Prediction of homodimeric residue-residue contacts through co-evolutionary and sequence-based analysis

Yin Hsieh (950221370020)

WUR BIF MSc. Thesis Report

December 7, 2018

Contents

1 Abstract

The coevolutionary method of predicting residue-residue contacts from protein sequences has existed for several decades, and has increasingly experienced notable improvements. This project applies this method on prediction of the interacting residues in homodimeric protein complexes by evaluating the extent to which a random forest classifier trained on a combination of coevolutionary signal and sequence-based information can distinguish a residue pair as being non-interacting or as an intra- or interprotein contact, with particular emphasis placed on differentiating between these two contact types. Our classification results show that, given these features and the current setup of our random forest, the classifier is largely unable to distinguish between these contacts. The classifier consistently achieves a moderate recall for our three classes, with high precision for noncontacts but very low precision for inter- and intraprotein contacts. Variation of input parameters to the random forest only bring about minor improvements to performance. Additionally, although the coevolutionary value turns out to be the feature of the highest importance, residue pairs with the highest coevolutionary values are not found to be interprotein contacts.

2 Introduction

The study of protein function relies on knowledge concerning a protein's amino acid sequence, structure, sub-cellular localisation, and patterns of interaction of residues within the protein and between other proteins. Well-developed methodologies exist for analysing each of these characteristics, and researchers have been increasingly focusing on the possibility of using sequence-based information to predict protein structural and functional information, with promising results [23]. It is well known, for example, that the basic elements of protein secondary structure have corresponding amino acid sequence motifs, thus facilitating structural prediction [17]. Similarly, subcellular localisation signals can also be found in many N-terminal regions of the protein sequence [5]. Furthermore, homologous sequence information contained in the multiple sequence alignments (MSAs) of a protein family give rise to correlated sequence-based patterns, which are seen as remnants of a coevolving relationship between particular amino acids, often due to a function-preserving interaction between them [2]. These coevolutionary patterns, therefore, can be used to predict residue-residue contacts, and this has been done for the past decades with notable success [8]. These recent advancements in coevolutionary-based contact predictions take place in part due to improvements in computational technique and sequence database availability [6, 20], and the refinement of contact prediction methods is currently still ongoing.

2.1 Objectives

This project is an application of the coevolutionary method specifically in the prediction of interprotein and intraprotein contacts in a homodimeric protein complex. We train random forest classifiers on a combination of coevolutionary signals and sequence-based information, to evaluate whether these features allow us to correctly identify a contact and distinguish between inter- and intraprotein contact types. We briefly elaborate on coevolutionary theory, and motivate our application of the methodology to homodimers.

2.2 Theory of coevolution

The core assumption underlying sequence-based prediction of residue-residue contacts stems from coevolution between proteins or within one protein [19]. Through co-evolution, selective pressures responsible for maintaining function within or between interacting proteins manifest through reciprocal mutations at the amino acid level [13, 21]. Coevolved residues therefore tend to exhibit high mutational correlation, which can either indicate residue proximity, functional importance, or neither of the two and thus be a result of noise. This noise could be entropic, for example, related to the conservation of either residue [28], or technical - related to the phylogenetic effects resulting from inherent sampling bias in sequence datasets used to construct an MSA. Noise can also originate from indirect interactions between residues. Such indirect couplings occur when two seemingly correlated residues both interact with a third entity such as another residue or ion [1, 8, 15, 17], and the correlation between them is thus misleading. Generally, the current global statistical protein contact prediction models in use have reduced technical noise sufficiently and are capable of differentiating between direct and indirect coupling pairs [1, 15, 18, 19, 30].

2.3 Homodimers

We choose to focus on homodimers for co-evolutionary based residue-residue contact prediction primarily out of a biological interest. Protein dimerisation is recognised as a critical regulatory process for many enzymatic, genetic, signalling, and molecular transport functions within a cell [16]. Homodimerisation, in particular, enables structural and population-based control of protein complex formation with minimal energetic expense, which is required for numerous cellular processes. Homodimerisation can also be detrimental and bring about diseases related to protein misfolding and aggregation [24]. For these reasons, the study of residue-residue contacts in interacting homodimeric complexes is relevant to numerous current biological problems.

Applying coevolutionary methods to homodimers introduces a unique, nontrivial problem in that we only build one MSA for one protein in the complex, therefore a high coevolutionary value between two residues could be corresponding to two residues within one copy of the protein, between the two proteins, or both. Therefore, differentiating between the two types of contacts is the motivation behind our research statement.

We present, in the following sections, our methodology and results, and conclude that even though the differentiation between inter and intraprotein contacts is not currently possible the way our methods are set up, there are still further patterns to be learned from our predictions.

3 Methods

Our methodology can be divided into four sequential processes: (1) data processing to narrow our dataset down to a suitable set of proteins (see Fig. 1) and database searching for homologous sequences to construct a putative protein family for each protein, (2) construction of multiple sequence alignments for each of these protein families, (3) protein residue-residue contact prediction through coevolutionary methods on these alignments, and (4) the training and testing of our machine learning classifier. We detail the procedure and parameter choices of each process in the following sections.

3.1 Dataset

The use of co-evolutionary prediction methods and machine learning in this project place constraints on the type of proteins usable as a dataset. Selected proteins should homodimerise, be of moderate size, have a considerable number of known homologs, and have a published structure, deposited in the Protein Database (PDB). We therefore adopt a dataset of verified homodimers curated from the PDB by Hou et al. (2015, 2017) in their previously published works [10, 11], and subsequently implement our own preprocessing and additional filters to keep only those proteins satisfying the desired criteria. The remaining proteins kept constitutes our training and testing dataset for the classifier.

3.2 Data preprocessing

We initially filter the dataset according to the availability and correctness of the sequence and structure files corresponding to a particular PDB entry. Protein IDs that exist as duplicates in the original dataset listing are identified and reduced to one. Proteins with obsolete structure files in the PDB are substituted by their newer counterparts. Proteins with unequal chains (by length and/or sequence identity) are discarded, as are proteins with more than two chains in their structure file. Whereas unequal or multiple chains in a PDB file do not necessarily indicate that a protein is non-homodimeric, we choose to be conservative and eliminate them regardless. Figure 1 illustrates the effect on dataset size of each step. Additionally, in each structural PDB file, we delete the non-standard residues (HETATMs), and all hydrogen atoms attached to standard residues.

We then control the sequence composition of the dataset, to minimise sequence similarity and therefore reduce a potential bias influencing the machine learning training step if we were to train on dataset with high sequence similarity. We do this by first clustering all sequences to determine sequence similarity, then selecting only one representative sequence to keep per cluster. Sequences are clustered using the UCLUST algorithm (*cluster_fast* command) of the USEARCH suite (v11) [3]. Clustering of the dataset is performed for the threshold sequence similarity levels 50-90%, in increments of 10, and the results are evaluated (see Section 4.1, Fig. 4). UCLUST groups sequences based on a threshold similarity to a representative or 'centroid' sequence of the group, such that every sequence within a group shares equal or higher sequence identity with the centroid, with centroids sharing no more than the threshold sequence similarity with each other. Centroid sequences are identified in a greedy manner, based on an initial sorting of the sequences by descending sequence length, so that the larger, more complete sequences are favoured as centroid candidates. Due to these characteristics, centroid sequences are selected as representative

