Binding affinity prediction of protein-protein complexes using Machine Learning.

Project report Submitted To

Faculty of Science, Savitribai Phule Pune University

By

Sukrut Digambar Shishupal
(IBB-2014-26)

Under the guidance of Dr. Sukanta Mondal

Institute of Bioinformatics & Biotechnology

Savitribai Phule Pune University
Pune, India

May, 2019

Certificate of Guide

This is to certify that the project entitled "Binding affinity prediction of protein-protein complex using machine learning", submitted by Mr. Sukrut Digambar Shishupal student of 5th year integrated MSc. Biotechnology, Institute of Bioinformatics and Biotechnology (IBB), Savitribai Phule Pune University (SPPU), was carried out under my guidance successfully.

Dr. Sukanta Mondal

Assistant Professor,

Department of Biological Sciences,

Birla Institute of Technology and Sciences (BITS- Pilani), K.K Birla Goa campus

Certificate of Director

This is to certify that Mr. Sukrut Digambar Shishupal (Integrated MSc 5 years, Semester X) has successfully completed his work on "**Binding affinity prediction of protein-protein complexes using machine learning**", in fulfillment of the course IBT-723 P during the period January-2019 to April-2019 in the Birla Institute of Technology and Sciences (BITS) Pilani, K.K. Birla Goa campus satisfactorily under the guidance of Professor Sukanta Mondal.

Professor Smita Zinjarde

Director,

Institute of Bioinformatics and Biotechnology,

Savitribai Phule Pune University (SPPU), Pune.

Acknowledgments

I would like to thank my guide Prof. Sukanta Mondal for his immense and timely guidance while carrying out this project and for constant encouragement to explore new topics which helped me in completing this project.

I would like to extend my thanks to Mr. Tirtharaj Dash for recommending me to my guide and for constantly keeping update about my progress during the work. I would also like to thank him for letting me entry in BITS library so that I can carry out my project there.

I take this opportunity to thank Institute of Bioinformatics and Biotechnology (IBB), SPPU and Birla Institute of Technology and Sciences (BITS) Pilani, K.K Birla Goa campus for granting me this opportunity to work on such an interesting topic.

I would like to thank my friends Surhud Sant, Vallari Ghanekar, Nikita Shah, Nidhi Gujar, Yogesh Bhonde, Durga Pawar to name a few for a timely conveyance.

I also place a sense of gratitude and conduct to all the people who directly or indirectly, have lent their helping hand in this venture.

Sukrut Shishupal

IBB2014-26

Abstract

Title of Project **:** Binding affinity prediction of protein-protein complexes using machine learning

Name of Student : Sukrut D. Shishupal

Roll Number **:** IBB-2014 26

Name of the guide **:** Dr. Sukanta Mondal

Assistant Professor, BITS Pilani, K. K. Birla Goa campus

Duration of Project **:** January 2019 - April 2019

Protein-Protein Interactions play a vital role in most of the biological activities. The study of functional residues (FRs) is necessary for understanding protein functions and biological processes. To understand the FRs, one of the widely used methods is the amino acid network (AAN). This network representation of protein provides a systems approach to topological analysis based on the three-dimensional structure of the complex, irrespective of secondary structure and folding types and provide vital information about the FRs. The current AAN models use two strategies for network construction, node and edge. Fundamentally each amino acid has its own importance and hence, it is necessary to treat each and every node as different. Here we compare two such AAN models, where different features based on protein complex are used and the best model to predict FRs is found out using machine learning. We used a set of 101 protein-protein complexes for which the interacting pairs are heterodimers. We assessed the performance of the model and conclude which parameters are crucial to discern high and low binding affinity complexes.

Keywords: Protein-Protein Interactions (PPIs), Functional residues (FRs), Machine-learning (ML), Amino acid network (AAN), Protein dynamics

Content Page No.

List of figures and tables

Figures:

Tables:

Abbreviation

RIN: Residue interaction network

PPI: Protein-Protein Interaction

AAN: Amino acid network

NAPS: Network-based analysis of protein structure

AACEN: Amino acid contact energy network

DSSP: Dictionary of protein secondary structure

NACEN: Node-weighted amino acid contact energy network

PDB: Protein Data Bank

PSAIA: Protein Structure and Interaction Analyzer

ML: Machine learning

AI: Artificial Intelligence

LOOCV: Leave one out cross-validation

TP: True Positive

TN: True Negative

FP: False Positive

FN: False Negative

Chapter 1: Literature Review

1. Graph Theory:

In mathematics, graph theory is the study of graphs which are mathematical structures in order to build a pairwise relation between multiple objects. The graph generated is made-up of nodes (vertices or points) and edge (links or lines). The first paper related to this theory can be found in 1736 where the knight problem was tried to be solved (1). Knights tour is a sequence of moves of a knight on a chessboard such that the knight visits each square only once and if one of the squares is traced again, the tour ends (2). Leonhard Euler along with Vandermonde worked on this problem and this marked the beginning of the branch of mathematics known as topology.

The term “graph” was first introduced by Sylvester in a paper published in 1878, where he drew the analogy between molecular diagrams (3). In 1936, the first book relating to graph theory was published by Dénes Kőnig (4), and later in 1969, another book by Frank Harary was published which was "considered the world over to be the definitive textbook on the subject" and this led to mathematicians, engineers, scientists to interact with each other in terms of numbers (5). There were many autonomous developments of topology from 1860 to 1930 which helped the theory to blossom. Modern algebra also helped in the development of graph theory, wherein Gustav Kirchhoff in 1845 published his Kirchhoff’s circuit law for calculating the voltage and current in electric circuits. After this paper was published, many researchers from different science fields started looking towards this theory and started making many interesting applications.

Network theory is a part of graph theory wherein the relation between discrete objects is presented. This theory plays an important role in a wide variety of disciplines ranging from computer science, engineering, sociology to molecular and population biology (6). The PPI network holds information about the behavior of different proteins in coordination with others to enable the biological processes within the cell. This is done by examining each and every amino acid individually and understanding the connection (7). Various parameters and their interactions are analyzed and a graph is created where thousands of nodes are connected via edges.

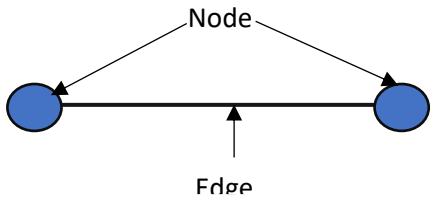

Fig 1.1: Representation of Edge and Node

1.1. **Type of graphs**:

Definitions in graph vary. Here are some basic ways to define a graph and the related mathematics related to it.

1.1.1 Undirected graph:

In this type of graph, we consider a pair (N, E), where N is the nodes while E is the edges representing the connection. For a single connection between node i and j, we can represent it as $E = \{(i, j)| \ i, j \in N\}$. For this case, we can say that i and j are neighbors. For network construction, a multi-edge connection is required where two or more edges have the same endpoint.

1.1.2 Directed graph:

A directed graph consists of triple components (N, E, f) where f is the function that maps each element of E to an ordered pair of the node in N. Ordered pair of nodes are called directed edges. An edge $E = (i, j)$ is having direction from i to j. Such type of graphs is used to describe biological pathways or procedures which show sequential interactions, such as metabolic signal transduction or even regulatory network.

Type of directed graph:

1.1.2.1 Weighted graph:

A weighted graph is defined as a graph G= (N, E) where N is a set of nodes and E is set of edges between the nodes $E= \{(u, n) \mid u, n \in V\}$ associated with a weight function of value w: $E \rightarrow R$, where R denotes set of real numbers. The weight $w_{i,j}$ of the edge between i and j represent the connection. Usually, larger weights represent the high reliability of a connection.

1.1.2.2 Unweighted graph:

An unweighted graph is defined as graph G= (N, E) where N is a set of nodes and E is a set of edges between the vertices E. There is no weight function associated with it and hence, the network generated is usually based on the edge $E= \{(u, n) \mid u, n \in V\}$. The graph generated using this method considers every edge as important.

