University of Zagreb

Faculty of Science

Department of Biology

Vedran Franke

PREDICTION OF PROTEIN-PROTEIN INTERACTIONS FROM PRIMARY STRUCTURE USING A RANDOM FOREST CLASSIFIER

Graduation Thesis

Zagreb, 2010.

Ovaj rad izrađen je u Grupi za bioinformatiku, Zavod za molekularnu biologiju, Biološki odsjek, PMF, pod vodstvom prof. dr. sc Kristiana Vlahovičeka te je predan na ocjenu Biološkom odsjeku Prirodoslovno-matematičkog fakulteta Sveučilišta u Zagrebu radi stjecanja zvanja dipl. ing .biologije, smjer molekularna biologija.

BASIC DOCUMENTATION CARD

University of Zagreb
Faculty of Science
Department of Biology

Graduation Thesis

PREDICTION OF PROTEIN-PROTEIN INTERACTIONS FROM PRIMARY STRUCTURE USING A RANDOM FOREST CLASSIFIER

Vedran Franke

Bioinformatics Group, Department of Molecular Biology, Division of Biology, Faculty of Science, University of Zagreb

ABSTRACT

The interaction between proteins is fundamental to a broad spectrum of biological functions, including regulation of metabolic pathways, immunological recognition, DNA replication, progression through the cell cycle, and protein synthesis. Due to the growing disparity between the amount of sequenced genomic content and functional data, there exist a pressing need for tools and methods that will enable prediction of phenotypic traits, on the molecular or organism level, based on the sequence alone. In this work we have constructed a high quality dataset of protein structures that has enabled us to use the Random Forest non-linear classificator to develop a method for prediction of interacting residues from the protein primary structure. Our results have shown that, although the Random Forest algorithm has a unique capability of accurately classifying highly dimensional data, we still have an incomplete knowledge of structural factors that determine the specificity of protein-protein interactions, thus putting an upper limit the on the usefulness of the machine learning approach in predicting protein interactions on the level of single amino-acids.

(32 pages, 11 figures, 2 tables , 27 references, original in: English)

Thesis is deposited in the Central biological library.

Key words: protein interactions, random forest, machine learning, prediction

Supervisor: Dr. sc. Kristian Vlahoviček, Assoc. Prof.

Reviewers: Dr. sc. Kristian Vlahoviček, Assoc. Prof.

Dr. sc. Željka Vidaković-Cifrek, Asst. Prof.

Dr. sc. Ita Gruić-Sovulj, Asst. Prof.

Thesis accepted: October, 2010.

TEMELJNA DOKUMENTACIJSKA KARTICA

Sveučilište u Zagrebu
Prirodoslovno-matematički fakultet
Biološki odsjek Diplomski rad

PREDVIĐANJE PROTEIN - PROTEIN INTERAKCIJA IZ PRIMARNE STRUKTURE PUTEM "RANDOM FOREST" KLASIFIKATORA

Vedran Franke

Grupa za bioinformatiku, Zavod za molekularnu biologiju, Biološki odsjek, Prirodoslovno-matematički fakultet, Sveučilište u Zagrebu

SAŽETAK

Međusobne interakcije između proteina temelj su niza bioloških procesa, od regulacije metaboličkih puteva, specifičnosti imunoloških reakcija, replikacije DNK do sinteze proteina. Nagli razvoj visoko-protočnih metoda doveo je do velikog povećanja produkcije bioloških sekvenci, stvorivši potrebu za razvojem metoda i alata za njihovu funkcijsku analizu, te predviđanje fenotipskih svojstava, kako na molekularnoj, tako i na razini cijelog organizma. U ovom radu smo agregirali strukturalne podatke iz postojećih baza podataka, čime smo dobili skup proteinskih kvaternih struktura visoke kvalitete koji nam je omogućio primjenu metoda strojnog učenja za predviđanje interakcija između proteina. Iskoristili smo „Random Forest“ algoritam za predviđanje interakcijskih aminokiselina iz primarnih struktura proteina. Pokazali smo da, iako „Random Forest“ alogritam ima mogućnost klasifikacije visokodimenzionalnih podataka s izuzetnom točnošću, trenutno znanje o strukturalnim faktorima koji utječu na specifičnost interakcija između proteina nije na razini koja bi omogučila predviđanje interakcija na razlučivosti pojedinih aminokiselina koristeći isključivo sekvence proteina.

(32 stranica, 11 slika, 2 tablice, 27 literaturnih navoda, jezik izvornika: engleski)

Rad je pohranjen u Središnjoj biološkoj knjižnici

Ključne riječi: interakcije proteina, strojno učenje, random forest, predikcija

Voditelj: Dr. sc. Kristian Vlahoviček, izv. prof.

Ocjenitelji: Dr. sc. Kristian Vlahoviček, izv. prof.
Dr. sc. Željka Vidaković-Cifrek, doc.
Dr. sc. Ita Gruić-Sovulj, doc.

Rad prihvaćen: listopad,2010.

1 TABLE OF CONTENTS

2 ABBREVIATIONS

AA – amino acid

ASA - accessible surface area

ASU – asymmetric unit

AUC – area under the curve

DPX – depth protrusion index

RASA - relative accessible surface area

PDB - protein data bank

PQS - protein quaternary structure server

ROC - receiver operating characteristic

3 INTRODUCTION

The interaction between proteins is fundamental to a broad spectrum of biological functions, including regulation of metabolic pathways, immunological recognition, DNA replication, progression through the cell cycle, and protein synthesis (Bruce Alberts 2002). Whether or not two proteins will bind to form a stable complex that is prerequisite to biological function is dependent on the three-dimensional conformations of the proteins (Jones 1996). For a given conformation, the chemical reactivity of an individual protein is defined by the type and spatial orientation of surface accessible amino acids. Conformation therefore determines protein–protein binding. Since the experiments done by Anfinsen, there has been a virtually axiomatic view in biology - that 'sequence specifies conformation', suggesting an intriguing postulate, that the knowledge of the amino acid sequence alone might be sufficient to estimate the propensity for two proteins to interact and effect useful biological function (Bock and Gough 2001).

Interactions between proteins play a crucial part in cellular function and form the backbone of almost all biochemical processes. Large-scale experiments on whole genomes have contributed huge amounts of data enabling the identification of many interacting protein pairs, still lacking are the high-throughput methods with sufficient resolution to map the exact residues involved in the interactions. The only way to achieve the atomic resolution is with structural characterization of protein complexes, which is still an expensive, laborious and a time consuming process, and is particularly problematic for transient protein complexes. The in silico alternative to protein interaction prediction is protein-protein docking. Procedures use surface complementarity and electrostatics to predict structural complexes, fitting together two or more known structures or reliable 3D structural models via their interacting surfaces. Although there have been some successes and advances, the methods are hampered by the lack of complete understanding of forces involved and by conformational changes that often take place upon protein complexation.