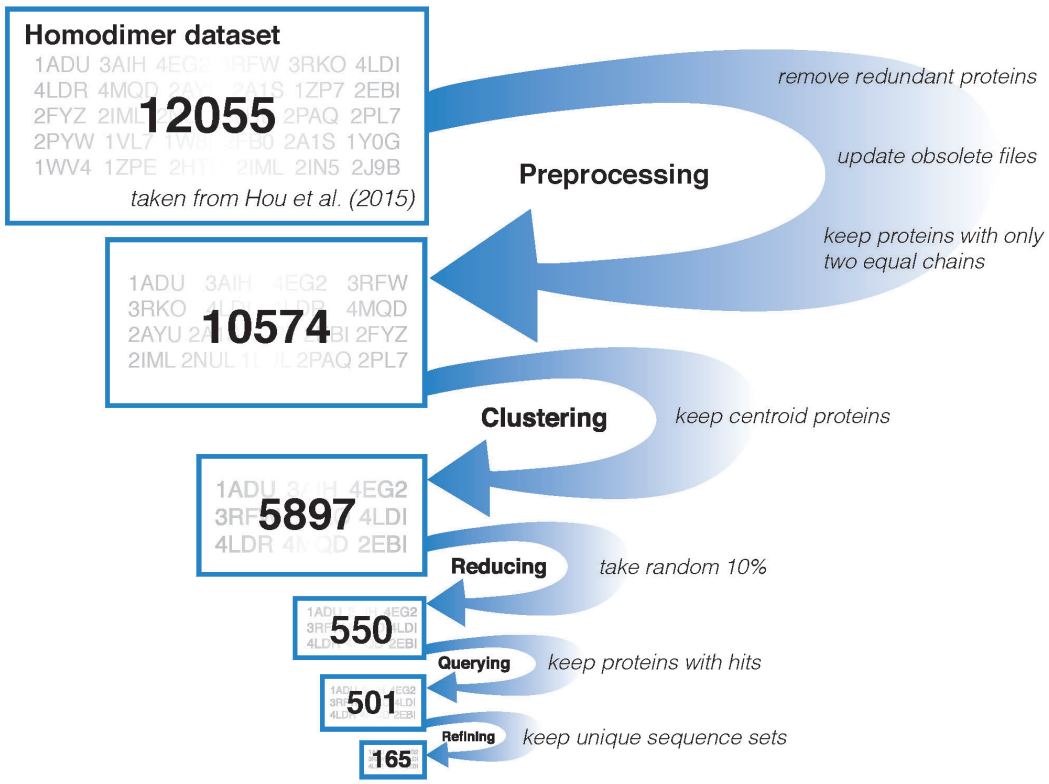


Figure 1: **Data Processing Workflow**: We show how the original Hou et al.(2015) dataset is first filtered to ensure non-redundancy and homodimeric correctness (*Preprocessing*), and then clustered based on sequences to identify and eliminate overrepresented sequences with UCLUST [3], where we keep only centroid sequences of identity no more than 50% (*Clustering*). We then reduce the set for computational reasons (*Reducing*) and submit the remaining protein sequences to Jackhmmer [7], which searches the Uniprot database for homologous sequences (*Querying*). Finally, we compare each set of sequence hits from each protein to identify proteins that share the same sequences, and keep only the proteins with unique sequence sets (*Refining*). This is the procedure by which we decrease the original 12055 proteins to the final 165 proteins that are usable for this project, with intermediate steps as shown in the diagram.

sequences. We take the clusters generated at threshold identity 50%, which is considered the lowest identity threshold at which UCLUST produces reliable results [3]. We select the centroid sequence from each cluster, and discard the remaining member sequences of the cluster from the dataset. Together, all the centroid sequences form a new dataset.

In the final data processing step, we further reduce this dataset by putting a cap on sequence length at 600 residues, and randomly taking 10% of the remaining sequences, to produce our final dataset. This is done to reduce the computational time necessary in the database search and multiple sequence alignment steps.

3.3 Database querying for homologous sequences

Coevolutionary methods predict contacts based on detecting mutational interdependencies between positions of multiple sequence alignments (MSA) of a protein and its family. We therefore

query each protein sequence against a protein database to generate additional sets of related proteins which are considered their putative families with which we can build multiple sequence alignments (MSAs). The query protein we henceforth refer to as the *reference protein* or *reference sequence*, and its set of sequence hits as the reference protein's *sequence set*. Jackhmmer (v3.1b2) [7] is used to iteratively query each protein sequence in our dataset against the UniProtKB/SwissProt database (release ver. 2018_07) and search for homologs using hidden Markov model profiles. We thus take the list of Jackhmmer hits within the inclusion threshold per reference protein sequence to be its putative protein family (see Fig.2, (A)). Parameters for Jackhmmer are kept at the defaults, and this applies to the sequence scoring thresholds (gap open probability 0.02, gap extend probability 0.4, substitution matrix BLOSUM62), the inclusion threshold (E-value ≤ 0.001), and the maximum number of iterations set at 5.

Sequence length can differ drastically in a group of Jackhmmer query hits, which can be problematic for coevolutionary contact prediction because the MSAs constructed from sequence sets with varying lengths will contain large consecutively gapped segments, regions where information is lost. Therefore, for each group of homologous protein sequences, we filter based on the median sequence length, by keeping only the reference protein and sequences in its sequence set of length between 0.5 to 1.5 times the median length (in other words, a range of 50% above and below the median length). The effects of this so-called *median-length filtering* step on the size of sequence sets used for building MSAs is shown in Section 4.1.2, Fig. 5, plot (a).

Additionally, proteins for which Jackhmmer returns no hits above the inclusion threshold are discarded from the dataset, as they cannot be used. We also compute pairwise comparisons each set of sequences to identify sequence sets which contain the same sequence hits. To show the degree to which sequence redundancy is an issue, we construct a network where reference protein identifiers are connected based on the number of sequences shared between their sequence sets (Section 4.1.2, Fig. 5, plot (b)). We only keep reference proteins for which no sequences are shared between their sets. This is to ensure that all MSAs are unique and, once again, reduce potential bias when training our machine learning algorithm.

3.4 Multiple sequence alignment

ClustalOmega (v1.2.1) [26] is used to build a multiple sequence alignment per protein family (defined as: reference protein and its filtered set of Jackhmmer hits). All parameters are set to defaults. For each MSA, we further remove all alignment positions for which the percentage of gaps exceeds a threshold of 50% of the total number of rows. To assess the sequence redundancy in each MSA, or, how much potential contact information is contained within each alignment, we compute the effective number of sequences, N_{eff}, equal to the number of clusters formed from the alignment sequences at 62% identity threshold. Henceforth when we refer to the size of an alignment, we are referring to the N_{eff} and not the raw size.

3.5 Coevolution-based contact prediction

We calculate the co-evolutionary strengths (or *coupling scores* [25]) or the potentially meaningful covariance between MSA positions, through contact prediction with CCMpred (v.0.1) [25] - an efficient, C-based pseudolikelihood maximisation implementation of direct coupling analysis (plmDCA) [4, 25]. CCMpred takes as input a multiple sequence alignment (length L) and returns

a $L \times L$ square matrix with coupling scores. These scores, as [25] indicates, are not probability scores, but log-odds ratios comparing the potential co-dependency of two amino acids in different positions to the random state, where both residues in positions evolve independently of each other. The main assumption is that higher scores are more associated to coevolving (and interacting) positions, and lower scores can indicate that the residues do not interact. However, the reality is not as straightforward, since interacting residues occupying highly conserved positions will not be detected by this method. We discuss this further in Section 4.3. We keep all CCMpred parameters at defaults except the number of iterations, which we increase from 50 to 100 to improve predictive performance.