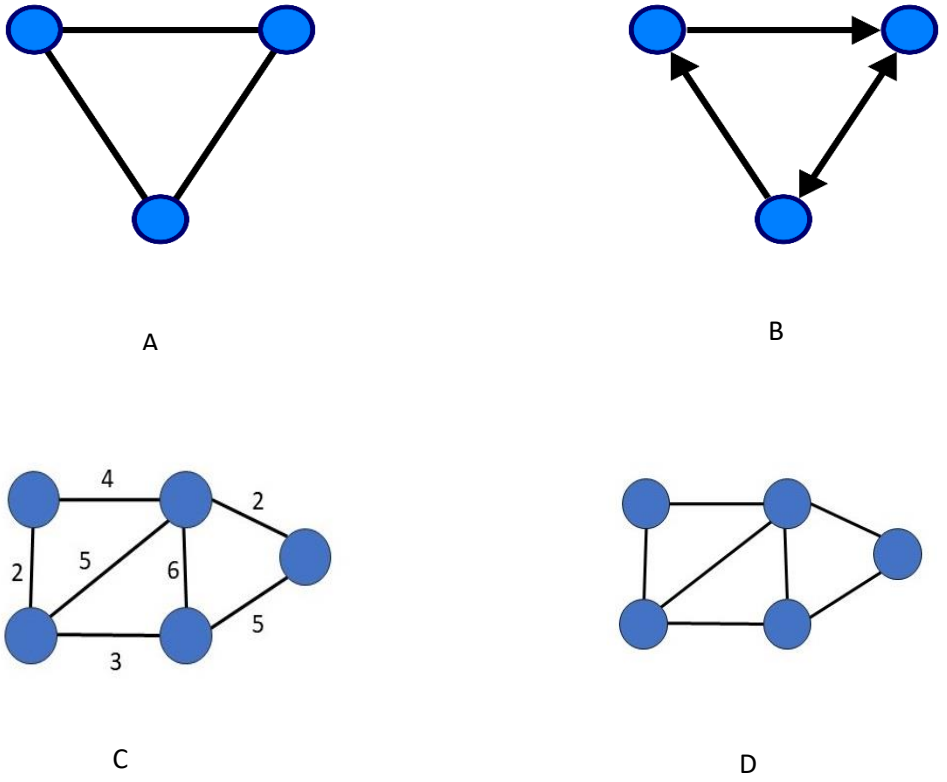

Fig 1.2: Type of graph (A) Undirected, (B) Directed, (C) Weighted, (D) Unweighted

These graphs can be used to model many types of relations in biology, where the node represent amino-acid while the edge makes up for the connections between the different amino acids (8, 9). The protein-protein interaction (PPI) networks are usually represented using the same method, which may be weighted or unweighted graph, depending on the users need. The network in which edges are defined as amino acid residues and their interaction which is based on different topological parameters has gained tremendous popularity with new studies suggesting new insight into protein structure-function relation. Graph generated using such definitions are termed as an amino acid network (AAN) or residue interaction network (RIN) (10). Recently, there is a tendency to integrate AAN features and other structural features in order to predict various properties by using machine learning methods (11).

AAN can be constructed by using the weighted or unweighted node or edges (12). In node-weighted AANs, the nodes are assigned by various weights to check the interaction in the model.

Similarly, in edge-weighted AANs, the edges are assigned weights to model the interaction strength between residues. These weights can be assigned by a number of possible methods such as atom-atom links (13), dynamic simulations (14), energy functions (15), and so on.

Most of the AAN models mostly emphasize topological properties and ignore the importance of heterogeneity of amino acid residues. Such models treat all the nodes as same in the network, hence data can be incorrect using such models (16, 17). While in the case of node weighted, the amino acids are not considered that important and only the number of nodes to a certain edge is considered as important. While the nodes which are weighted differently usually reflect specific features of the diverse amino acids and that improves the complex representation of AANs (18).

The volume of data on PPI's is rapidly increasing due to the improvements in high throughput techniques such as yeast-2-hybrid screening or mass spectrometry (19). Data is been constantly added and it becomes essential to study and understand the correlation between the produced data. Hence, graph theory plays an important role to help correlate the data and make a meaningful conclusion from it. Analysis of the topological parameters of proteins with structures is of great value and is an active field of research. Due to the use of X-ray crystallography, many protein structures are being solved and hence, the need of automated methods for analysis are required due to which the tools from graph theory are being explored for such analysis.

Protein-protein complexes can be classified into various types such as dimeric-multimeric complexes, homodimer-heterodimer complexes and even on the biological significance of the complex. The binding affinity of the protein-protein complexes can be used as one such parameter which can be related to most of the functional aspects of the proteins. The data regarding interacting pairs of proteins have been deposited in databases such as STRING (20), BioGRID (21), DIP (22).

While generating a RIN, there are several methods which can be used to construct this network such as CyToStructure (23), RINalyzer (24), which are plugins for Cytoscape (25) which can be integrated with other features of Cytoscape for analyzing the protein structure. In the case of PyMOL, plugins such as xPyder (26) and PyInteraph (27) are used which include various features for extensive molecular analysis. The major limitation of these tools is that they usually depend on other software and need a specific system requirement.

Hence, sometimes the user needs to download various dependencies in order to run one program. In this study, we compare two such tools, one which generates a graph based on a node while other generates the graph based on the edge. NACEN (28) is an R based package which needs dependencies to run while the other tool NAPS (29) is an online web server which generates the graph instantly and needs no dependencies.

Chapter 2: Genesis of hypothesis

There are few tools available which analyze the protein structures on a network basis. There are two ways in which a network can be constructed, node based or edge based. Thus, the residue interaction network (RIN) generated varies based on the feature the user has used and what parameters one selects. It is difficult to predict which feature will predict the hot patch and thus, we tried to use both the tools and compared them to understand which one gives a better result, node generated graph or edge generated graph. As per our knowledge, there are no studies which compare these two types of graphs and hence, we had to construct a workflow to find out which method of generating the graph will be better for protein complexes of the dataset. Since the graph can be weighted or non-weighted, the features increase and hence, it becomes difficult to select the total number of features which are actually contributing and which are reducing the performance. We selected some features which might play a crucial role in differentiating the complexes. We also used a machine learning tool in order to differentiate between high and low binding affinity complexes from the dataset.

Chapter 3: Materials and Methods

3.1 Generating the graph:

Previous studies have also suggested that topological parameters play an important role in determining crucial residues for protein stability (30), protein dynamics (31), carrying out the enzymatic activity (32) and also understanding protein folding kinetics. There are online websites and standalone tools available in order to visualize and analyze the protein contact maps.

3.1.1 Network-based analysis of protein structure (NAPS):

NAPS (29) is an online tool for network generation which provides features for analysis and interactive visualization of the network. The first step is to provide the protein structure information which can be done by uploading a PDB file from the local machine or by simply entering the four-letter PDB code (where the PDB file is fetched from the PDB mirror from backend). NAPS offer five methods to generate the network which is as follows:

3.1.1.1 Network Construction:

Following methods can be used in order to generate a network.

3.1.1.1.1 Cα network:

A Cα atom of an amino acid residue is considered as a node and an edge is constructed if the distance between Cα-Cα is in between user-defined threshold (Default upper threshold = 7 A˚, lower threshold= 0 A˚). This method is widely used for generating the network that provides a good 3D topology of the protein structure.

3.1.1.1.2 Cβ network:

The side chain Cβ atom of amino acid is considered as a node with an edge constructed if Cβ-Cβ distance (Cα in case of Gly) between two residues is in between user-defined threshold (Default upper threshold = 7 A˚, lower threshold= 0 A˚). It is useful for understanding the 3D topology of the protein fold through side-chain interactions.

