module B, which performs feature selection according to some predefined procedure. The selected features and the positive and negative data examples are fed into module C, which is a random forest classifier and trains a model.

4.4 Protein Complex Prediction

A Random Forest Classifier is used to predict whether the given proteins form a complex or not. The feature extraction for each protein has been already detailed. The positive and negative examples of complex formation using 3 proteins have been prepared using the previously detailed methods. After utilising a suitable selection procedure as described above, the selected features for the 3 given proteins are combined to form the features for the given data instance. The Random Forest Classifier is used to train the model using given data.

4.5 Performance Measurement

The following evaluation metrics are used to compare the performance of the different subsets of selected features.

4.5.1 Accuracy

It is the ratio of the number of correct predictions made by the model to the total number of predictions made by the model. So,

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

where, TN = Number of negative predictions which were classified as negative
TP = Number of positive predictions which were classified as positive
FP = Number of negative predictions which were classified as positive

FN = Number of positive predictions which were classified as negative

Higher Accuracy means that more data instances have been classified correctly, i.e., the performance of the model is better.

4.5.2 F1 Score

F1 score is defined as the harmonic mean of precision and recall.

$$F1Score = \frac{2 * Precision * Recall}{Precision + Recall}$$

where,

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

Higher F1 Score indicates that both precision and recall are high, i.e., the model is better.

4.5.3 Area under ROC Curve

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the True Positive Rate against False Positive Rate at various threshold values. In other words, it shows the performance of a classification model at all classification thresholds. The Area Under the Curve (AUC) is the measure of the ability of a binary classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the model's performance at distinguishing between the positive and negative classes. When AUC = 1, the classifier can correctly distinguish between all the Positive and the Negative class points.

4.5.4 Area under Precision-Recall Curve

The precision-recall curve is constructed by calculating and plotting the precision against the recall for a single classifier at a variety of thresholds. AUC-PR is the area under this curve. The higher the AUC-PR score, the better a classifier performs for the given task. In a perfect classifier, AUC-PR = 1.

4.5.5 Matthew's Correlation Coefficient

The Matthews correlation coefficient (MCC) is used as a measure of the quality of binary (two-class) classifications. It is essentially a correlation coefficient between the observed and predicted binary classifications and returns a value between -1 and $+1$. A coefficient of $+1$ represents a perfect prediction, 0 no better than random prediction and 1 indicates total disagreement between prediction and observation.

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

4.5.6 True Positive

It is the number of predictions which actually belong to the positive class and which are classified as positive by the model.

4.5.7 True Negative

It is the number of predictions which actually belong to the negative class and which are classified as negative by the model.

4.5.8 False Positive

It is the number of predictions which actually belong to the negative class and which are incorrectly classified as positive by the model.

4.5.9 False Negative

It is the number of predictions which actually belong to the positive class and which are incorrectly classified as negative by the model.

5 Results and Discussion

This section describes and compares the performance of the proposed methods for predicting Protein Complex Formation in terms of various evaluation metrics as described above.

5.1 Experimental Setup

We use the scikit-learn toolkit to build the propose Protein Complex Prediction model. The Random Forest Classifier is used to train the model.

The parameters are chosen as follows:

No. of estimators = 300, Maximum Depth = 4, Maximum features = 0.3

5.2 Performance

The performances of the different selection methods based on approaches mentioned earlier are enumerated in the following tables. The performance was evaluated using 10-fold cross-validation and using holdout test set. The evaluation metrics in the tables, in order, are Accuracy, Area under the ROC Curve, Area under the Precision-Recall Curve, F1 Score, Matthews Correlation Coefficient, Number of True Positives, Number of True Negatives, Number of False Positives and Number of False Negatives. The values are averaged over 10 folds.

K	Accu	AUC	AUPRC	F1	MCC	TP	TN	FP	FN
100	0.94	0.98	0.98	0.94	0.89	89.10	99.50	0.50	10.90
150	0.95	0.98	0.98	0.94	0.90	89.90	99.40	0.60	10.10
200	0.94	0.98	0.98	0.94	0.89	89.30	99.60	0.40	10.70

Table 1: Performance of Protein Complex Prediction using the Orbit counts having top K Variances as node features (10-fold cross-validation)

K	Accu	AUC	AUPRC	F1	MCC	TP	TN	FP	FN
100	0.92	0.97	0.98	0.91	0.85	99.3	117.9	0.1	18.7
150	0.93	0.98	0.99	0.92	0.86	100.5	117.9	0.1	17.5
200	0.93	0.97	0.98	0.92	0.86	100.80	117.90	0.10	17.20

Table 2: Performance of Protein Complex Prediction using the Orbit counts having top K Variances as node features on holdout test set.