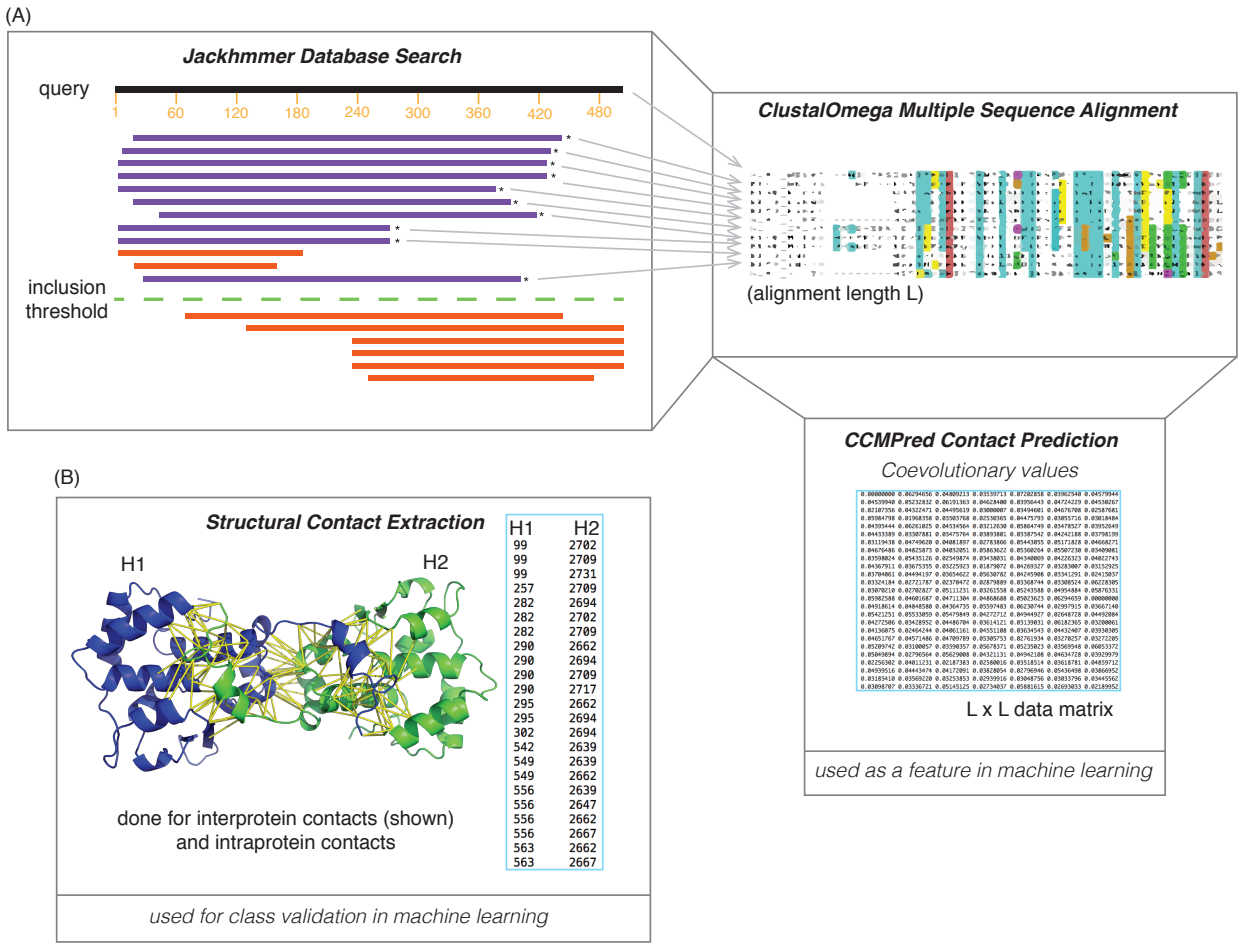


Figure 2: **Procedure for Obtaining Coevolutionary Values and Verified Contacts**: (A) As described in Sections 4.3 - 4.5, we define a protein family as the set of sequences returned as hits above the inclusion threshold for a Jackhmmer query, and filter the sequences based on the median length constraint. The sequences passing the filter (shown in purple, labeled by asterisk) comprise the protein's sequence set, which are aligned by ClustalOmega. The resulting MSA is given to CCMpred, which implements plmDCA to calculate the coevolutionary signal (contact predictions) between every position in the MSA. The coevolutionary signal is a feature in our machine learning feature set. (B) True contacts are extracted from the structure for inter- and intraprotein cases, and used as validation for our machine learning classifier.

3.6 Contact extraction from structure

In addition to predicting contacts, we also extract the structural residue-residue contacts, to use as validation of the predictive performance of our machine learning algorithm (see Fig.2, (B)). A contact is defined as two residues for which the Euclidean distance between their Cβ atoms (or Cα in the case of glycine) is less than or equal to 8 angstroms, and we calculate the contacts both within one copy of the protein (intramolecular contacts) and between the two proteins in the homodimeric complex (intermolecular contacts).

3.7 Random forest classification

Our random forest classifier is built to take input in the form of a list of residue *pairs* (defined below) and return one of three possible classifications (0 - noncontact, 1 - interprotein contact, 2 - intraprotein contact) for each pair, based on learning from features derived from the protein sequence and the MSA of its homologs.

3.7.1 Definition of a pair

Given a protein σ that forms a homodimer, if we label its amino acid sequence as $\sigma = \{\sigma_1, \sigma_2, ..., \sigma_N\}$, where N is the sequence length, a *pair* is defined as any tuple of the set given by taking the Cartesian product of σ with itself. For a peptide of length $N = 3$ for example, our set of pairs would be:

$$\{(\sigma_1, \sigma_1), (\sigma_1, \sigma_2), (\sigma_1, \sigma_3), (\sigma_2, \sigma_1), (\sigma_2, \sigma_2), (\sigma_2, \sigma_3), (\sigma_3, \sigma_1), (\sigma_3, \sigma_3), (\sigma_3, \sigma_3)\} \tag{1}$$

There is a bit of ambiguity in this definition. A homodimer contains two identical copies of the same protein, so a single pair, for example - (σ_1, σ_2), can be interpreted as the pair consisting of the first amino acid from one protein and the second amino acid from the other protein - *or* - the first and second amino acids from the same protein. We adhere to the first interpretation, that the first and second element of a tuple (σ_a, σ_b) are always from different proteins, and therefore there is no symmetry (i.e. $(\sigma_a, \sigma_b) \neq (\sigma_b, \sigma_a)$, where $a \neq b$).

Furthermore, pairs given as input to our machine learning algorithm correspond only to pairwise combinations of residues that are present both in the PDB structure and the alignment, since only these pairs have a co-evolutionary value (arguably our main feature of interest) and a structural verification of the class. In other words, pairs consist only of combinations of residues present in the reference sequence from the MSA that are also present in the PDB structure. This means that residues from the reference protein that aligned to large gapped regions in the MSA that are thus removed from the alignment editing step, (see 3.4) are not included, and that missing residues are also not included in the pairs.

3.7.2 Cross-validation

We take a non-exhaustive cross-validation (CV) approach to divide our original dataset into training and test sets, through the '*leave p-groups out*' method, a variant of k-fold CV that allows for more control over the membership of each fold. Instead of arbitrarily partitioning the dataset into k equally-sized subsets, we group the dataset in such a way that, for a user-determined number

k (where $k > 1$) of groups, three criteria are met: (1) group membership is unique across all groups, (2) group sizes are (roughly) equal, (3) and the distribution of sequence lengths within each group is also approximately equal across all groups. The user defines a value p (where $p < k$) - which is the number of groups to be held out as a test set, and the remaining $k - p$ group(s) serve as the training set. The baseline assumption behind the choice for this approach is that the number of potential contacts varies based on sequence length of a protein (and of course, its structure, but here we take only the sequence into consideration, as our classification problem is to ultimately predict inter/intracontacts for proteins that do not have structural data). A larger homodimer may, on average, have more contacts within one protein and between the two proteins in the complex, although this is not necessarily always the case. Regardless, protein size plays a factor. We thus manually ensure that our classifier is trained on pairs taken from proteins of mixed sizes (but of roughly the same mix of sizes), and is an attempt to alleviate the potential bias of protein size on the final pairs given as input to the machine learning training step. Due to the expected low number of true interprotein contacts, our CV step does not take into account the protein from which the pairs originate, for instance, by randomly sampling only a certain number of pairs from each protein in a group, and instead, takes all pairs from all proteins for training.

3.7.3 Feature encoding

We extract five types of numeric features from the alignment and the sequence to train our classifier. Aside from the coevolutionary signal, which is computed per pair, all features are calculated per residue (in the pair). Table 1 shows the definition and calculation procedure of each feature type.

3.7.4 Balancing input

Our training data, a matrix with a residue pair per row, and a feature per column (or set of columns), is highly imbalanced, as less than 5% of all pairs are actually contacts. Out of all true contacts, less than 2% are interprotein contacts, and the rest are intraprotein contacts. This is problematic since interprotein contacts are the main category of interest in our classification problem, so we balance our input. We do this by keeping only a certain number of data points per class equal to the total number of interprotein contacts. Data points from the noncontact and intraprotein categories are randomly sampled from the original data matrix. Together, these points and the interprotein contacts form a new datamatrix. Additionally, a residue pair (i, j) can be characterised as inter- and intra-protein if there exists a contact between residues i and j of the same copy of the protein and between the two proteins in the homodimer. We remove such data points from the training matrix, as our focus in this preliminary study is on differentiating between truly distinct inter- and intraprotein contacts.