3.1.1.1.3 Any pair contact network:

The geometric center of the amino acid is considered as a node and an edge is constructed if the distance between any two atoms of the residue is in between user-defined threshold (Default upper threshold = 5 A˚, lower threshold= 0 A˚). This network provides analysis at the atomic level.

3.1.1.1.4 Centroid network:

The center of mass of an amino acid residue is used as a node and an edge is constructed if the distance between the centroids of the two residues is in between user-defined threshold (Default upper threshold = 8.5 A˚, lower threshold= 0 A˚).

3.1.1.1.5 Interaction strength network:

The geometric center of the side chain of amino acid residue is used as a node and an edge is constructed if the interaction strength between two residues, is given by,

$$\text{Iij} = \left[\frac{nij}{\sqrt{Ni * Nj}}\right] X100$$

is $\geq I_c$, threshold interaction strength (4%).

Where n_{ij} is the number of side chain atom pairs of residue $_i$ and $_j$ within 4.5 A^o, N_i and N_j are the normalization factor (33).

While generating the graph using the weighted method, the following formula is used:

$$Wij = \frac{1}{dij}$$

Where d_{ij} is the Euclidean distance between atoms for the respective parameter of i^{th} and j^{th} residue. While in case of interaction strength, the weigh is given as $W_{ij} = I_{ij}$.

3.1.1.2 Centrality analysis:

This feature plays a most important role since it identified the most central or most important or the most significant node in a network. Centrality measure of node provides a quantification of the topological importance of the node in the network (34). Different centrality measures have been proposed for ranking the nodes in a complex network and quantify their importance. NAPS provides a total of seven node based centrality measures which are shown in table 3.I.

Name	Description	Definition
Degree	Number of edges directly incident to the node.	$Cd(u) = \sum_{v \in V} Auv$ A is the adjacency matrix, V is the set of all nodes and u, v are the nodes
Closeness	The average length of the shortest path between the node and all other nodes in the graph	$Ccl(u) = \frac{(n-1)}{\sum_{v \in V} dist(u,v)}$ d(u, v): shortest path distance between nodes u and v, n: number of nodes in the network.
Betweenness	Number of times a node acts as a bridge along the shortest path between two different nodes	$Cb(u) = \sum_{s \neq u \in V} \sum_{t \neq u \in V} \sigma st\,(u)/\sigma st$ σst: shortest path between s and t, σst(u): shortest path between s and t passing through u
Clustering Coefficient	The ratio of connected neighbors of a node to the total number of connections possible between the neighbors.	$Ccc(u) = \frac{\lambda(u)}{\tau(u)}$ $\tau(u) = Cd(u)(Cd(u)-1)/2$, while $\lambda(u)$ is neighbors of u connected by an edge
Eigenvector	It assigns a relative score to all nodes in the network based on the concept that connections more to the score of the node.	$Xi = \frac{1}{\lambda} \sum_{j=1}^{N} Aij\ Xj$ A^{ij} is the ijth element of the adjacency matrix, Λ: largest eigenvalue of A, X_i: eigenvector centrality of node i
Eccentricity	The shortest path distance of the node to the farthest node in the network.	$Ce(u) = \max(dist(u,v))$ N(u) is the neighbors of u.
Strength	The weighted degree which is represented by cumulative weights of all the edges connected to a node.	$Cs(u) = \sum_{v \in N(u)} Wuv$ W_{uv}: weight of the edge joining u and v

Table 3.I: NAPS features

Once the edge-weighted graph is constructed, using the topological parameters as features, we can do various calculations in order to generate new features.

3.1.2 Node-weighted amino acid contact energy network (NACEN):

This is a standalone tool based on the package in R. NACENs are constructed based on amino acid contact energy network (AACEN) (35). It is a node-weighted amino acid network and has six different properties for the residue. The user can give the input in two ways, PDB file located on the local machine or by entering the four-digit PDB code which then connects the DSSP (36) server and then downloads the file in order to plot the graph. DSSP is an online database of secondary structure assignments for all the protein entries in Protein Data Bank (PDB). It also has pre-calculated feature files which help in generating the graph much faster.

3.1.2.1 Network Construction:

The network is created using six different features which belong to four types as node weights in the AAN.

3.1.2.1.1 Solvent accessibility:

Relative solvent accessibility is calculated using the Dictionary of protein secondary structure (DSSP) database and then normalized by the side chain surface area. This feature is important for studying and understanding the functional residues in the complex.

3.1.2.1.2 Functional residues:

Features from Amino acid index database are used, functional residues such as catalytic residues can be classified as hydrophobic or polar residues based on their physiochemical properties. In this feature, mass, hydrophobicity, and polarity of residues are obtained from the database (37).

3.1.2.1.3 Flexibility:

The flexibility of backbone residues is calculated by DynaMine (38). It quantifies the backbone flexibility on the amino-acid level. A value of 1 indicated rigid conformation of the residue, while a value of zero indicates a highly flexible residue.

3.1.2.1.4 Jensen-Shannon Divergence (JSD) score:

The conservation score of JSD is been used to estimate sequence conservation (39). The residues in alignment with more than 30% gaps are ignored when calculating the JSD score.

There are a total of six features of residues as node weight types, namely solvent accessibility (s), mass (m), hydrophobicity (h), polarity (p), flexibility (f) and JSD conservation score (j), which are used to predict the functional residues (FRs) (40, 41). Depending on the user choice of node weight, a network is generated. The node weights of residue i is defined as

$$W_i = 1\text{- } s_i, m_i, h_i, p_i, f_i \text{ or } j_i$$

The normalized score is used to generate the network.

3.1.2.2 Topological parameters:

There is a total of four parameters to reveal both the physiochemical properties and topological characters for node i in the network, which includes node weighted degree K_{wi}, betweenness B_{wi}, closeness centrality C_{wi} as given in table 3.II.

Name	Description	AACEN	NACEN
Degree	Total number of edges directly incident to the node	$Ki = \sum_{j \neq i}^{n} AMij$ AM_{ij}: Two nodes connected by edges	$K_{wi} = w_i K_i$
Closeness	The average length of the shortest path between the node and all other nodes in the graph. Hence, the more central the node is, the closer it is to all other nodes.	$Ci = \frac{1}{\sum_{j \neq i}^{n} dij}$ d_{ij}: Distance from node i and j	$C_i = wiCi$
Betweenness	Number of times a node acts as a bridge along the shortest path between two different nodes.	$Bi = \sum_{v \neq i \neq j}^{n} \partial(i)jv / \partial jv$ $\partial(i)_{jv}$: Number of shortest paths between node j and v passing through i, ∂_{jv}: Number of shortest paths from node j to v.	$B_{wi} = w_i B_i$.

Table 3.II: NACEN features

After constructing the network, the next thing is predicting the interface residues which can be done by using tools.

The total number of features available for NAPS are five network type (Cα, Cβ, atom pair contact, centroid, and interaction strength) and seven topological parameters (degree, closeness, betweenness, clustering coefficient, eigenvector, eccentricity, strength) which gives a feature set of 35, for weighted or non-weighted. While in case of NACEN, there are of six network type (solvent accessibility, mass, hydrophobicity, polarity, flexibility, and JSD conservation score) and three topological parameters (degree, closeness, betweenness), hence a total of 18 parameters, for weighted or non-weighted.

3.1.3 Interface residues:

Once the network is constructed, we need to determine places where the actual protein-protein interaction occurs. Tools for determining such interactions are very scarce. The current tools such as NACCESS (42), DSSP (43), DPX server, CX server can process only one molecular structure at a time and the user needs to run the program manually each and every time in order to get the result. In addition to this, it is important to generate the data in a user-friendly manner and which can be easily edited.