In this work we have used the available structural data to construct a high quality dataset of interacting proteins on the level of individual amino acids, which could then be used to test the predictive performance of nonlinear classificators. We used a Random Forest algorithm to predict individual interacting amino-acid pairs from the sequence alone (without any prior evolutionary knowledge about the sequences).

3.1 PROPERTIES OF INTERACTING RESIDUES

Most of the prediction methods are based on features of amino acids found in protein interacting surfaces. Discriminative characteristics were found by comparing the interacting residues with those in the interior of the protein.

The most prominent differences are:

1. **Sequence conservation.** Interface residues are more conserved relative to non-interface surface residues (Teichmann 2002). Conservation may arise either for functional (Lichtarge, Bourne et al. 1996) or for structural reasons.
2. **Proportions of the 20 types of amino acids.** In protein– protein interfaces, hydrophobic, aromatic residues and arginine are enriched whereas other charged residues are depleted (Lo Conte, Chothia et al. 1999). There also appears to be a stronger tendency for hydrophobic residues in interfaces to cluster relative to those in non-interface surfaces (Neuvirth, Raz et al. 2004).
3. **Secondary structure**. Interfaces seem to favor β-strands while disfavoring α-helices; loops in interfaces also tend to be longer (Neuvirth, Raz et al. 2004).
4. **Solvent accessibility.** Interface residues have higher solvent accessibilities than non-interface surface residues (Jones and Thornton 1997). The latter residues do not have intermolecular interaction partners upon complex formation and will thus tend to maximize intramolecular interactions, reducing their sol vent accessibilities. Solvent accessibilities can be predicted from protein sequence; these methods are typically trained on datasets in which interface residues are grossly underrepresented, and thus will tend to under predict the solvent accessibilities of interface residues (Zhou and Qin 2007).
5. **Side-chain conformational entropy**. Interface residues appear to be less likely to sample alternative side-chain rotamers, perhaps to minimize entropic cost upon complex formation (Cole and Warwicker 2002).

3.2 CLASSIFICATION OF PROTEIN COMPLEXES

Protein-protein complexes can be divided into complexes for which the interacting proteins are always bound in the complex – obligate complexes, and complexes for which the interacting proteins can also exist in the unbound form - non-obligate (permanent/transient complexes). Protomers of

obligate protein complexes are not found as stable structures in vivo. In contrast to obligate complexes there exists a continuum of properties between permanent and transient non-obligate protein interactions that can range from dynamic equilibrium between oligomeric states to where interactions are continuously broken and formed and strong associations that require a molecular trigger to shift the oligomeric equilibrium (Nooren and Thornton 2003). The equilibrium point very much depends on the physiological conditions and the environment, thus an interaction that is mainly transient in vivo can become permanent when the conditions change. Cells have threefold control of the oligomeric state:

1. Colocalization
2. Local concentration of the protomers
3. Local physicochemical environment (including the posttranslational modifications).

It turned out that the type of the complex has considerable consequences for prediction of the interface. Obligate interfaces are larger, flatter and better conserved than transient interfaces and consist primarily of side chain-side chain contacts whereas in transient interfaces the backbone plays a more prominent role. Overall interface residue propensities are similar between obligate and transient interfaces: aromatic and hydrophobic residues and atoms are usually enriched in interfaces, whereas charged groups (except arginine) are usually depleted. However, these preferences are stronger in obligate complexes than in transient complexes . In addition to the larger interface-to-surface ratio, this causes obligate complexes to be considerably easier to predict than transient complexes. (de Vries and Bonvin 2008).

4 DATASET CREATION AND COMPUTATIONAL METHODS

4.1 PROTEIN STRUCTURE DATA – PDB, PQS and PiQSiE

Within a crystal, proteins are arranged in a regular pattern that can be imagined as a mosaic. The repeating unit of this mosaic is termed the asymmetric unit (ASU). The structure that can be downloaded by default from the Protein Data Bank (PDB) website corresponds to this ASU (Berman, Battistuz et al. 2002). Importantly, the ASU does not necessarily reflect the biological state of a protein. For example, a monomeric protein may crystallize with two or more copies of the polypeptide chain in the ASU, and the opposite may also happen, where a dimeric protein crystallizes with a single chain in the ASU. When carrying out an analysis on protein structure, it is important to use the biological state of a protein and not the ASU (Henrick and Thornton 1998). For this reason, methods were developed to predict the biological state of a protein from its crystallographic structure, like the protein quaternary structure server (PQS) . The difficulty in those methods is to determine which contacts in the crystal are biological and which are artifacts. The predictive power of criteria such as buried surface area (Henrick and Thornton 1998) has been investigated to discriminate between biological and crystallographic contacts, but none is perfect. As a result, a significant number of predictions are erroneous—for example, the error rate of PQS predictions was estimated to be about 16%.

Because the error rate of automatic prediction methods is not negligible, there is a need for large scale manual curation of structural data. PiQSi (Levy 2007) is the first database that improves the accuracy of the data by using a wiki approach to enable hands on expert assessment of protein complexes. It provides more than 10000 biological quaternary structures, and a benchmark test confirmed the high quality of the annotations.

3D Complex (Levy, Pereira-Leal et al. 2006) is a database of hierarchical classifications of protein quaternary structures. The database contains clusters of structural data based on different complexity levels, ranging from structure spatial topology to sequence similarities. The prerequisite for structural classification was an ordered and repaired set of protein quaternary structures in a format that can easily be used in an automated pipeline. Although 3D Complex is primarily a database of structural classes, users can download high quality structural information in PDB format, which makes computational analysis much much easier.

4.2 RANDOM FOREST

Random forest (Breiman 2001) is a relatively recent invention in the field of supervised machine learning algorithms. It is based on the classical CART (Classification and regression tree) algorithm, a decision tree algorithm used for nonlinear classification or regression of the input space.

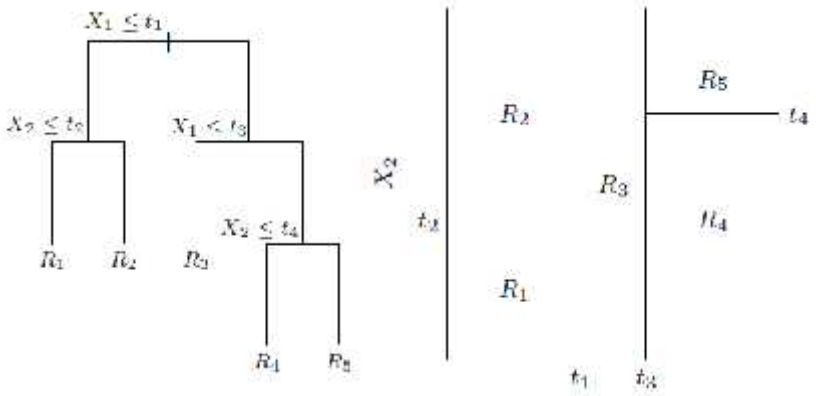

Figure 1. Left panel shows an example of a binary tree constructed using a CART algorithm . Right panel shows the splits in the input space of two variables (X1, X2) corresponding to the tree on the left.