3.7.5 Random forest classification

We use a random forest classifier to predict, per feature vector of residue pair, whether the pair is a noncontact, interprotein, or intraprotein contact. The random forest algorithm randomly takes subsets of the training data and features to grow a number of decision trees, and predicts the class of a sample by effectively averaging over the resultant outputs of all its trees. We use the Python package SciKit Learn's (v.0.20.0) [22] implementation of random forest classification (*sklearn.ensemble.RandomForestClassifier*) to compute our predictions. We vary the number of trees constructed (10, 100, 500, 1000), but otherwise stick to default parameters (e.g. number

Table 1: **Feature Set**: This feature set consists of characteristics of protein sequences considered important in determining whether two residues are in contact or not, and serves to add information to the coevolutionary value, our main feature of interest. In total, a data matrix built with these encoded features has the dimension (n points x 68 feature columns).

Feature	**Explanation**	**Calculation**
(1)Coevolutionary signal c : $c \in [0, 1)$	Mutual covariance between MSA positions, calculated per pair of residues	CCMpred [25]
(2)Amino acid identity	Proportion of each amino acid found in column of MSA	Vector of 21 percentage identities (20 amino acids + 1 gap) per residue
(3)Predicted solvent accessibility	Probability of residue being buried ($p : 0 - 10$), medium ($p : 11 - 40$), exposed ($p : 41 - 100$)	RaptorXProperty [14]
(4)Predicted secondary structure	Probability of residue belonging to one of eight categories: (alpha helix H, 3-helix G, 5-helix I, extended strand in beta-ladder E, isolated beta-bridge B, hydrogen bonded turn T, bend S, and loop L) [29]	RaptorXProperty [14, 29]
(5)Local sequence conservation	Degree of conservation in a position of the MSA is given through sequence entropy	Sequence entropy: $E_i = -\sum_{x \in \{A,R,N,...\}} k_{i,x} log(k_{i,x})$ where i is a position in MSA, x is the set of all amino acids, excluding gaps

of variables randomly sampled is equal to the square root of the number of features, and the minimum impurity decrease is set at 0).

3.7.6 Performance evaluation

The performance of our random forest classifier is evaluated through the precision, recall, specificity, mean accuracy, and F1 scores of each of the predictions. Figure 3 shows the setup of our confusion matrix, and the computation of the performance metrics. We compare the performance to what we would expect from a random prediction scenario, through computing the Matthews correlation coefficient for the binary case (contact vs. noncontact) and the extension thereof to a multi-class case (for all three classes), the K-category correlation [9]. The Matthews correlation coefficient takes into account class imbalance in a test set and gives the correlation between predicted classifications and true classifications. A value of 1 indicates that all predictions were correct, a value of -1 indicates that all predictions were the reverse from the true class, and 0 indicates that there is no correlation, or that the predictions could have just as well been random.

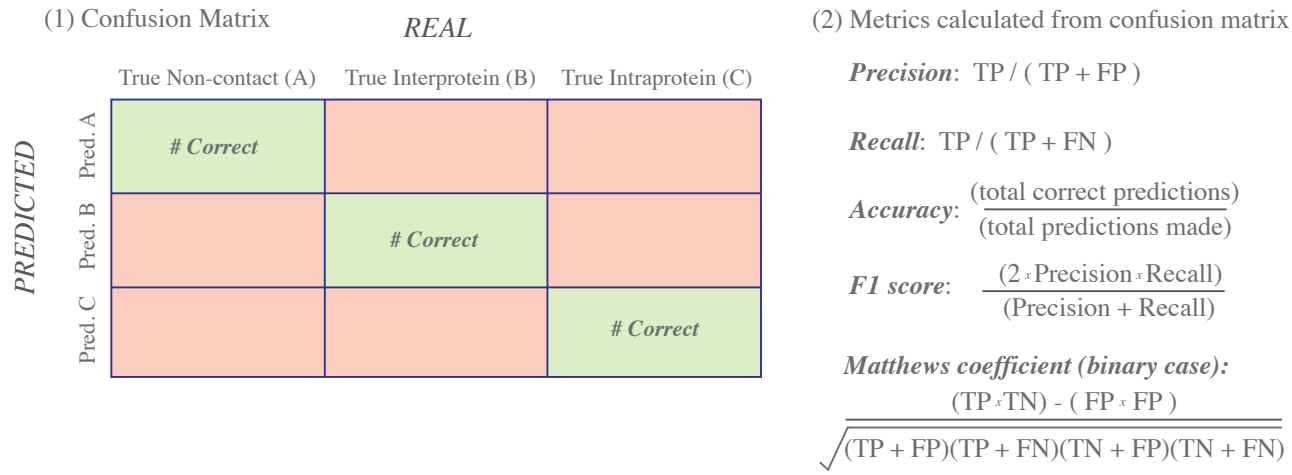


Figure 3: **Classifier Performance Metrics**: We evaluate the performance of our classifier through the following metrics, extended to the 3-class case. The Matthews correlation coefficient shown is the equation for the binary classification case, to illustrate in terms of TP / FP / TN / FN. The calculation and application of this correlation coefficient to our 3-class case follows the K-category correlation given in [9].

4 Results & Discussion

4.1 Dataset Definition

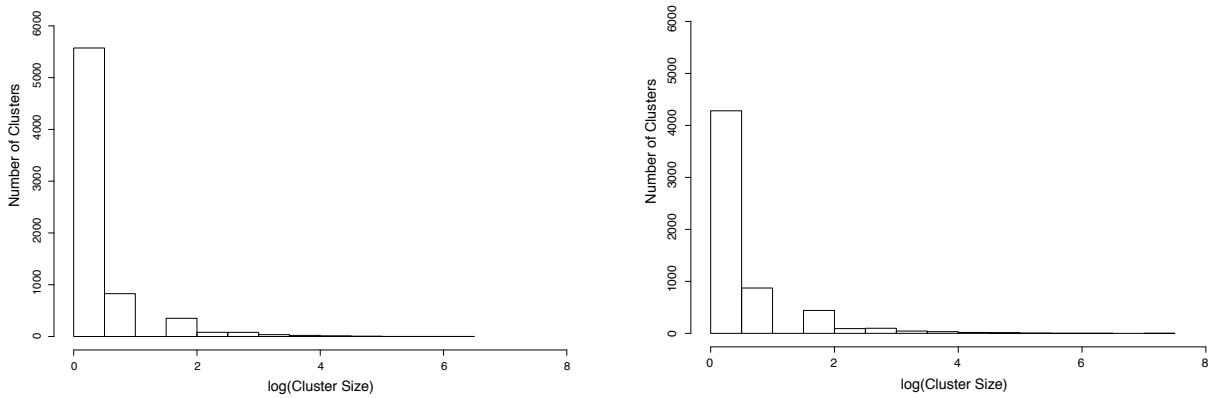
The Hou dataset consists of a total of 12055 homodimeric proteins (all homodimers from Test-set 1 and Test-set 2 combined, as given in [10]), taken originally from the PISA database [12] and given as corresponding PDB IDs. Sequence lengths of the proteins varied from just above 50 residues to several thousand residues, with the majority falling in the 300 -1000 residue range, and a quick pass of the identifiers through RCSB PDB's web-based summary reporting service reveals a dataset consisting of a diverse array of membrane, nucleus, cytoplasmic, and intracellular homodimers, largely of bacterial or archaeal origin but also from humans and model eukaryotic organisms. The exact biological origin of a homodimer is not particularly of interest in this project, since we are interested in finding overall patterns not specific to a particular type of homodimer. What is important, however, is maximising overall diversity (measured as sequence variability) in terms of the proteins contained in our dataset, as this has implications for the performance of our classifier, by reducing the amount of bias in the training process. Therefore, much of our data processing is aimed at reducing similarity in the dataset.

One important concept we purposely do not consider when maximising diversity in our dataset and minimising redundancy is structural variability. A robust classifier should be trained not only on a diverse set of sequences, but also sequences corresponding to variable structures, since ultimately, the intercontact and intracontact differentiation problem is one of structural nature. However, we do not perform any sort of structural alignment to filter out structurally similar sequences because the goal of the classifier is to be applicable to sequences without structures. It could be the case that this choice biases our dataset towards a certain type of structure, therefore impacting our classifier results.