No. of nodes	Accu	AUC	AUPRC	F1	MCC	TP	TN	FP	FN
4	0.93	0.96	0.97	0.93	0.87	104.00	116.00	2.00	14.00
5	0.93	0.97	0.98	0.93	0.87	102.50	117.70	0.30	15.50

Table 3: Performance of Protein Complex Prediction using the Non-redundant Orbit counts (proposed by Yaverouglu et al) as node features on holdout test set

No. of nodes	Accu	AUC	AUPRC	F1	MCC	TP	TN	FP	FN
4	0.94	0.97	0.98	0.94	0.89	90.40	98.50	1.50	9.60
5	0.94	0.97	0.98	0.94	0.89	89.00	99.70	0.30	11.00

Table 4: Performance of Protein Complex Prediction using the Non-redundant Orbit counts (proposed by Yaverouglu et al) as node features (10-fold cross-validation)

K	Accu	AUC	AUPRC	F1	MCC	TP	TN	FP	FN
34	0.93	0.98	0.98	0.93	0.87	104.00	115.90	2.10	14.00
60	0.94	0.98	0.98	0.93	0.88	103.00	118.00	0.00	15.00
132	0.95	0.98	0.99	0.95	0.90	106.30	117.70	0.30	11.70
138	0.95	0.98	0.98	0.95	0.91	107.70	117.00	1.00	10.30
144	0.96	0.98	0.99	0.95	0.92	107.70	117.90	0.10	10.30
151	0.91	0.97	0.98	0.91	0.84	97.80	118.00	0.00	20.20
160	0.93	0.98	0.99	0.92	0.86	101.90	116.90	1.10	16.10
229	0.94	0.97	0.98	0.93	0.88	103.90	117.00	1.00	14.10

Table 5: Performance of Protein Complex Prediction using first K Graphlet orbit counts as node features on holdout test set

K	Accu	AUC	AUPRC	F1	MCC	TP	TN	FP	FN
34	0.95	0.98	0.99	0.94	0.90	90.00	99.50	0.50	10.00
60	0.95	0.98	0.98	0.94	0.90	90.10	99.50	0.50	9.90
132	0.95	0.98	0.99	0.95	0.91	90.60	99.90	0.10	9.40
138	0.95	0.98	0.98	0.95	0.90	90.50	99.40	0.60	9.50
144	0.95	0.98	0.98	0.95	0.91	91.20	99.10	0.90	8.80
151	0.95	0.98	0.99	0.95	0.91	91.10	99.80	0.20	8.90
160	0.94	0.98	0.98	0.94	0.89	89.50	99.20	0.80	10.50
229	0.95	0.98	0.99	0.95	0.90	89.80	100.00	0.00	10.20

Table 6: Performance of Protein Complex Prediction using first K Graphlet orbit counts as node features (10-fold cross-validation)

As is evident, the set of the first 144 orbit counts gives the best performance over all the selection methods. For the selection according to variances, the top 150 orbits having greatest variances gives the best performance. These performances are also better than the performance using the non-redundant set of orbits. For first K orbits, it is found that the performance gets better when value of K increases, then at a certain point it reaches its maximum value and then the performance does not increase for greater values of K.

6 Conclusion

In this work, the different orbit counts upto orbits belonging to 6-node graphlets were considered and used as features for performing protein complex prediction task. Different methods were used

to select subsets of features which were then fed into the random forest classifier in order to generate a model. The performances of the different models were evaluated on holdout set and by performing cross-validation and compared based on different metrics. It was found that some methods perform better than others. It was found that incorporating some orbit features in addition to the first 73 orbits (upto 5-node graphlets) increased the effectiveness of the model. However, the performance tapers down at some point, presumably when the features start becoming redundant. As observed, the best subset of features is the set of the first 144 orbit counts. Also, the protein complex prediction task was performed with high accuracy even though only the structural properties of the protein interaction network were considered.

References

[1] T. Hočevar and J. Demšar. A combinatorial approach to graphlet counting. *Bioinformatics*, 30(4):559–565, 2014.

[2] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20(11):1746–1758, 2004.

[3] I. Melckenbeeck, P. Audenaert, D. Colle, and M. Pickavet. Efficiently counting all orbits of graphlets of any order in a graph using autogenerated equations. *Bioinformatics*, 34(8):1372–1380, 2018.

[4] I. Melckenbeeck, P. Audenaert, T. Michoel, D. Colle, and M. Pickavet. An algorithm to automatically generate the combinatorial orbit counting equations. *PloS one*, 11(1):e0147078, 2016.

[5] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298(5594):824–827, 2002.

[6] N. Pržulj. Biological network comparison using graphlet degree distribution. *Bioinformatics*, 23(2):e177–e183, 2007.

[7] N. Pržulj, D. G. Corneil, and I. Jurisica. Modeling interactome: scale-free or geometric? *Bioinformatics*, 20(18):3508–3515, 2004.

[8] N. Pržulj, D. G. Corneil, and I. Jurisica. Efficient estimation of graphlet frequency distributions in protein–protein interaction networks. *Bioinformatics*, 22(8):974–980, 2006.

[9] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature genetics*, 31(1):64–68, 2002.

[10] Ö. N. Yaveroğlu, N. Malod-Dognin, D. Davis, Z. Levnajic, V. Janjic, R. Karapandza, A. Stojmirovic, and N. Pržulj. Revealing the hidden language of complex networks. *Scientific reports*, 4(1):4547, 2014.