When the input space is extraordinarily large, random subspace feature selection can potentially improve classifier diversity. Random forest classifier develops an ensemble of decision trees from randomly sampled subspaces of the input features, and final classification is obtained by combining results from trees via voting (machine learning ensemble refers to the use of multiple models to obtain better predictive performance than could be obtained by use of any of the constituent classifiers by themselves, e.g. Random forest combines multiple decision trees to get better performance than a single tree). Combining multiple trees produced in randomly selected sub spaces can improve the generalization accuracy. It is crucial to produce a large number of sufficiently different trees because the combined power of multiple heterogeneous trees reduces significantly reduces the variance of the classifier. Use of randomization in feature selection is a way to explore various possibilities of subspaces.

Random decision forest constructs many decision trees and each tree is grown from a different set of training data. To construct individual decision trees, training samples are randomly selected with replacement from the original training dataset. If the number of samples in the original training set is N, then N samples are randomly drawn with replacement. At each splitting or decision node it determines the best splitting feature from a randomly selected subspace of m features where m is much smaller than M – the total number of features. Each tree in the forest is grown to the largest

extent possible without pruning. To classify a new object, each tree in the forest gives a classification, which is interpreted as the tree voting for that class. The final classification of the object is determined by majority votes among the classes decided by the forest of trees.

We used the R implementation of Random Forest algorithm implemented in the „randomForest“ package. (Liaw and Wiener 2002)

4.3 BLASTCLUST

BLASTClust is a program, within the standalone BLAST package, used to cluster either protein or nucleotide sequences. The program begins with pair wise matches. For each pair of sequences the top-scoring alignment is evaluated upon which the algorithm decides whether to cluster the corresponding sequences. If the coverage is above a certain threshold and the score density is above a certain threshold, two sequences are considered to be neighbored. Thus determined neighbor relationship is considered symmetric and provides a base for clustering by the single-linkage method (which puts a sequence to a cluster if the sequence is a neighbor to at least one sequence in the cluster).

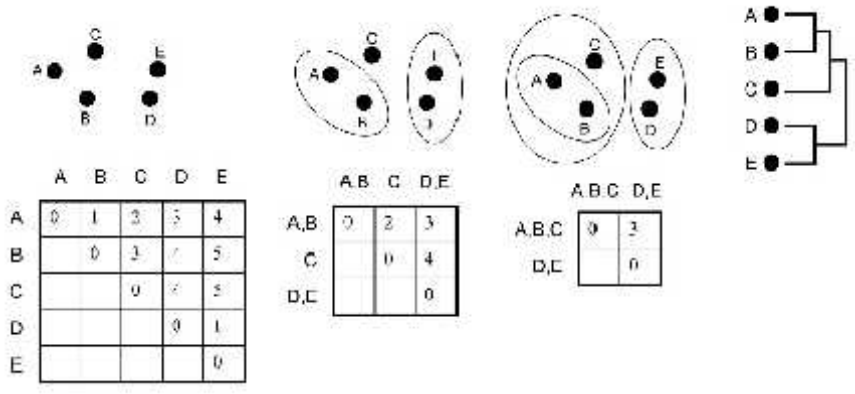

Figure 2. Example of a hierarchical clustering using single linkage algorithm. Consider five genes and the distances between them as shown in the table. In the fi rst step, genes that are close to each other are grouped together and the distances are re-calculated using the single linkage algorithm. This procedure is repeated until all genes are grouped into one cluster. This information can be represented as a tree (shown to the right), where the distance from the branch point refl ects the distance between genes or clusters. This image was adapted from (Causton 2003)

In the case of proteins, the blastp algorithm is used to compute the pair wise matches; in the case of nucleotide sequences, the Megablast algorithm is used.

BLASTClust was run as a standalone application on the UNIX operating system using the following command line:

```
blastclust -i $infile -o $outfile -p T -L 1 -b T -S 100
```

-i, -o flags specify the input file in fasta format, and the file for cluster output correspondingly

-l, -S flags tell the algorithm the minimal length (-l) over which the sequences need to have minimal percent identity (-S)

-p T specifies the chemical nature of the sequences (e.g. protein or DNA)

4.4 PSAIA – Protein Structure and Interaction Analyzer

PSAIA is a standalone application designed to simplify the use of algorithms for the analysis of sets of Protein Data Bank (PDB)files and to provide access to algorithms for interaction analysis.

It enables easy calculation of a number of structural parameters for each chain in a protein structure file, some of which are going to be explained below.

4.4.1 ASA AND RASA

The accessible surface area (ASA) is the atomic surface area of a molecule that is exposed to the solvent, and is usually expressed in $Å^2$ (square angstroms). ASA is calculated using the 'rolling ball' algorithm which uses a solvent molecule (represented as a hard sphere of a fixed particular radius), to scan the surface of the molecule. A typical value of the radius of the solvent molecule is 1.4 Å, which approximates the radius of a water molecule.

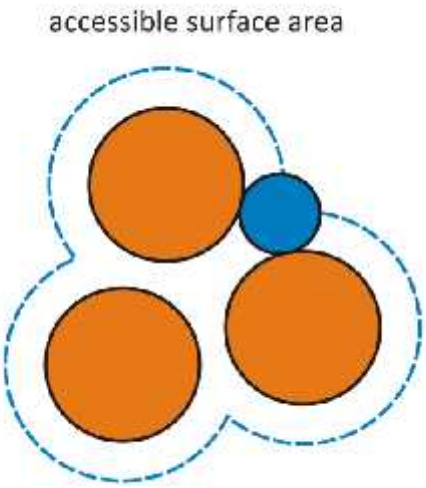

Figure 3. Graphical representation of the method for calculation of accessible surface area. ASA is calculated by rolling a solvent molecule (blue circle) of a fixed radius over the surface of constituent atoms to the molecule of interest. Dotted blue line represents the calculated solvent accessible surface.

Relative ASA (RASA) is the per-residue ratio between the calculated ASA and standard ASA for a particular residue. PSAIA calculates RASA in the following way: for each of the twenty standard AAs, one thousand triples with the corresponding residue in the middle are sampled from random PDB files. For each central residue within triplets, five ASA attributes were calculated and their mean values were determined. These mean values are taken as representative standard ASA attributes and given in a file provided within the PSAIA distribution.