Protein Structure and Interaction Analyzer (PSAIA) (44) is one such tool which can process each chain within the given molecular structure file separately (while ignoring other chains). It is easy to use and generates output which can be easily processed for other purposes. PSAIA consists of two separate tools PSA (Protein Structure Analyzer) and PIA (Protein Interaction Analyzer). User needs to upload PDB file in order to determine the interaction, while the output is generated in the text (table) and XML format. Here we've used Interaction analyzer only and we'll be mentioning about this only.

1.3.1 Interaction Algorithm:

These are the following algorithms used in PSAIA:

1.3.1.1 Atom Nucleus Distance:

In this method, two residues from opposite chains are defined as interacting if there is at least one pair of the non-hydrogen atom, one from each residue, at a distance below the user specified threshold (Threshold value between 4.5 – 6 A^o) (45).

1.3.1.2 Atom Van der Walls Radii Distance:

Two residues from opposite chains are marked as interacting if there is at least one pair of non-hydrogen atoms, one from each residue, at a distance smaller than the sum of their van der Waals radii plus a user-defined threshold (Threshold value between 0.5 – 1.5 A^o) (46).

1.3.1.3 Accessible Surface Area (ASA) change:

The accessible surface area is the atomic surface area of a molecule that is accessible to solvent and is usually represented as A^{o2} (square Angstroms) (47). ASA is calculated using 'rolling ball' algorithm (48), which uses a sphere (representing a solvent molecule) of a user-defined radius and 'probes' the surface of the molecule (usual value 1.4 A^o).

ASA change is calculated by calculating ASA for a particular residue before and after the process of complexation. If the difference between ASA in bound and unbound form is above the user-defined threshold, then a residue is defined as an interacting residue.

Radii file consists of Van der Waals Radii for each atom of particular residue and nucleotide, ligand atoms and heteroatoms which is included in the installation package. Linking of the file with software can be done easily and then calculations can be carried out.

In output options, contact shows the residues which in contact, binding residues give a list of residues which are in contact with an amino acid of another chain while residue binding status shows a list of each and every residue irrespective of their binding status.

3.1.4 Machine learning:

The term machine learning was first coined by Arthur Samuel in the year 1959 who is a pioneer in the field of computer gaming and artificial intelligence (AI) (49). Initially, AI and ML were considered the same thing since both used the same type of approach to solve problems. A few years later, there was a need to use logic and knowledge-based approach (50). This created a rift between AI and ML, by 1980's AI became dominant since work was carried out in knowledge-based learning while ML flourished in the 1990s when it started using models based on statistics and probability theory (51). It started tackling problems of particular nature and started gathering data, which leads to us to today ML algorithms which are state-of-art techniques.

3.1.4.1 Learning methods:

Once the data is been provided, ML uses the data to train itself, forms a model and then it generates the output.

Supervised learning: In this method, examples having a predefined-solution are given as input and then test data is presented to it. During the process, it learns certain rules which can map the input and output data (52).

Unsupervised learning: The algorithm builds a mathematical model from the input data which contains only the input and no desired output labels. This method can discover patterns in the data and can group the input into categories as in feature-based learning.

Semi-supervised learning: The algorithm develops mathematical models from incomplete data where a portion of input data doesn't have a label.

Reinforcement learning: In this type of learning, data is provided as feedback to the algorithm. During the training, the environment can go into a new state which can be positive or negative for the model depending on the feedback, and by learning from these feedbacks, the model trains itself. In this way, the algorithm not only learns to get short time rewards but also gets better in less amount of time. This kind of algorithms is used in autonomous vehicles or in learning to play a game against a human opponent (53).

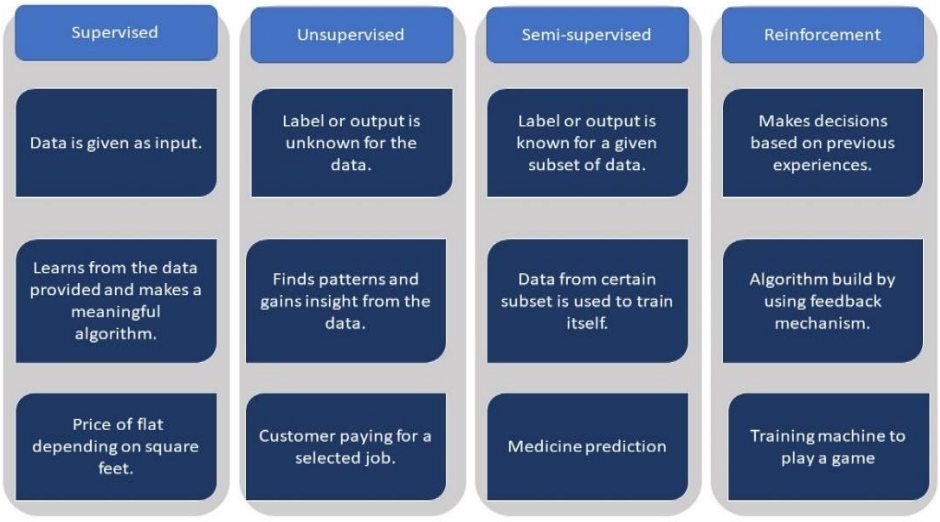

Fig 2.1: Learning methods used by ML.

Random forest: It is an ensemble learning method which operates by constructing a multitude of decision tree at training time and generates the output which is generally mean of all the individual trees. This method helps in correcting the overfitting of data which may occur in the decision tree (54).

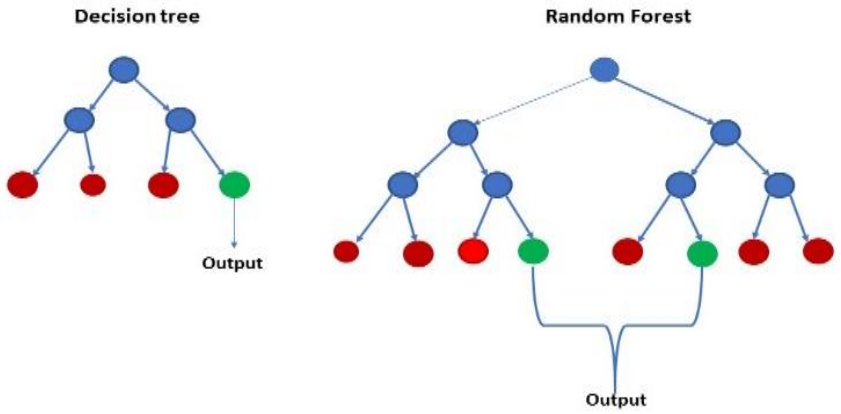

Fig 2.2 Decision tree and random forest.

3.1.4.2 Cross-Validation:

Cross-validation is a technique used to evaluate ML models by training several ML models on the subset of available input data and to evaluate them based on the complementary data. It is used to detect overfitting of the data. There are many cross-validation techniques such as k-fold-cross-validation, leave one out cross validation (LOOCV), holdout method and bootstrap method. In this study, we've used LOOCV to evaluate our model.

In LOOCV, the training is performed on the whole dataset but by leaving only one data point, which is used as test data and then iterate for each data point. Since each and every data point is been used, there is low bias (55).

3.1.5 Dataset

We have compiled a dataset of 101 protein-protein complexes for this study from the earlier reported dataset (56). The dataset contains multimeric complexes, in order to reduce the complexity, we selected only dimeric complexes. We also removed the missing residues, heterogenous atoms from the complexes so that we can concentrate specifically on interchain and intrachain residues only. The dataset includes protein-protein complexes with diverse functions (antigen-antibody, enzyme-inhibitor, enzyme-substrate, other G-protein, etc.).

These complexes were classified into two groups based on their binding affinity. The complexes with K_d value < 10^{-8} M were considered as a high-affinity class while K_d > 10^{-8} M were considered as a low-affinity class. The K_d range for the high-affinity class is generally considered as permanent protein-protein complex. Using this criterion, we obtained a balanced dataset which includes, 50 high affinities and 51 low-affinity class. For preliminary studies, we used a total of 8 complexes, the Protein Data Bank (PDB) code for these eight complexes is given in Table 3.III. While the description for all the complexes is provided in supporting information Table S1.