4.1.1 UCLUST clustering to identify overrepresented sequences

One of the initial steps of our data processing procedure involves clustering all proteins in the dataset to test the sequence similarity, and keeping one protein (centroid sequence) per cluster for the reduced dataset. Clustering results show that the Hou dataset, despite being biologically diverse, contains highly similar sequences. Given the total dataset of roughly twelve thousand protein sequences, UCLUST returns, per threshold identity level, an average of 5000 clusters, with a consistent $\sim 65\%$ of these clusters existing as singletons and the overall average membership per cluster found to be 2 sequences. The top hundred largest clusters per threshold level, however, still contain from 10 to 150 sequences, indicating the presence of some overrepresented sequences in the original dataset.

In Fig. 4, we show the histograms of the results of the highest and lowest similarity threshold. Aside from a decreasing number of singleton clusters, lowering the sequence identity threshold does not affect much the clustering results, in terms of the number of total clusters and the distribution of cluster sizes. We refine our dataset based on the lowest similarity level, 50%, to allow cluster centroids maximum sequence variation with respect to each other, since the centroids are the reference proteins that we chose to keep in our dataset. At this 50% sequence similarity threshold, we have 5897 clusters, so selecting the centroid sequences and taking 10% of this set, after controlling for max sequence length of 600 residues, reduces the dataset to 550. For a review of the dataset sizes at each step of our reduction procedure, refer to Fig. 1 in the Methods.



(a) Clusters at 90% similarity level

(b) Clusters at 50% similarity level

Figure 4: **Two histograms of UCLUST clustering results:** Plots of the total numbers of clusters against the log_2 transformed cluster sizes (total number of sequences belonging to a cluster) for the maximum identity threshold, 90% (a) and the chosen identity threshold for our study, 50% (b). Overall distributions of the cluster sizes do not change drastically as the cluster similarity threshold is decreased, although the number of single-sequence clusters does decrease. The existence of clusters with many sequences verifies that our dataset has some sequences of high similarity.

The reduced set of 550 proteins is the set of sequences we submit as query to Jackhmmer database searching. Our decision to select the 10% of the 5897 centroid sequences is to avoid the computational load of for performing Jackhmmer sequence searching for thousands of sequences. In hindsight, this may have been too low of a percentage, considering the impact of the following filters on the size of the dataset.

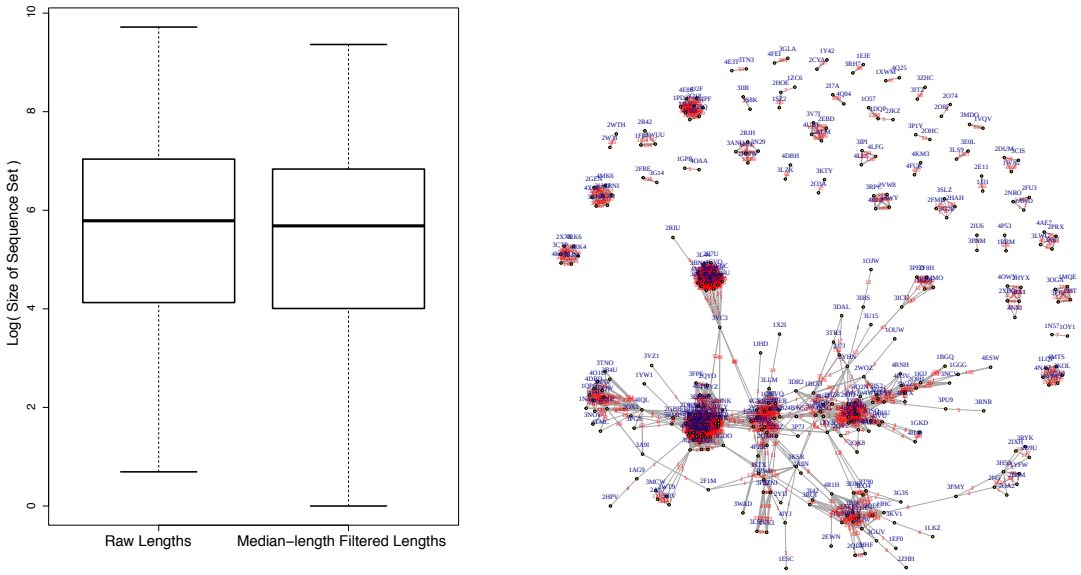
4.1.2 Jackhmmer database search

We use Jackhmmer database searching to build protein families per protein in our dataset, from which we can construct MSAs on which to apply coevolutionary contact prediction. With respect to the quality of our Jackhmmer results, we are primarily interested in how many hits Jackhmmer returns per protein within the inclusion threshold (which we refer to as a protein or query sequence's *sequence set*), how the size of a sequence set compares to the sequence length of the original query, and more importantly, if there are redundant sequences being returned from separate queries. We evaluate the sequence set sizes in Figure **??**, log-transformed. The number of sequences Jackhmmer returns per query is variable, ranging from 0 to 16596, with a median of 326, an average of 1052, and a standard deviation of 1818. There are 49 sequences for which jackhmmer found no hits above the inclusion threshold, and these are discarded from our dataset, reducing the dataset to 501.

Additionally, there is no observable correlation between sequence length of the reference protein and the number of hits returned, and we discuss the comparison of these two terms more thoroughly with N_{eff}, in the following section.

We filter each sequence set by keeping only the sequences of length between 50% to 150% of the median sequence length in the set. This is to eliminate sequences that, in our alignment step, would introduce large gapped regions, from which no coevolutionary information can be drawn. This median-length-based filtering step length reduced the sizes of the sequence sets slightly (refer to boxplots of Fig. 5 (a)), amounting to an average deduction of about 12% of the original number of sequences.

After this filtering step, to check how many sequences are shared between each set of sequence hits, we compare the sequence sets from all possible protein pairs and use this information to build a network: where a node represents a set of sequence hits (labeled by its query protein PDB ID), an edge represents at least one shared sequence, and the thicker the edge, the more sequences are shared between the two sets (Fig. 5 (b)). In constructing the network, we purposely leave out singular nodes that represent sequence sets that do not share hits with any other sets (total of 165), to visualise the degree to which sequences are shared, and how problematic this may be. Each of the 9 tightly clustered groups of sequence sets in our network share more than 1000 sequences within the groups. Instead of developing a fair way to salvage these sequences, we choose to discard all sequence sets that have any shared sequences, and keep the remaining 165 proteins and sequence sets as our final dataset. 165 proteins is not a large set, but considering that the input to training our classifier will not be in terms of proteins but in terms of all potential pairs of residues in a protein, the dataset size is satisfactory.



(a) Effect of Filtering on Numbers of Sequences

(b) Network of Shared Sequences

Figure 5: **Summary of Jackhmmer Results:** In (a) removing sequences of length 50% above and below the median length of the set of Jackhmmer sequence hits per protein results does not bring about a drastic change in total number of sequences per set, but an average reduction of 12%. Our network in (b) shows how many and between which sequence sets (labeled by PDB ID of their query protein) sequences hits are shared, basically how many unique sequences Jackhmmer database searching returned overall. From this information we exclude proteins for which their sequence set shares any sequences with another.

4.1.3 Raw and effective MSA sizes

The larger the size of our multiple sequence alignments, the more information they contain from which coevolutionary methods can more accurately derive contact predictions. This is valid only if the sequences are nonredundant and show enough sequence variation between them, which we measure through the number of effective sequences, the N_{eff} value. We define the N_{eff} as the number of sequence clusters present if we cluster all sequences based on a threshold of 62% sequence similarity. Figure 6 shows the N_{eff} of all proteins with Jackhmmer hits before median-length and redundancy filtering (dataset of 501 sequences) plotted against the sequence length of their representative (centroid/query) sequence. In Fig. 6, black dots are proteins for which their sequence set was discarded in the filtering, and red dots represent the sequence sets that are included in the final dataset of 165 proteins. Points lying above blue line ($y = 5x$), show how many proteins satisfy the *5L rule* of having a total number of sequences in their sequence set equal to a minimum of five times the reference sequence length, which is the general consensus regarding an acceptable alignment depth [10]. Unfortunately, the majority of sequences do satisfy this rule. Furthermore, there are only six proteins from the final set of 165 that lie above the line. We would ideally to apply the coevolutionary method on proteins with a sufficient N_{eff} value, but this is not feasible from this set at least and we stand to reduce our dataset too drastically and bias our machine learning classifier towards smaller proteins if we choose only those proteins that do fit the 5L rule, so we opt to not implement an extra layer of filtering based on N_{eff} and instead keep the existing 165 protein set.