The ASA and RASA can be calculated for whole quaternary complexes or for each polypeptide chain in the complex separately.

Following residue attributes are calculated for ASA and RASA:

1. Total – sum of all atom values.
2. Backbone – sum of all backbone atom values.
3. Side-chain – sum of all side-chain atom values.
4. Polar – sum of all oxygen, nitrogen and phosphorus (for nucleic acids) atom values.
5. Non-polar – sum of all carbon atom values.

4.4.2 DPX

Depth protrusion index for each residue is defined as the minimal distance (in Å)form the nearest solvent accessible atom (solvent accessibility > 0)

$$\mathbf{DPXi} = \min(\mathbf{d1}, \mathbf{d2}, \mathbf{d3}, \ldots, \mathbf{dn})$$

The depth (DPX) is thus zero for solvent accessible atoms, and > 0 for atoms buried in the protein interior, with deeply buried atoms having higher DPX values (Pintar, Carugo et al. 2003).

4.4.3 CX

CX is inverse density of the protein structure in a certain point in space. For each non-hydrogen atom in the structure, the program calculates the number of heavy atoms within a fixed distance (default value is 10 Å). The number of atoms within the sphere is multiplied by the mean atomic volume found in proteins, which then gives the volume within the sphere occupied by the protein - Vinternal. The remaining volume is calculated as the difference between the volume of the sphere and the volume occupied by the protein – Vexternal. CX value is defined as Vexternal/Vinternal.

4.5 ROC – Receiver operating characteristic

A ROC graph is a technique for visualizing, organizing and selecting classifiers based on their performance. ROC graphs have long been used in signal detection theory to depict the tradeoff between hit rates and false alarm rates of classifier. To explain the nature of the ROC graph we will first explain a two-way classification problem. Formally, each instance I (an instance is a single object of the world from which a model will be learned, or on which a model will be used, e.g. one row in the table) is mapped to one element of the set of positive and negative class labels. A classification model is a mapping from instances to predicted classes. Some classification models produce a continuous output – an estimate of instance's class membership probability to which different threshold must be applied to predict the class membership.

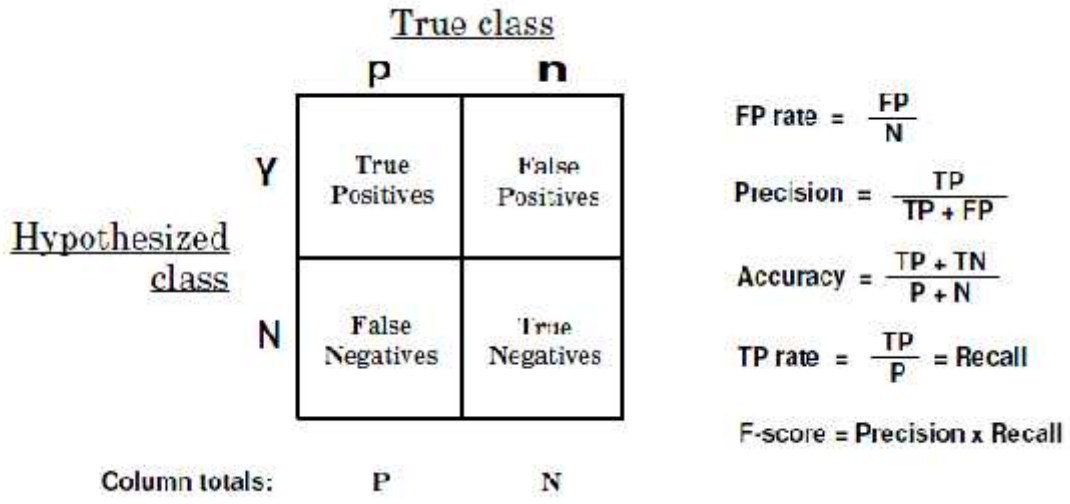

Figure 4. shows a confusion matrix and equations of several common metrics that can be calculated from it.

Given a classifier and an instance, there are four possible outcomes. If the instance is positive and it is classified as positive it is counted as true positive; if it is classified as negative, it is counted as a false negative. If the instance is negative and it is classified as negative it is counted as a true negative; if it is classified as positive it is counted as false positive. Given a classifier and a set of instances, a two-by-two confusion matrix can be constructed, representing the dispositions of the set of instances. This matrix forms the basis for many common metrics:

True positive rate, also known as Recall or Sensitivity,

$$TP\,rate = \frac{\textbf{positives correctly classified}}{\textbf{total positives}}$$

False positive rate, also called false positive alarm,

$$FP\,rate = \frac{\textbf{negatives incorrectly classified}}{\textbf{total negatives}}$$

$$Specificity = \frac{\textbf{True negatives}}{\textbf{False positives} + \textbf{True negatives}}$$

The numbers along the major diagonal represent the correct decisions made, and the numbers off this diagonal represent the errors between the various classes.

ROC graphs are two-dimensional graphs in which TP rate is plotted on the Y axis and FP rate is plotted on the X axis. An ROC graph depicts relative trade-offs between benefits (true positives) and costs (false positives).

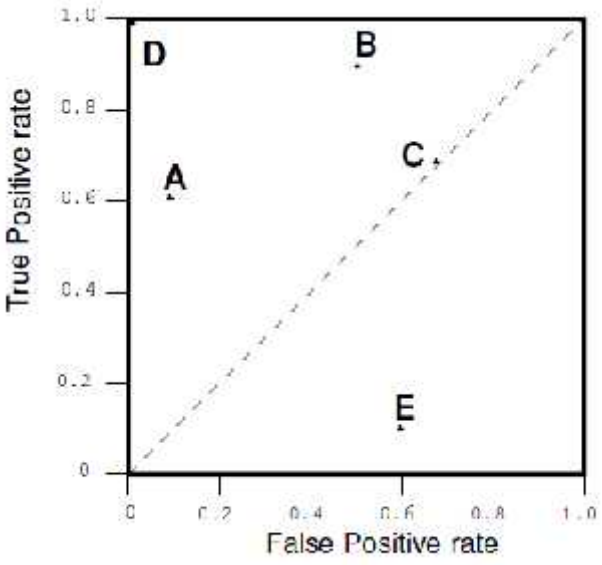

Figure 5. A basic ROC graph showing five classifiers named A - E

A discrete classifier is one that outputs only a class label. Each discrete classifier produces an (FP rate, TP rate) pair, which corresponds to a single point in ROC space. The classifiers in figure 2 are all discrete classifiers.

Several points in ROC space are important to note:

The lower left point (0; 0) represents the strategy of never issuing a positive classification; such a classifier commits no false positive errors but also gains no true positives.