High binding affinity	Low binding affinity
1AVX	1BUH
1AY7	1E96
AM10	1KAC
2B42	1ZHI

Table 3.III: List of PDB code for preliminary studies.

3.1.6 Calculation:

Once the graph is generated using the topological parameters, we used the text file in case of NAPS and CSV file in case of NACEN and calculated the Z-Score for the file programmatically. Here we used python 3.6 while the text editor we used was sublime text 3. Latest python libraries were used (pandas (v: 0.23.4), numpy (v: 1.16.2), scipy (v: 1.1.0)).

$$Z - \text{Score} = \frac{X - \mu}{\sigma}$$

In this equation, X is the actual value, μ is the mean and σ is the standard deviation.

Once the calculations were made, the output was a CSV file which contained the values of the parameter as well as the Z-score values in separate columns, so that user can use whichever value seems beneficial. During the same process, we added the PSAIA data in the new column so that we can know which residues are interacting.

3.1.7 Assessment of the performance:

As mentioned earlier, we used leave one out cross-validation (LOOCV) method to evaluate the performance of the model. The prediction performance is assessed using the following measures:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP+FN}}$$

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

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

$$\text{F} - \text{measure} = \frac{2\,(\,\text{Sensitivity x Precision})}{\text{Sensitivity+Precision}}$$

In the above equations, TP, TN, FP, FN represent, true positive, true negative, false positive, false negative respectively.

Chapter 4: Results and Discussion

4.1 Feature selection:

We tried various combination of features from NAPS and NACEN to obtain the best feature for discriminating the protein-protein complexes with high and low affinity. We selected 4 protein-protein complexes at random which are shown in table 3 and analyzed each and every parameter possible for these complexes. Parameters such as the seven topological parameters and their weighted and non-weighted feature. To reduce the complexity, we used only interacting residues which we obtained by using PSAIA. We found that the weighted graph gives a much better graph as compared to the non-weighted graph.

In the next step, we tried optimizing the threshold value for the maximum distance contact criterion in PSAIA. We selected different threshold values ranging from 4-9 A^o since in most of the studies these values have been used. Hence, we plotted a graph for all the range and found that 7 A^o gives the best result since most of the interface residues were visible in that range. We also made a comparison between different topological parameter to understand which parameter is showing more importance to differentiate between high and low binding affinity complexes.

For NAPS, which has a feature set of 32, we calculated the Z-score for each and every complex. In order to simplify the data, we selected the data in a way such that Z-score values above 2, 1.5 and 1 were only selected since most the hot patches or hotspots must occur in the top values only. The total number of values in the selected data was calculated and then divided by the total number of interface residues, to get the normalized number. We then divided value greater than 2 with values greater than 1. Since there was a pattern recognized in this method, we carried out the same procedure with the remaining parameters. Similarly, for NACEN, for a feature set of 18, we calculated the Z-score and then the same procedure was followed.

In order to find the effect of these topological parameters on the amino acid level, we selected amino acids for the same values, greater than 2, 1.5 and 1 and carried out a similar procedure. We found that aromatic and positively charged residues at the binding sites are identified as an important parameter for discriminating protein-protein complexes based on binding affinities. This observation was similar to previously reported data (57, 58). This finding specifically emphasizes the importance of aromatic and positively charged residues in the binding site.

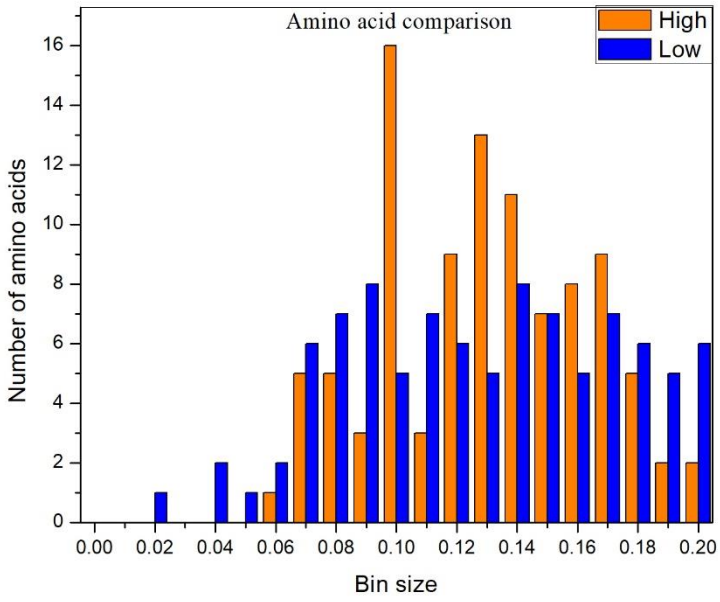

Fig 4.1: Comparison between aromatic and positively charged residues for high and low binding affinity complexes.

We selected this feature to differentiate between high and low binding affinity complexes. After seeing a trend in this feature, we considered taking only the interface residues and dividing it with the whole length of the complex. This will not only give us the exact number of interface residues but it'll also help us to build us a new feature. We tried finding a correlation between the various features but we couldn't find any direct correlation. Every-time we across a graph where the high binding and low binding affinity complexes were showing similar importance it became clear that each and every feature has its own importance and no single feature or combination of feature can be used to differentiate between the binding affinity of the complexes.

4.2 Analysis of selected features:

We started using machine learning to train out data and simultaneously checked its performance using the cross-validation LOOCV method. Here we saw the feature importance, which feature is playing a crucial role in differentiating the complexes and also calculated the true positive (TP), true negative (TN), false negative (FN), false positive (FP) values for the data so that we can determine the performance of the data. We tried using different features in order to increase the overall performance of the model. Performance for the features is given in table 3.

Features	TP	FN	FP	TN	Sensitivity	Precision	Specificity	F- measure
Amino acid	35	15	20	31	70	63.6	60.8	66.7
Interface residue	29	21	20	31	58	59.2	60.8	58.6
Amino acid + Interface residue	37	13	15	36	74	71.2	70.6	72.5
Topology parameters + Amino acid + interface	35	15	12	39	70	74.5	76.5	72.2
Strength + Amino acid + Interface residue	32	18	13	38	64	71.1	74.5	67.4
ANN + Strength + Amino acid + Interface residue	33	17	13	38	66	71.7	74.5	68.8
ANN + Amino acid + Interface residue	34	16	15	36	68	69.4	70.6	68.7
Topological parameters	35	15	18	33	70	66.0	64.7	68.0

Table 4.I: Comparison between different features of NAPS and their analysis.

On comparing different features, we can see that by adding some features such as interface residues to amino acids, the performance is increasing and the machine can differentiate between the data much more effectively as compared to only amino acids. When we add the topological parameters, the performance decreases since randomness in the data increases and hence, the overall performance decreases.

The feature importance parameter showed that two features played an important role in differentiation, strength, and ANN. When we tried adding both the parameters, the performance almost remains the same which clearly indicates that differentiation cannot be made simply by taking into consideration one or two features. On selecting only the topological parameters feature, it can be seen that the result is better and it can be said that adding additional features such as amino acids and interface residues don’t make much of difference for our dataset.

We carried out the same exercise using the NACEN generated topological parameters. The interface residue feature will remain the same since the complexes are the same while the amino acid feature will change since the process of generating graph is different.