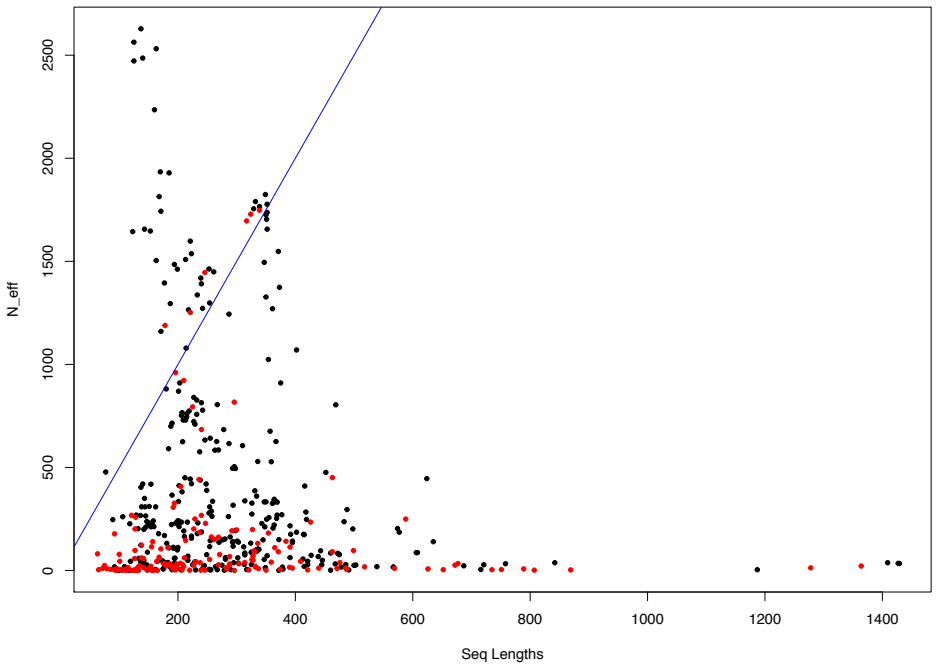


Figure 6: **Comparison of N_{eff} to Protein Sequence Length**: Proteins with sequence sets that satisfy the 5L rule [10] of having an N_{eff} equal to or higher are 5 times the sequence length are shown above the blue line. Black points represent proteins with sequence sets removed by the median-length filter or the shared sequence filter, and red points represent sequence sets that passed both filters that we keep. There are many proteins in our dataset with an insufficient N_{eff} value.

4.2 Random forest classifier

In order to discern the effects of different combinations of parameters on the performance of our classifier, we build random forest models based on the scheme in Table 2. The two parameters we aim to optimise are (1) the size of training and testing datasets, and (2) the number of trees built in the random forest. Therefore, each model built is given a unique identifier based on the cross-validation parameters used to split the dataset into training and testing set (the CV grouping), the number of trees in the random forest (Nr. Trees), and model number.

To illustrate: *G3P1* corresponds to all the models built from forming 3 groups ('G3') out of the total dataset, and leaving 1 group out ('P1') as testing. This is done over three iterations (there are three unique ways to leave one group out), so each of the groups appears at most once in the test set, and the remaining two groups form the training set. *G3P1_10_1* thus refers to the first of the models built from the *G3P1* category specifically when the random forest is built with 10 trees, and *G3P1_10_2* would be the second model from the same category. Similarly, *G3P1_100_1* would be the first model with 100 trees, *G3P1_100_2* the second model, and so on. We stress model number because within the same CV grouping category, model numbers correspond to classifiers subject to the *same training and testing sets*. This means that *G3P1_10_1* is trained on the exact same set of proteins as *G3P1_100_1*, *G3P1_500_1*, and *G3P1_1000_1*, and also tested on the same proteins. The only difference between these models is the number of trees built in the random forest. This is done to exclude the effect of variation in composition of training and testing sets when evaluating an optimal number of trees for our random forest. Due to time constraints, we only vary the number of trees for the *G3P1* and *G5P1* groups of models, and the remaining

models are built from 1000 trees, with the assumption that a higher number of trees gives us a more accurate classifier.

Varying the CV grouping affects the number of residue pairs included in our training and testing sets, but we do not directly set this number. Instead, the number of pairs is determined by the sizes of the proteins in either set. Table 2, Columns 2 - 4, give an overview of the numbers of residue pairs and the ratio between this number between the training and testing sets. These ratios are low because we balance our input before the training step, requiring equal numbers of pairs per class based on the smallest class, the interprotein contacts. If we did not balance the set, we would be training our classifier on 20 times more noncontacts than contacts, and predictive power for our group of interest, the interprotein contacts, would be drastically reduced. Furthermore, considering that we have only 165 proteins, our maximum number of groups is set to 80 (translates to approximately 2-3 proteins per group). We consistently leave only one group out for practical computational reasons, namely - to avoid the need to build more than three thousand models if we were to choose *G80P2*, for example.

Table 2: **Model Labelling Scheme**

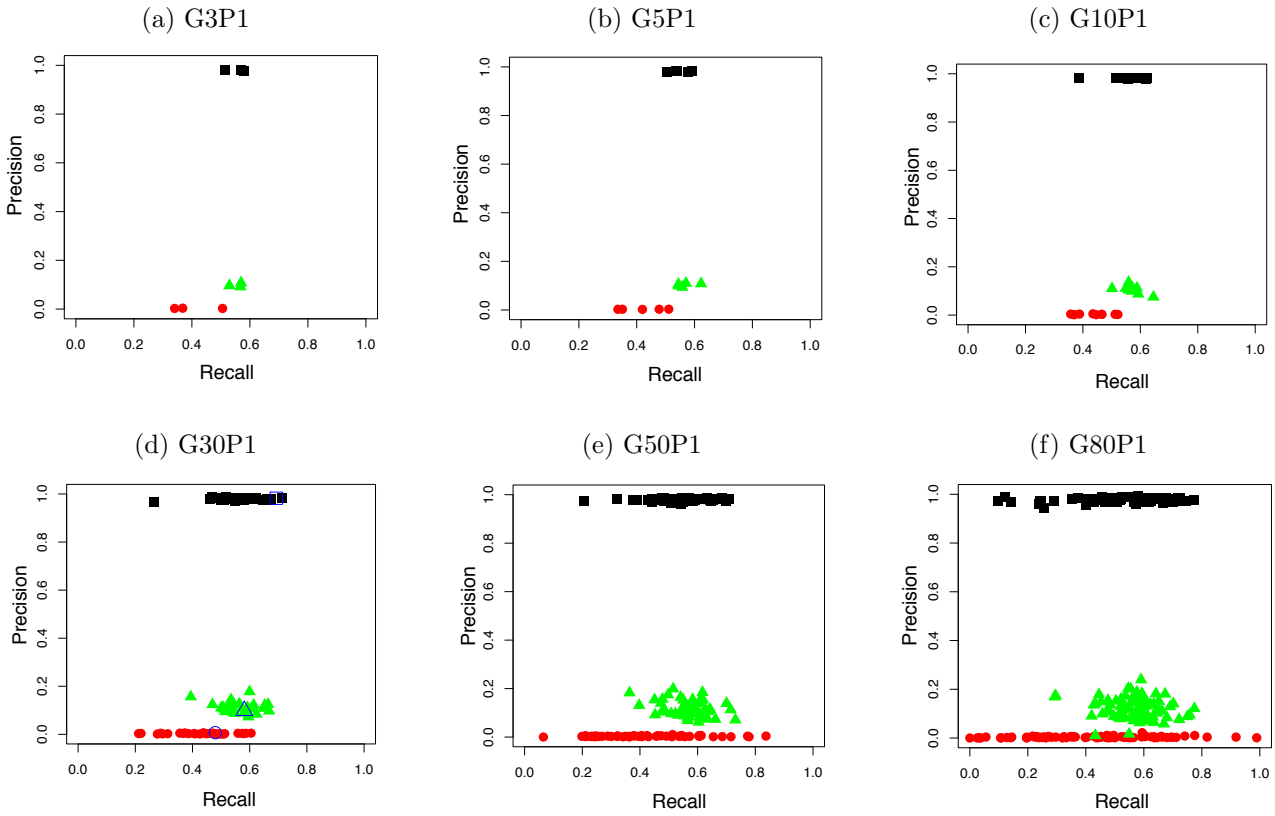
CV Grouping	**Training set size**	**Testing set size**	**Size ratio (train:test)**	**Nr. Trees**	**Total Nr. Models**
G3P1	21900 pairs (111 proteins)	2476806 pairs (54 proteins)	0.009	10,100 500,1000	12 (3 x 4)
G5P1	26280 pairs (132 proteins)	1486084 pairs (33 proteins)	0.018	10,100 500,1000	20 (5 x 4)
G10P1	29565 pairs (149 proteins)	743041 pairs (16 proteins)	0.040	1000	10
G30P1	31755 pairs (159 proteins)	247680 pairs (6 proteins)	0.140	1000	30
G50P1	32193 pairs (161 proteins)	148608 pairs (4 proteins)	0.267	1000	50
G80P1	32439 pairs (162 proteins)	22880 pairs (3 proteins)	0.539	1000	80