The opposite strategy, of unconditionally issuing positive classifications, is represented by the upper right point (1; 1).

The point (0; 1) represents perfect classification. Informally, one point in ROC space is better than another if it is to the northwest (TP rate is higher, FP rate is lower, or both) of the first. Classifiers appearing on the left hand-side of an ROC graph, near the X axis, may be thought of as conservative - they make positive classifications only with strong evidence so they make few false positive errors, but they often have low true positive rates as well. Classifiers on the upper right-hand side of an ROC graph may be thought of as liberal - they make positive classifications with weak evidence so they classify nearly all positives correctly, but they often have high false positive rates.

The diagonal line y = x represents the strategy of randomly guessing a class. If a classifier randomly guesses the positive class half the time, it can be expected to get half the positives and half the negatives correct; this yields the point (0:5; 0:5) in ROC space. If it guesses the positive class 90% of the time, it can be expected to get 90% of the positives correct but its false positive rate will increase

to 90% as well, yielding (0:9; 0:9) in ROC space. Thus a random classifier will produce a ROC point that slides back and forth on the diagonal based on the frequency with which it guesses the positive class. In order to get away from this diagonal into the upper triangular region, the classifier must exploit some information in the data.

In Figure 5, C's performance is virtually random. At (0:7; 0:7), C may be said to be guessing the positive class 70% of the time, An ROC curve is a two-dimensional depiction of classifier performance. To compare classifiers we may want to reduce ROC performance to a single scalar value representing expected performance. A common method is to calculate the area under the ROC curve, abbreviated AUC. Since the AUC is a portion of the area of the unit square, its value will always be between 0 and 1.0. However, because random guessing produces the diagonal line between (0; 0) and (1; 1), which has an area of 0.5, no realistic classifier should have an AUC less than 0.5.

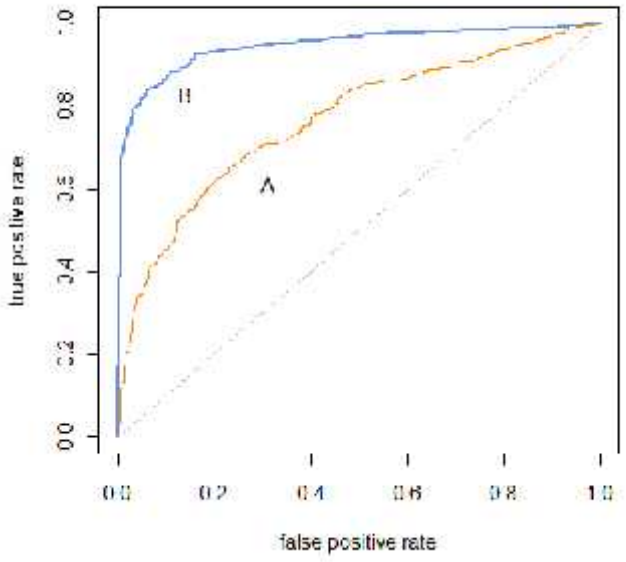

Figure 6. Two ROC curves representing the difference in the accuracy of two classifiers to classify the same data. The dotted gray line x=y represents the „random guess“ classificator.

Figure 6. Two ROC curves representing the difference in the accuracy of two classifiers to classify the same data. The dotted gray line x=y represents the „random guess“ classificator. shows the areas under two ROC curves, A and B. Classifier B has greater area and therefore better average performance. It is possible for a high-AUC classifier to perform worse in a species region of ROC space than a low-AUC classifier.

5 RESULTS

5.1 CONSTRUCTION OF A HIGH QUALITY DATASET

Due to the poor quality of the data in the RSCB databank, files containing spatial coordinates of biologically relevant protein quaternary structures were downloaded from the 3D complex database. (Levy, Pereira-Leal et al. 2006). To maximally reduce the number of falsely assigned interaction residues, obtained structures were subjected to a series of quality filters, as to establish a rigorous dataset.

In the first step all structures with resolution greater than 2.7 Å were removed from the set. From the subset, we then selected all structures that were annotated, in PiQSi database (Levy 2007), as „no PDB error“, and we required that none of the chains in the structure are shorter than 10 residues. The completed dataset now contained ~4100 structural files, out of which 55 % represented dimeric complexes.

All protein structure databases do not have a uniform coverage of the protein sequence space. This is partly because it is much easier to get crystals for homomeric complexes and partly because the research community dictates for which protein structures there is interest. This, in the end, results in the fact that some parts of the sequence space are much more abundantly represented, creating a significant sequence redundancy in the databases.

To remove the redundancy on the sequence level, which could cause a large sampling bias in our statistical analysis, we used the BLASTClust program, as described in the methods section. Our set of ~4100 sequences was clustered to 3694 clusters. From each of the clusters a single chain was selected at random as a representative sequence, for further analysis.

5.2 CALCULATION OF STRUCTURAL CHARACTERISTICS

To make our dataset usable for further exploratory analysis we needed to designate interacting residues on each interacting surface in every quaternary complex and then calculate various structural properties for every AA. Both of these tasks were done using PSAIA (Mihel, Sikic et al. 2008).

For calculation of structural parameter it is necessary to specify a number of parameters:

1. The standard Van der Waals radi, which were taken from (Tsai, Taylor et al. 1999).
2. Radius of the sphere for calculation of accessible surface area – taken as radius of a single water molecule (1.4 Å).
3. An AA pair was designated as interacting if any of the atoms in the AA pair were distant less then 0.5 Å + Van der Waals radii of the corresponding atoms.

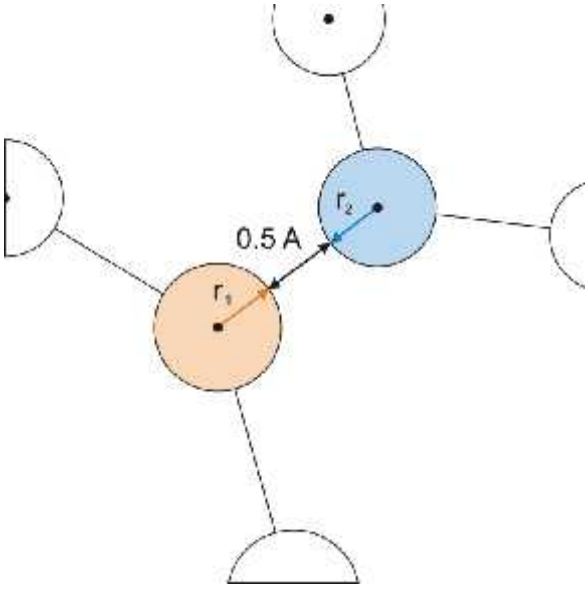

Figure 7. Showing the method used to determine which residues in a pair of structures are interacting. A residue was labeled as interacting if the distance between any of the atoms in the residue was less than 0.5 Å + the Van der Waals radii of the corressponding residues.