Features	TP	FN	FP	TN	Sensitivity	Precision	Specificity	F-measure
Amino acid	31	19	14	37	62	68.9	72.5	65.3
Amino acid + Interface residue	36	14	9	42	72	80.0	82.4	75.8
Topology parameters + Amino acid + interface	36	14	7	44	72	83.7	86.3	77.4
Topological parameters	37	13	7	44	74	84.1	86.3	78.7

Table 4.II: Comparison between different features of NACEN and their analysis

In this method, there is a substantial increase in the performance as it can be seen from the F-measure. Amino acids did not play a substantial role in differentiation but the topological parameters generated during the construction of graph plays an important role in differentiating high binding and low binding affinity complexes. Interface residues when added to the amino acid feature tends to increase the overall performance but the most crucial role is played by the topological parameters. We did not consider any other individual feature from the topological parameters since each and every parameter played an equally important role. It is important to note that the performance of amino acid and interface has increased after the addition of topological parameters but the overall performance of topological parameters is higher by a little margin. Hence, calculating only topological parameters is sufficient and there is no need to consider other features for our dataset.

Chapter 5:

Conclusion and future perspective

We used two different methods of generating the network and used topological parameters for comparison. We also used amino acids feature in order to understand whether a trend exists or not and on comparison and found that it does play a role in differentiating the two binding affinity complexes. We also used a unique feature by just comparing the interface residues from the whole chain and found that the total number of interfaces in high binding affinity complexes is more as compared to low binding affinity complexes. Hence, this feature was also used for comparison. Later we used a machine learning tool in order to generate feature importance and finally understanding which features from the graph can be used to make the differentiation.

From this study, we concluded that for our dataset, NACEN generated graph (edge) is a better option as compared to NAPS (node). We also tried new features which showed improvement in the overall performance and hence, these features can be used in further studies.

Future trends:

After analysis of the topological parameters, we selected the five residues and located them on the structure. We saw that these residues formed a cluster and the rend was seen in mostly each and every complex. The NAPS showed cluster formation whereas NACEN did not show any such trend. Were curious to understand why these residues form a cluster and do they indicate a hotspot patch.

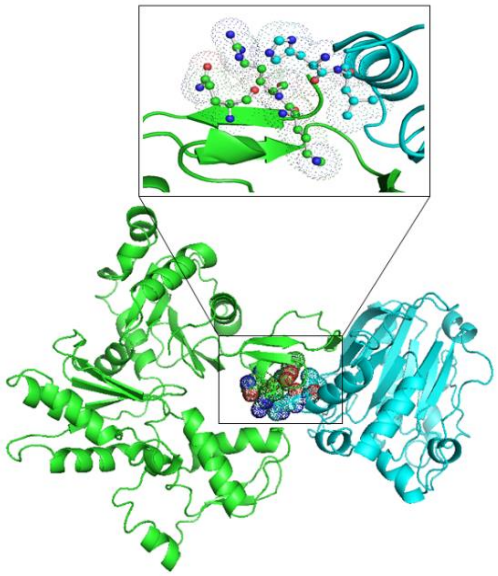

Fig 5.1: Clustered residues generated using NAPS topological parameters. (PDB: 1ATN_AD)

We would also like to check the mobility of each and every complex since the hotspots are known to be rigid and hence are less mobile while residues of intrachain are more mobile. This can give us an insight where are the above-mentioned residues placed and are they near the hotspot. In addition to this, we are also planning to have a look at the uniport entries of the complexes and check if there are any reports about the mutation at the specific amino acid and what is the effect due to this on the entire structure. So that we can prove that the specific residue in a certain complex plays an important role.

Chapter 6: References

1) Biggs, N., Lloyd, E. K., & Wilson, R. J. (1986). Graph Theory, Oxford University Press, 1736-1936.
2) Hooper, David; Whyld, Kenneth (1996) [First pub. 1992]. "knight's tour". The Oxford Companion to Chess (2nd ed.). Oxford University Press. p. 204.
3) Sylvester, J. J. (1878). Chemistry and algebra, Nature, 284
4) Tutte, W.T. (2001), Graph Theory, Cambridge University Press, p. 30
5) Gardner, M. (1992). Fractal music, hypercards and more--. WH Freeman, P. 203
6) Habibi, I., Emamian, E. S., & Abdi, A. (2014). Quantitative analysis of intracellular communication and signaling errors in signaling networks. BMC systems biology, 8(1), 89.
7) Habibi, I., Emamian, E. S., & Abdi, A. (2014). Advanced fault diagnosis methods in molecular networks. PloS one, 9(10), e108830.
8) Jailkhani, N., Ravichandran, S., Hegde, S. R., Siddiqui, Z., Mande, S. C., & Rao, K. V. (2011). Delineation of key regulatory elements identifies points of vulnerability in the mitogen-activated signaling network. Genome research, 21(12), 2067-2081.
9) Vishveshwara, S., Brinda, K. V., & Kannan, N. (2002). Protein structure: insights from graph theory. Journal of Theoretical and Computational Chemistry, 1(01), 187-211.
10) Csermely, P., Korcsmáros, T., Kiss, H. J., London, G., & Nussinov, R. (2013). Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. Pharmacology & therapeutics, 138(3), 333-408.
11) Song, J., Li, F., Takemoto, K., Haffari, G., Akutsu, T., Chou, K. C., & Webb, G. I. (2018). PREvaIL, an integrative approach for inferring catalytic residues using sequence, structural, and network features in a machine-learning framework. Journal of theoretical biology, 443, 125-137.
12) Yan, W., Zhou, J., Sun, M., Chen, J., Hu, G., & Shen, B. (2014). The construction of an amino acid network for understanding protein structure and function. *Amino acids*, *46*(6), 1419-1439.
13) Aftabuddin, M., & Kundu, S. (2007). Hydrophobic, hydrophilic, and charged amino acid networks within protein. Biophysical journal, 93(1), 225-231.
14) Karain, W. I., & Qaraeen, N. I. (2015). Weighted protein residue networks based on joint recurrences between residues. BMC bioinformatics, 16(1), 173.

15) Fokas, A. S., Cole, D. J., Ahnert, S. E., & Chin, A. W. (2016). Residue geometry networks: A rigidity-based approach to the amino acid network and evolutionary rate analysis. Scientific reports, 6, 33213.
16) Piovesan, D., Minervini, G., & Tosatto, S. C. (2016). The RING 2.0 web server for high quality residue interaction networks. Nucleic acids research, 44(W1), W367-W374.
17) Bhattacharyya, M., Bhat, C. R., & Vishveshwara, S. (2013). An automated approach to network features of protein structure ensembles. Protein Science, 22(10), 1399-1416.
18) Wiedermann, M., Donges, J. F., Heitzig, J., & Kurths, J. (2013). Node-weighted interacting network measures improve the representation of real-world complex systems. *EPL* (Europhysics Letters), 102(2), 28007.
19) Phizicky, E. M., & Fields, S. (1995). Protein-protein interactions: methods for detection and analysis. Microbiol. Mol. Biol. Rev., 59(1), 94-123.
20) Szklarczyk, D., Franceschini, A., Kuhn, M., Simonovic, M., Roth, A., Minguez, P., & Jensen, L. J. (2010). The STRING database in 2011: functional interaction networks of proteins, globally integrated and scored. Nucleic acids research, 39(suppl_1), D561-D568.
21) Stark, C., Breitkreutz, B. J., Reguly, T., Boucher, L., Breitkreutz, A., & Tyers, M. (2006). BioGRID: a general repository for interaction datasets. Nucleic acids research, 34(suppl_1), D535-D539.
22) Salwinski, L., Miller, C. S., Smith, A. J., Pettit, F. K., Bowie, J. U., & Eisenberg, D. (2004). The database of interacting proteins: 2004 update. Nucleic acids research, 32(suppl_1), D449-D451.
23) Nepomnyachiy, S., Ben-Tal, N., & Kolodny, R (2015). CyToStruct: augmenting the network visualization of cytoscape with the power of molecular viewers. Structure, 23(5), 941-948.
24) Doncheva, N. T., Klein, K., Domingues, F. S., & Albrecht, M. (2011). Analyzing and visualizing residue networks of protein structures. Trends in biochemical sciences, 36(4), 179-182.
25) Shannon, P., Markiel, A., Ozier, O., Baliga, N. S., Wang, J. T., Ramage, D., & Ideker, T. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. Genome research, 13(11), 2498-2504.