4.2.1 Overall performance

We evaluate the overall performance of our classifier first based on the general patterns in the prediction metrics reported between all three classes for all models. The precision and recall of all three classes (depicted in Fig. 7) in each 10000-tree classifier model per CV grouping show a distinct pattern such that prediction of noncontacts consistently achieves high precision (between 0.97 - 0.98 approximately), whereas interprotein contacts are relatively hard to predict (precision around 0.002 - 0.007). The precision of intraprotein contact prediction is slightly higher than that for interprotein contacts, ranging from 0.005 - 0.02, and is more variable within this range. Recall for all three classes for centers around 0.4 - 0.6, and generally as the number of groups increases (training set size increases, testing set decreases) the variance of recall values for all classes increases but particularly for the interprotein contacts. The highest and lowest recall values are observable only for the prediction of interprotein contacts, which may be due primarily to the difference in total number of interprotein contacts vs. the other classes, but also to an instability in predictions

introduced by testing only on a small set of 2 - 3 proteins, which generally occurs in models from CV grouping *G80P1*.

Figure 7: **Precision vs. Recall Plots for All Models**: We show precision and recall for all three classes (noncontact - black squares, intraprotein contact - green triangles, and interprotein contact - red circular dots) for all classifier models built with 1000 trees from each CV grouping. Recall for all three classes is not high, but tends to be higher for noncontact and intraprotein contact prediction. Noncontact prediction ranks the highest in terms of precision, followed by intraprotein and interprotein contact prediction. In (d), the blue triangle, square, and circle represent the precision and recall for the three classes in *G30P1_1000_1*, which is shown in detail in Fig. 8.



In Fig. 8 we focus on one particular model, *G30P1_1000_1*, chosen for being representative of the general predictive results of all models. Despite the large difference between precision values of noncontact vs contact predictions, recall stays relatively within the same range, as not much more than half of the data points per class are correctly identified. To assign a score to this relationship between precision and recall, we compute the F1 scores to get an idea of the average performance quality per class, since the overall mean accuracy (0.688) is skewed by the noncontacts, which, partially due to the high number, are well predicted. Additionally, Matthew's correlation coefficient is low, at 0.154, indicating that the predicted results are close to what we'd expect if the classifier predicted by random.

Model G30P1_1000_1

REAL

PREDICTED	True Non-contact (0)	True Interprotein (1)	True Intraprotein (2)	Precision	Recall	Fscore
Pred. 0	225494	128	3931	0.982	0.693	0.812
Pred.1	28754	212	1666	0.007	0.479	0.014
Pred. 2	71136	102	7770	0.098	0.581	0.168
Totals:	*325384*	*442*	*13367*	Mean accuracy:		0.688
				Matthews Correlation:		0.154

Figure 8: **Sample Model Predictive Metrics**: Shown for a model selected to be representative of the overall performance of all models. Precision and recall tend to be highest for noncontacts and lowest for interprotein contacts. Across all classes, recall averages around 0.5. The model is not much better than random predictions, as the low Matthews correlation coefficient indicates, and the mean accuracy averages around 0.6.

From the overall performance of our models, it is evident that the current setup of our classifier is largely unable to reliably differentiate between inter- and intraprotein contacts, and we are therefore unable to achieve our ultimate goal. In fact, the classifier also does not reliably differentiate between noncontacts and contacts, as the moderately high recall values for noncontact prediction indicate. However, within these limitations, we observe that there are improvements brought about mainly by variation of the size of the training and testing test as well as the number of trees, and the scale of such improvements we investigate in the following sections. Furthermore, several models from the *G50P1* and *G80P1* groups show unusually high precision and recall values for predicting interprotein contacts, indicating that there may be a hidden pattern in feature set that influences this predictive power. We discuss these improvements and anomalies in the following subsections, then analyse our best model in terms of performance and feature importance of the classifier.

4.2.2 Training and testing set size

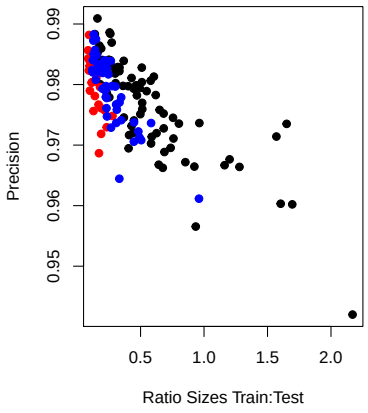
When we increase our training set size and decrease the testing set size by raising the number of groups from 3 to 80, we induce a change in the ratio of the sizes (measured in terms of the number of residue pairs) of training set to test set, the *train* : *test* ratio, from approximately 0.01 to 0.6. If we view this same size change by total number of proteins contributing their residue pairs to the training and testing sets, this is a change of approximately 111 : 54 proteins to 162 : 3 proteins. Due to the input data balancing step before we train our classifiers, we generally have less than half the number of datapoints in our training set as opposed to the test set. Only eight models in the *G80P1* category have equal or up to twice as many datapoints in their training set as opposed to the test set. The effects of increasing the training set size and reducing the testing set size on the precision and recall of our predictions is best visualised by the models built with CV groupings *G30P1*, *G50P1*, *G80P1*, due to the higher number of models compared to *G3P1* or *G5P1*, and consequently more data points. In Fig. 9 precision is plotted against the *train* : *test* ratio for each class (a - c), with black points representing *G80P1* models, blue points representing *G50P1* models, and red points representing *G30P1* models. Analogously, Fig. 10 shows recall plot-

ted against the *train* : *test* ratio for all three classes, with the exact same colorscheme as in Fig. 9.

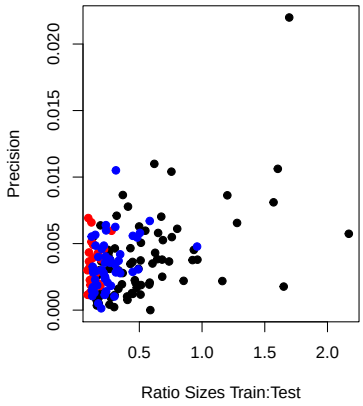
We would expect that training on more data improves the performance, but this cannot be concluded from the patterns our precision and recall values exhibit. In fact, hardly any robust conclusion can be made, except that compared to recall, precision is more likely to be affected by increasing the *train* : *test* ratio, at least considering the spread of data points per class. While the precision of noncontact predictions appears to decrease with an increase in the *train* : *test* ratio, this is only a slight decrease. Similarly, the same applies to intraprotein predictions, but in the reverse direction. Precision for interprotein contact prediction is likely to be unafffected. Each of our recall vs. *train* : *test* ratio plots (Fig. 10) show very few correlated effects.

Figure 9: **Comparisons of Precision vs. Ratios of Training to Testing Size**: Colour-coding of points is as follows: red - *G30P1_1000* models, blue - *G50P1_1000* models, black - *G30P1_1000* models. Precision for noncontact and intraprotein contact prediction is likely affected by an increase in the *train* : *test* ratio, but the effect is inconclusive given the number of data points and the range over which the ratio spans.