For each amino acid we calculated all of the structural parameters supported by PSAIA (Mihel, Sikic et al. 2008). The properties were calculated for each chain in a quaternary structure separately .

5.3 DESIGNATION OF SURFACE RESIDUES

With our hypothesis being that the surface residues on a protein structure are the most important factors for the specificity and strength of an interaction between two proteins, we wanted to narrow the search space by predicting which residues in a given protein sequence (primary structure) are exposed on a surface. To be able to train the Random Forest predictor we needed a large set of true positives and false negatives – residues for which we have one hundred percent certainty of being on the surface and residues which are hidden in the interior of the structure.

By finding which residues in our dataset are interacting, we created a set that is positioned on the surface of the protein structures. By comparison of structural parameters for the interacting set and

the not interacting set of residues we wanted to find which parameter could be used as the best discriminator between buried and exposed AA. We calculated a empirical probability distributions for each structural parameter, separately for the set of the interacting residues and non-interacting ones, expecting to find a bimodal distribution in the set of the non-interacting ones. One peak of the distribution would correspond to the surface residues and the other one would be highly similar to the interacting set.

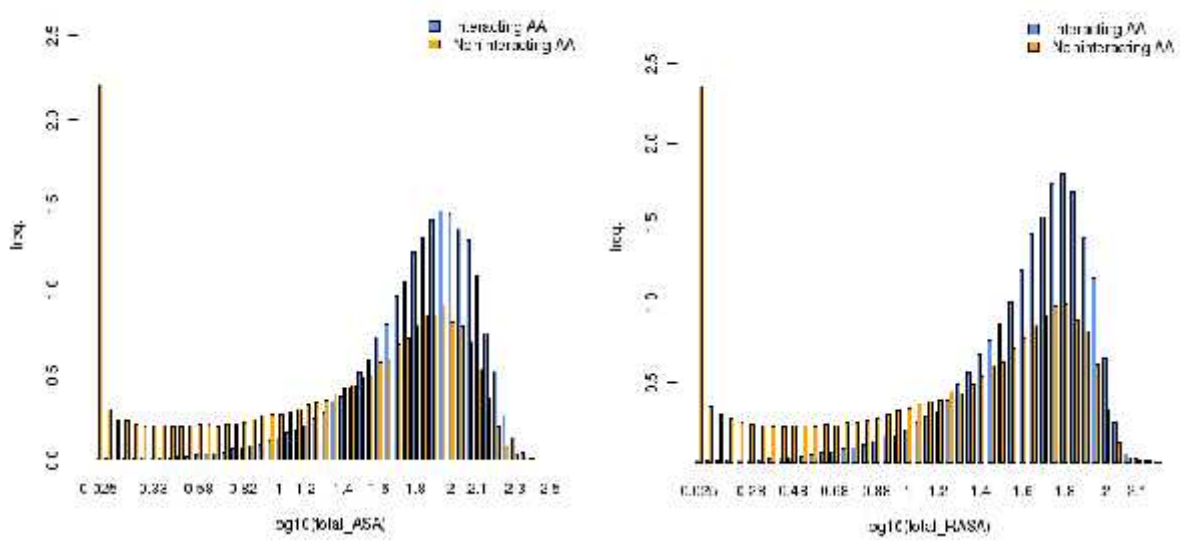

Figure 8. a) log10 of total ASA distribution for interacting and non-interacting AAs. b) log10 of total RASA for interacting and non-interacting AAs

As can be seen on Figure 8, the empirical distributions of ASA and RASA for complete residues (backbone plus side chain) follow our predictions nicely. Both of the parameters showed comparable performance for surface designation, so we choose ASA as for further analysis.

The cutoff value in the distribution was taken as the point of inflexion on the distribution, which resulted in a value of 22.5 $Å^2$ (e.g. all residues with total ASA greater than 22.5 $Å^2$ were taken as surface exposed residues, and all residues with total ASA smaller than 22.5 $Å^2$ were taken as buried).

5.4 PREDICTION OF SURFACE RESIDUES

Attribute vectors were constructed using a sliding window approach. A window of length n (n ϵ {3, 5, 7, 9}; although in the previous studies it was shown that the window of length 9 has the minimal

entropy content (Mihel, Sikic et al. 2008)) was tilled over each sequence in the dataset. The class of the vector (surface vs. buried) was determined according to the class of the middle residue in the vector (Ofran and Rost 2003)(e.g. if the middle residue of the vector was on the surface of the structure, the whole vector was labeled as surface vector, as shown in Figure 9).

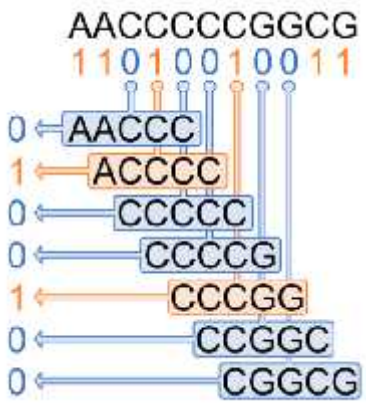

Figure 9. Shows the principle of tilling window vector construction. A sliding widow of length n was moved across the sequence constructing the vectors. Each vector had a class designated according to the class of the middle residue.

The set of vectors was then split 80% - 20% on the training and testing sets correspondingly.

The accuracy of the classificator was visualized using the specificity - sensitivity curve.

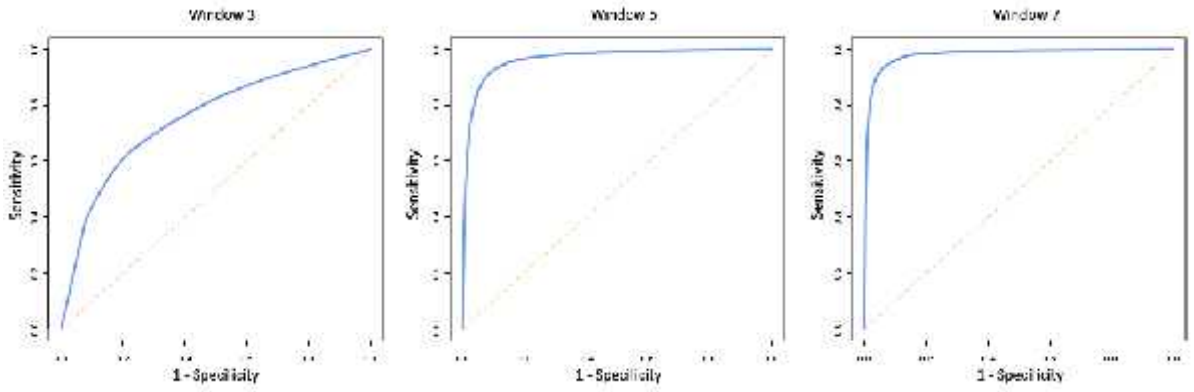

Figure 10. Sensitivity - specificity curve for the prediction of surface residues. The graphs represent the accuracy of the classificator for windows of length 3 – 7. The dotted orange x=y line represent the accuracy of random classification.