26) Pasi, M., Tiberti, M., Arrigoni, A., & Papaleo, E. (2012). xPyder: a PyMOL plugin to analyze coupled residues and their networks in protein structures. Journal of chemical information and modeling, 52(7), 1865-1874.
27) Tiberti, M., Invernizzi, G., Lambrughi, M., Inbar, Y., Schreiber, G., & Papaleo, E. (2014). PyInteraph: a framework for the analysis of interaction networks in structural ensembles of proteins. Journal of chemical information and modeling, 54(5), 1537-1551.
28) Yan, W., Hu, G., Liang, Z., Zhou, J., Yang, Y., Chen, J., & Shen, B. (2018). Node-weighted amino acid network strategy for characterization and identification of protein functional residues. Journal of chemical information and modeling, 58(9), 2024-2032.
29) Chakrabarty, B., & Parekh, N. (2016). NAPS: Network analysis of protein structures. Nucleic acids research, 44(W1), W375-W382.
30) Brinda, K. V., & Vishveshwara, S. (2005). A network representation of protein structures: implications for protein stability. Biophysical journal, 89(6), 4159-4170.
31) Böde, C., Kovács, I. A., Szalay, M. S., Palotai, R., Korcsmáros, T., & Csermely, P. (2007). Network analysis of protein dynamics. Febs Letters, 581(15), 2776-2782.
32) Amitai, G., Shemesh, A., Sitbon, E., Shklar, M., Netanely, D., Venger, I., & Pietrokovski, S. (2004). Network analysis of protein structures identifies functional residues. Journal of molecular biology, 344(4), 1135-1146.
33) Kannan, N., & Vishveshwara, S. (1999). Identification of side-chain clusters in protein structures by a graph spectral method. Journal of molecular biology, 292(2), 441-464.
34) Thibert, B., Bredesen, D. E., & del Rio, G. (2005). Improved prediction of critical residues for protein function based on network and phylogenetic analyses. BMC bioinformatics, 6(1), 213.
35) Yan, W., Sun, M., Hu, G., Zhou, J., Zhang, W., Chen, J., & Shen, B. (2014). Amino acid contact energy networks impact protein structure and evolution. Journal of theoretical biology, 355, 95-104.
36) Kabsch, W., & Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers, 22(12), 2577-2637.
37) Kawashima, S., & Kanehisa, M. (2000). AAindex: amino acid index database. Nucleic acids research, 28(1), 374-374.

38) Cilia, E., Pancsa, R., Tompa, P., Lenaerts, T., & Vranken, W. F. (2013). From protein sequence to dynamics and disorder with DynaMine. Nature communications, 4, 2741.
39) Capra, J. A., & Singh, M. (2007). Predicting functionally important residues from sequence conservation. Bioinformatics, 23(15), 1875-1882.
40) Greener, J. G., & Sternberg, M. J. (2015). AlloPred: prediction of allosteric pockets on proteins using normal mode perturbation analysis. BMC bioinformatics, 16(1), 335.
41) Huang, W., Lu, S., Huang, Z., Liu, X., Mou, L., Luo, Y., & Zhang, J. (2013). Allosite: a method for predicting allosteric sites. Bioinformatics, 29(18), 2357-2359.
42) Hubbard, S. J., & Thornton, J. M. (1993). Naccess. Computer Program, Department of Biochemistry and Molecular Biology, University College London, 2(1).
43) Kabsch, W., & Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. Biopolymers, 22(12), 2577-2637.
44) Mihel, J., Šikić, M., Tomić, S., Jeren, B., & Vlahoviček, K. (2008). PSAIA–protein structure and interaction analyzer. BMC structural biology, 8(1), 21.
45) Ofran, Y., & Rost, B. (2003). Predicted protein–protein interaction sites from local sequence information. FEBS letters, 544(1-3), 236-239.
46) Aytuna, A. S., Gursoy, A., & Keskin, O. (2005). Prediction of protein–protein interactions by combining structure and sequence conservation in protein interfaces. Bioinformatics, 21(12), 2850-2855.
47) Jones, S., & Thornton, J. M. (1997). Analysis of protein-protein interaction sites using surface patches. Journal of molecular biology, 272(1), 121-132.
48) Shrake, A., & Rupley, J. A. (1973). Environment and exposure to solvent of protein atoms. Lysozyme and insulin. Journal of molecular biology, 79(2), 351-371.
49) Samuel, A. L. (1988). Some Studies in Machine Learning Using the Game of Checkers. II—Recent Progress. In Computer Games I (pp. 366-400). Springer, New York, NY.
50) Russell, Stuart; Norvig, Peter (2003) [1995]. Artificial Intelligence: A Modern Approach (2nd ed.)
51) Langley, P. (2011). The changing science of machine learning. Machine Learning, 82(3), 275-279.

52) Russell, S. J., & Norvig, P. (2010). Artificial Intelligence-A Modern Approach (3. internat. ed.) Pearson Education, p.
53) Bishop, C. M. (2006). Pattern recognition and machine learning. springer.
54) Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning (Vol. 1, No. 10). New York: Springer series in statistics.
55) Molinaro, A. M., Simon, R., & Pfeiffer, R. M. (2005). Prediction error estimation: a comparison of resampling methods. Bioinformatics, 21(15), 3301-3307.
56) Yugandhar, K., & Gromiha, M. M. (2014). Feature selection and classification of protein-protein complexes based on their binding affinities using machine learning approaches. Proteins: Structure, Function, and Bioinformatics, 82(9), 2088-2096.
57) Gromiha, M. M., Selvaraj, S., Jayaram, B., & Fukui, K. (2010, August). Identification and analysis of binding site residues in protein complexes: Energy-based approach. In International Conference on Intelligent Computing (pp. 626-633). Springer, Berlin, Heidelberg.
58) Gromiha, M. M., Saranya, N., Selvaraj, S., Jayaram, B., & Fukui, K. (2011, December). Sequence and structural features of binding site residues in protein-protein complexes: comparison with protein-nucleic acid complexes. In Proteome science (Vol. 9, No. 1, p. S13). BioMed Central.

NACEN (http://sysbio.suda.edu.cn/NACEN/index.html)

NAPS (http://bioinf.iiit.ac.in/NAPS/index.php)

DPX server (http://hydra.icgeb.trieste.it/dpx/)

CX server (http://hydra.icgeb.trieste.it/cx/)

Appendix

Complex (High binding affinity)	Category	Complex (High binding affinity)	Category
1ATN_AD	OX	1TEC_EI	EI
1AVW_AB	EI	1TPA_EI	EI
1AVX_AB	EI	1UUG_AB	EI
1AY7_AB	EI	1YVB_AI	EI
1BKD_RS	OG	1ZLI_AB	EI
1BRS_AD	OX	2B42_AB	EI
1BVN_PT	EI	2GOX_AB	EI
1CGI_EI	EI	2HRK_AB	OX
1CSE_EI	EI	2I25_NL	AB
1DFJ_EI	EI	2J0T_AD	EI
1EAW_AB	EI	2O3B_AB	EI
1EMV_AB	OX	2OUL_AB	EI
1FLE_EI	EI	2PTC_EI	EI
1FSS_AB	OX	2SEC_EI	EI
1GPW_AB	OX	2SIC_EI	EI
1GXD_AC	EI	2SNI_EI	EI
1JIW_PI	EI	2UUY_AB	EI
1JTG_AB	EI	2VDB_AB	OX
1KXP_AD	OX	3SGB_EI	OX
1M10_AB	ER	4SGB_EI	EI
1MAH_AF	EI	4TPI_ZI	EI
1OC0_AB	EI	7CEI_AB	EI
1OPH_AB	EI	ER: Complexes with regulatory chain	
1PXV_AC	EI	EI: Enzyme-Inhibitor	
1R0R_EI	EI	ES: Enzyme-Substrate	
1RRP_AB	OX	A: Antigen-Antibody	
1STF_EI	EI	AB: Antigen-Bound Antibody	
1T6B_XY	OR	OG: Other G-Protein	
		OX: Other Miscellaneous	