Table 1. The AUC measure for surface residue prediction using vectors of length 1-3.

Window length	Area under the curve (AUC)
3	0.755
5	0.965
7	0.980

Random forest classificator performed extremely well in discriminating surface from buried residues, achieving accuracy of 98% when trained on vectors of length 7. Unfortunately the model of window size 9 was not usable due to the restrictions of the available computational resources.

5.5 PREDICTION OF INTERACTING RESIDUES

To be able to predict interacting residues between sequences we needed to construct the feature vectors and annotate them as interacting/non-interacting. First we constructed a contact map for each of the interacting pair of proteins – a graphical representation of N x M matrix (where N is the length of the sequence a and M is the length of the sequence b), where each field is colored based on the interaction status of marginal AA (e.g. if two AA are interacting that position in the matrix is colored, and if they are not, it is left colorless). It was immediately obvious that the interacting residues are not interspersed uniformly among the fields of the contact map, but cluster in certain regions of the plot, thus making evident the existence of interacting patches on the surfaces of protein structures.

We took two approaches to constructing the feature vectors, which differ in the way residues are taken into a vector and in the method of classification of the vector as interacting and non-interacting. In the first approach, for each interacting AA pair, we placed a square with sides of length n ($n \in \{3, 5, 7, 9\}$) with the AA pair at the center. The feature vector consisted of concatenated marginal sequences of the square (e.g. if the square is 3x3, the feature vector was of length 6, and consisted of the AA interacting pair – 2 attributes, and AAs found on +1 and -1 positions relative to the interacting AA in each of the sequences – 4 attributes). Because the number of interacting residue pairs is an order of magnitude smaller than the number of non-interacting residues, the disbalance in the sizes of the true positive vs. false positive datasets would severely influence the

performance of the Random Forest classificator. To get around this, negative dataset was constructed by random sampling the same number of residues as in the positive dataset, but taking into account that the corresponding patch around each non-interacting pair does not contain any interacting amino acids. When the data set was complete we split the data at random into 80% - 20% parts for training and testing the performance of our classificator.

The second approach was a bit more complicated. Prior to constructing the vectors we took our predictor of surface residues and classified each AA in each sequence as a surface residue or a buried residue. Then we removed, from the sequences, all of the AA which were classified as buried, and were left with a reduced dataset that, we hoped, would reduce the sequence space and thus enable the classificator to get better performance. After the removal of buried residues, we again constructed a contact map for each pair of interacting protein sequences and tilled them using the above described method. This time we applied two different criteria for patch designation as interacting of non-interacting. The first one was the same as above, the patch was designated as interacting if the middle amino-acid was interacting, but in the second method we took a patch as interacting if the ratio of interacting vs. non-interacting residue pairs crossed a certain ratio (which was variable based on the patch size). The accuracy of all of the predictors was evaluated using the sensitivity – specificity graph and the corresponding AUC measure.

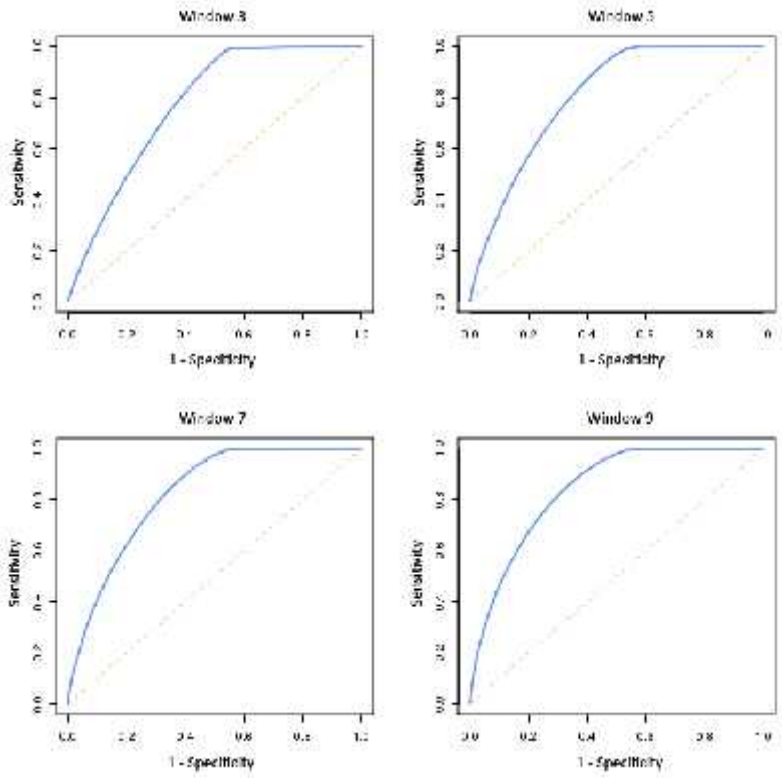

Figure 11. Show the sensitivity - specificity curves for prediction of interacting residues from the sequence alone. The graphs are ordered by increasing window length that was used to construct feature vectors.

Table 2. Shows the area under the curve measure for windows of increasing length used in interacting residue prediction.

Window length	AUC
3	0.7725975
5	0.8052316
7	0.8242232
9	0.841974

As can be seen from the graphs, already at windows of size 3, models achieve 100 sensitivity with 50% false positives. Slope of the curve, at that point, stays the same with increasing window size, even as the area under the curve steadily increases.

Unfortunately, because of poor implementation of Random forest algorithm in R language, the amount of time it takes to predict surface residues from sequences is so large, that it is practicly not feasible to do the analysis on a large dataset. Unless a faster implementation of Random forest becomes avaliable, a truly ingenious algorithmic approach would be necessary for the analysis to be done in a finite amount of time.

6 DISCUSSION

Protein – protein interactions in biological systems can be studied on three different conceptual levels organized in a hierarchical structure of increasing complexity:

1. Macromolecular level - level of interacting proteins.
2. Level of interacting protein domains.
3. Individual interacting AA residues.

Because of the differences in organizational complexity between the levels, each one requires a different set of computational tools for their analysis, and correspondingly, a different set of exploratory and confirmational biological and chemical techniques. Although, both the macromolecular and domain levels are of utmost importance for understanding the inner workings of the cell, knowledge about the interaction on the residue level brings with itself a mechanistic approach, opening a window towards protein function engineering and a path towards synthetic biology.