Complex (Low binding affinity)	Category	Complex (Low binding affinity)	Category
1A0O_AB	OX	1SBB_AB	OR
1AK4_AD	OX	1TMQ_AB	EI
1B6C_AB	OX	1US7_AB	ER
1BUH_AB	EI	1WQ1_RG	OG
1E6E_AB	ES	1XD3_AB	OX
1E96_AB	OG	1XQS_AC	OX
1EWY_AC	ES	1YCS_AB	OX
1F6M_AC	ES	1Z0K_AB	OG
1FC2_CD	OX	1ZM4_AB	ES
1FFW_AB	OX	2A9K_AB	ES
1FQJ_AB	OG	2AJF_AE	OR
1GCQ_BC	OX	2AQ3_AB	OX
1GHQ_AB	OR	2BF_AP	OX
1GLA_FG	ER	2C0I_AB	OX
1GRN_AB	OG	2FJU_AB	OG
1GUA_AB	OX	2HLE_AB	OR
1H1V_AG	OX	2OOB_AB	ES
1HE1_AC	OX	2PCB_AB	OX
1HE8_AB	OG	2PCC_AB	ES
1KAC_AB	OR	2TGP_ZI	EI
1KTZ_AB	OR	2WPT_AB	OX
1LFD_AB	OG	3BZD_AB	OX
1MEL_BM	A	3CPH_GA	OG
1MQ8_AB	OX	EI: Enzyme-Inhibitor	
1PVH_AB	OR	ES: Enzyme-Substrate	
1QA9_AB	OX	A: Antigen-Antibody	
1R8S_AE	OG	ER: Complexes regulatory chain	
1S1Q_AB	OX	OG: Other G-Protein	
		OX: Other Miscellaneous	

System requirements and installation

1) NACEN:
   Requirements:
   The main requirement is to have a working installation of R >= 3.2.0. It also requires dependencies such as bio3d and igraph. DSSP 3.0 is also required which is included in the installation package or can be downloaded separately. The installation method is given in the supporting information.
2) NAPS:
   The main advantage over here is that no additional plugins are required. The network visualization is performed by WebGL which is preinstalled in al modern browsers.
   Firefox Mozilla: Version 4.0 and above.
   Google Chrome: Version 9.0 and above.
   Safari: Version 6.0 and newer versions on OS X Mountain Lion, Mac OS X Lion and Safari 5.1 on Mac OS X Snow Leopard
3) PSAIA
   The software can be downloaded from the link and requires no additional plugins.

Examples

1) One can follow the steps given for both tools to generate a graph. Following is the network generated from the tools. Here we're using 1ATN as an example.

Node	Degree	Cluster_ coeff	Closeness	Betweenness	Eigenvecto r centrality	Eccentricity	Average neighbor degree	Strength
A1	6	1.000000	4.550683	0.000076	0.000054	10	12.38645	251.0000
A2	9	0.833333	4.584678	0.000000	0.000062	10	12.30909	220.0000
A3	17	0.477941	4.966116	0.001884	0.000151	10	14.52124	353.0000
A4	12	0.636364	4.716793	0.000457	0.000078	9	13.49342	304.0000
A5	13	0.628205	4.898657	0.001432	0.000112	9	14.75919	299.0000

Table SII: NAPS generated graph:

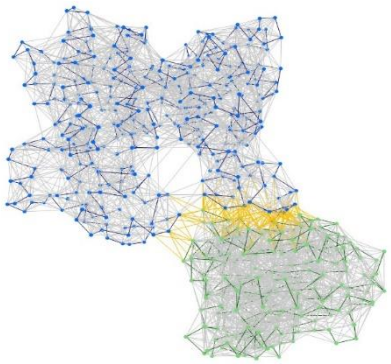

Fig S1: NAPS generated network.

Table S III: NACEN generated graph:

	ID	chain	Resid	Res	K	B	C	Kw	Bw	Cw
A:1:ASP	A:1:ASP	A	1	ASP	1	0	0.000105	0	0	0.100088
A:2:GLU	A:2:GLU	A	2	GLU	2	627	0.000112	0.033974	0.004533	0.107823
A:3:ASP	A:3:ASP	A	3	ASP	2	1252	0.00012	0.033409	0.008901	0.14077
A:4:GLU	A:4:GLU	A	4	GLU	2	1875	0.00013	0.053668	0.021415	0.29079
A:5:THR	A:5:THR	A	5	THR	2	2496	0.000141	0.042913	0.022795	0.293112

We also obtain the predicted protein-protein interaction from PSAIA.

Chain	Residue no.	Residue	Contact	Output type
A	1	ASP	0	MaximunDistance
A	2	GLU	0	MaximunDistance
A	3	ASP	0	MaximunDistance
A	4	GLU	0	MaximunDistance
A	5	THR	0	MaximunDistance

2) Once all the data is obtained, we use a python program to convert the graph values into Z-score and also concatenate the PSAIA result in the same CSV file. During this step, we consider only the interface residues.

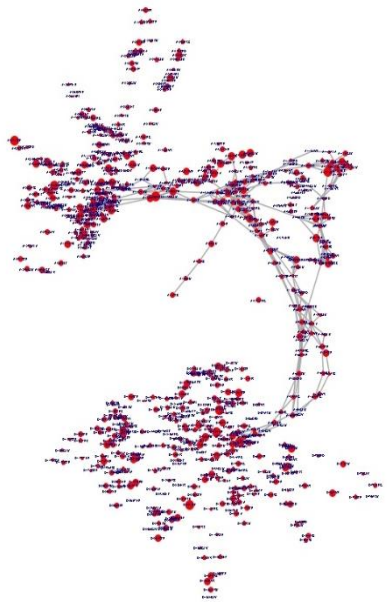

Fig SII: NACEN generated network

	Degree	Cluster_coeff	Closeness	Betweenness	Eigenvector centrality	Eccentricity	Average neighbor degree	Strength
61	21	0.447619	7.813632	0.102099	0.013915	7	20.11033	571
414	19	0.497076	7.698478	0.168419	0.020464	7	20.44935	543
415	24	0.427536	7.580601	0.095816	0.030927	8	21.502	500
60	22	0.458874	7.570512	0.112171	0.017244	8	21.58058	515
58	20	0.484211	7.526447	0.034895	0.007657	7	19.71678	459

Degree_ zscore	Cluster_coeff_ zscore	Closeness_ zscore	Betweenness_ zscore	Eigenvector centrality_ zscore	Eccentricity_ zscore	Average neighbor degree_ zscore	Strength_ zscore
0.459561	-0.64015	2.872291	3.967458	-0.25	-1.96144	0.208009	1.06037
0.08293	-0.21414	2.719437	6.88946	-0.05228	-1.96144	0.324221	0.867438
1.024506	-0.81314	2.562968	3.690634	0.263595	-1.16701	0.685053	0.57115
0.647876	-0.5432	2.549576	4.411221	-0.1495	-1.16701	0.711991	0.674506
0.271246	-0.32496	2.491084	1.006506	-0.43893	-1.96144	0.073103	0.288642

Chain	Residue No.	Residue	Interaction
A	62	ARG	1
D	44	HIS	1
D	45	LEU	1
A	61	LYS	1
A	59	GLN	1

Select the feature set and train the data to calculate feature importance (calculation done on multiple files).

[0.03056898, 0.01111046, 0.16024753, 0.01566347, 0.02120994, 0.02303171, 0.02493199, 0.00765866]

4) We can also calculate TP, FP, FN, TN and calculate the performance of the model as discussed earlier.