Unfortunately, the only laboratory techniques with resolution sufficient for mapping interaction residues are, resolution of 3D structures of protein complexes by means of X-ray crystallography, or alanine mutagenesis coupled with thermodynamic measurements- both still a very tedious and slow processes. This coupled with recent advances in sequencing technology, which started generating enormous amounts of biological data needing functional annotation, has prompted a fast development of computational techniques.

Because of their ability to make predictive models, adaptation of machine learning algorithms to biological systems was a natural solution to the problem. Recent years have seen the development of numerous algorithmic approaches for functional classification of interacting residues. Most machine learning methods are based on support vector machines, neural networks or Bayesian networks (Ezkurdia, Bartoli et al. 2009).

However, upon a closer survey, there is a realization that many of the methods are not properly benchmarked, or are tested on the ill-sized data sets, often highly redundant in homologous sequences. Moreover, newer methods are often published without performance comparison with previously proposed ones. Thus, it is not clear how good they are and whether there are significant performance differences among them. These are important issues to investigate for both a true

advancement of this research field and maximizing the benefits of computational predictions for the general research community.

In this work we have used an ensemble classifier “Random forest” which can classify highly dimensional data. In prediction of surface residues it showed exceptional performance, having more than 90% AUC on all feature vector types. Although the performance of the classificator was also substantially better than random, in distinguishing interacting from non interacting AA residues, there still exists the possibility of bias in estimator performance due to sequence redundancy in vector sets. It would be interesting to see the change in classifier performance if the dataset were to be made even more strict – if the sequences were clustered with minimal identity of less than 100%. But it is doubtful whether reducing the number of sequences used to construct the vectors would show the true predictive power of the classifier, or its performance would just be a result of a significant reduction of sequence space coverage.

We have a need for ever-increasing number of data which could be used to make predictive models for functional annotation of biological sequences, but that brings with itself a requirement for development of methods for quality assessment and rigorous and reproducible testing of computational methods to see whether they are useful enough to used on “real world” data and not just biased preselected sets.

7 CONCLUSIONS

- Drastic increase in the production of sequence data necessitates the development of computational methods for functional annotation.
- Random forest classifier performs very well on highly dimensional data.
- It is possible to predict the ASA of a certain residue in a protein from the sequence alone, and Random forest algorithm does it very well.
- There are no standard datasets for comparison of predictive performance of different machine learning methods.
- Accurate prediction of interacting residues requires construction of a more complex attribute set, taking into account long range influences between residues.
- We are still missing a good part of the picture to be able to predict functional characteristics on protein level, purely from sequence data.

8 LITERATURE

Berman, H. M., T. Battistuz, et al. (2002). "The Protein Data Bank." Acta Crystallogr D Biol Crystallogr **58**(Pt 6 No 1): 899-907.

Bock, J. R. and D. A. Gough (2001). "Predicting protein protein interactions from primary sequence." Bioinformatics **17**(5): 455-460.

Breiman, L. (2001). "Random forests." Machine Learning **45**(1): 5-32.

Bruce Alberts, A. J., Julian Lewis, Martin Raff, Keith Roberts, and Peter Walter (2002). Molecular Biology of the Cell, 4th edition. New York, Garland Science.

Causton, H., Quackenbush, J., and Brazma, A. (2003). Microarray Gene Expression Data Analysis: A Beginner?s

Guide, UK: Blackwell Science.

Cole, C. and J. Warwicker (2002). "Side-chain conformational entropy at protein-protein interfaces." Protein Sci **11**(12): 2860-2870.

de Vries, S. J. and A. M. J. J. Bonvin (2008). "How Proteins Get in Touch: Interface Prediction in the Study of Biomolecular Complexes." Current Protein and Peptide Science **9**(4): 394-406.

Ezkurdia, I., L. Bartoli, et al. (2009). "Progress and challenges in predicting protein-protein interaction sites." Briefings in bioinformatics **10**(3): 233-246.

Henrick, K. and J. M. Thornton (1998). "PQS: a protein quaternary structure file server." Trends Biochem Sci **23**(9): 358-361.

Jones, S. (1996). "Principles of protein-protein interactions." Proceedings of the National Academy of Sciences **93**(1): 13-20.

Jones, S. and J. M. Thornton (1997). "Analysis of protein-protein interaction sites using surface patches." Journal of Molecular Biology **272**(1): 121-132.

Levy, E. D. (2007). "PiQSi: protein quaternary structure investigation." Structure (London, England : 1993) **15**(11): 1364-1367.

Levy, E. D., J. B. Pereira-Leal, et al. (2006). "3D complex: a structural classification of protein complexes." PLoS Comput Biol **2**(11): e155.

Liaw, A. and M. Wiener (2002). "Classification and Regression by randomForest." Glass **2**(December): 18-22.

Lichtarge, O., H. R. Bourne, et al. (1996). "An evolutionary trace method defines binding surfaces common to protein families." J Mol Biol **257**(2): 342-358.

Lo Conte, L., C. Chothia, et al. (1999). "The atomic structure of protein-protein recognition sites." J Mol Biol **285**(5): 2177-2198.

Mihel, J., M. Sikic, et al. (2008). "PSAIA - protein structure and interaction analyzer." BMC Struct Biol **8**: 21.

Mintseris, J. and Z. Weng (2005). "Structure, function, and evolution of transient and obligate protein-protein interactions." Proceedings of the National Academy of Sciences of the United States of America **102**(31): 10930-10935.

Neuvirth, H., R. Raz, et al. (2004). "ProMate: a structure based prediction program to identify the location of protein-protein binding sites." J Mol Biol **338**(1): 181-199.

Nooren, I. M. A. and J. M. Thornton (2003). "NEW EMBO MEMBER ' S REVIEW Diversity of protein ± protein interactions." EMBO Journal **22**(14).

Ofran, Y. and B. Rost (2003). "Predicted protein–protein interaction sites from local sequence information." FEBS Letters **544**(1-3): 236-239.

Pintar, A., O. Carugo, et al. (2003). "DPX: for the analysis of the protein core." Bioinformatics **19**(2): 313-314.

Teichmann, S. a. (2002). "The Constraints Protein–Protein Interactions Place on Sequence Divergence." Journal of Molecular Biology **324**(3): 399-407.

Tsai, J., R. Taylor, et al. (1999). "The packing density in proteins: standard radii and volumes." J Mol Biol **290**(1): 253-266.

Zhou, H.-X. and S. Qin (2007). "Interaction-site prediction for protein complexes: a critical assessment." Bioinformatics (Oxford, England) **23**(17): 2203-2209.