(a) Noncontact



(b) Interprotein



(c) Intraprotein

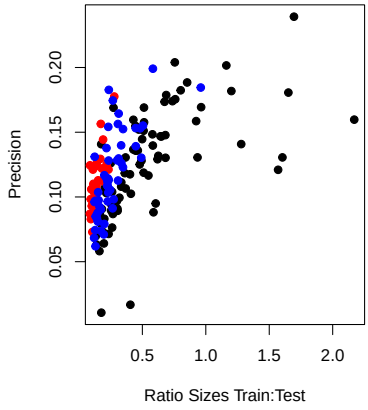
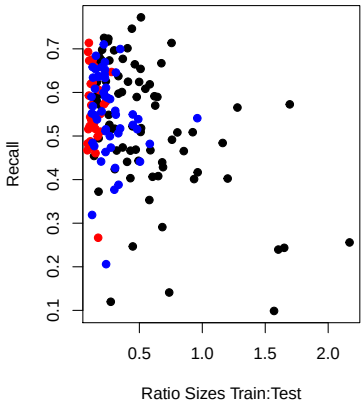
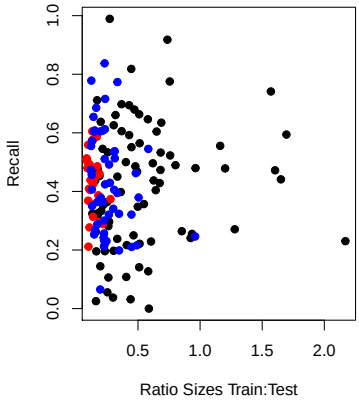


Figure 10: **Comparisons of Recall vs. Ratios of Training to Testing Size**: (refer to colour-coding in Fig. 9) Recall values do not show significant changes due to increasing the *train* : *test* ratio for any of the classes.

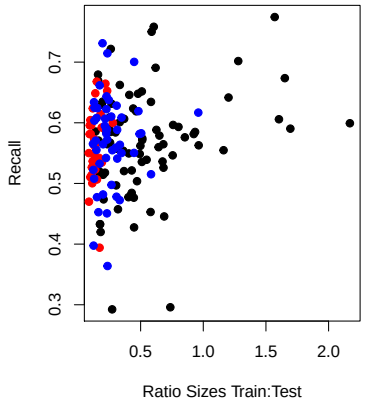
(a) Noncontact



(b) Interprotein



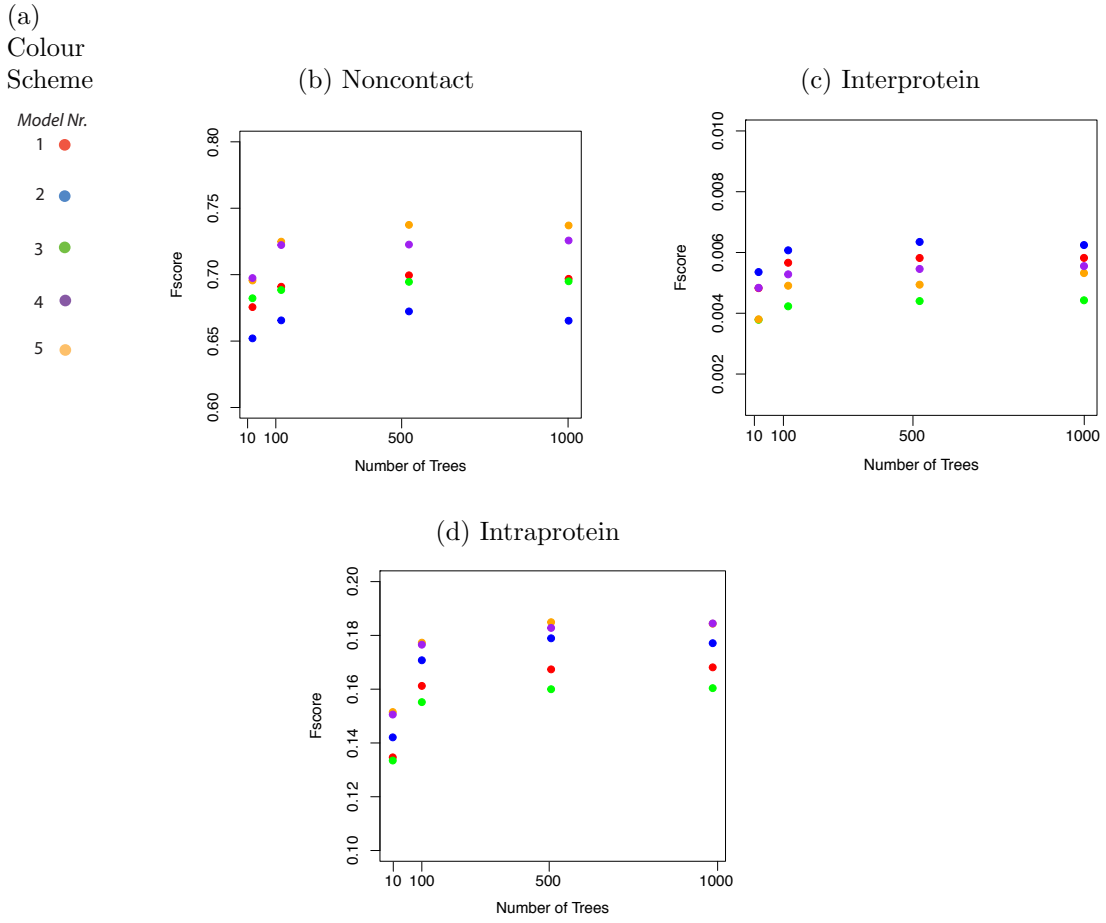
(c) Intraprotein



4.2.3 Number of random forest trees

Increasing the number of trees in our random forest improves predictive power, but not drastically. We measure the effect of increasing the number of trees through evaluating the change in performance metrics of 12 models from the *G3P1* and 20 from *G5P1* category, based on random forests with 10, 100, 500, and 1000 trees. Across all models, the largest improvement in precision, recall, fscore, and overall accuracy occurs when using a 100-tree random forest predictor instead of 10 trees. Models built on 500 trees improve slightly on those built with 100 trees, but their performance is overall rather indistinguishable from models with 1000 trees, and is in some cases even lower. Fig. 11 shows the F1 score of each class for all 20 models from the *G5P1* category (very comparable to those of the *G3P1* category), plotted against the number of trees used in the model. The colouring of the dots relate to the particular model number, meaning that dots with the same colour result from models trained on the same training set, and tested on the same testing set. We see that the largest improvement occurs in predicting intraprotein contacts (roughly an increase in Fscore of 0.03, or a gain of 6000 correctly predicted intraprotein contacts). For the other two classes, improvement in Fscore is minimal. We can, in fact, see that prediction is actually not the best with 1000-tree random forests, as most models achieve roughly equal Fscores and often also lower scores with 1000 trees as compared to 500 trees, but with our current number of models, no solid pattern can be derived. We choose, based on these results, to build random forests with a standard 1000 trees, as we have done with the *G10P1, G30P1, G50P1*, and *G80P1* groups.

Figure 11: **Comparisons of F1 score to Number of Trees for *G5P1***: We plot the F1 scores each of the five models for each of the G5P1 random forest classifiers built on 10, 100, 500, and 1000 trees to visualise average performance improvement while the number of trees varies, and the training/testing sets (per model) are held constant. As expected, the improvement is largest transitioning from 10 to 100 trees, but plateaus out between 100 - 1000 trees. We would need more data points to verify these patterns.



4.2.4 Prediction performance of the highest performing models

Our highest performing models are selected based on best overall predictive power for each class, and also for our particular class of interest, the interprotein contacts, under the condition that the testing set contains five or more proteins. We set this minimum testing set size condition when comparing our models because the models trained on large sets and tested on only up to three proteins (mainly from the *G80P1* group) show erratic predictive patterns for the different classes, sometimes correctly predicting almost all interprotein contacts (high recall for this class) but barely any of the other classes (low recall for noncontacts and intraprotein contacts, and low precision for interprotein contacts), a trait which we attribute to classification instabilities brought about by a small test set size.

The best overall model, with precision and recall values within the top five range for each class, compared across all CV groupings, is *G30P1_1000_4*, with the performance metrics listed in Table 3. If we compare this model to that of Fig. 8, which was chosen as our average-performance model, there is not a big difference in performance metrics, indicating that even our so-called 'best

overall' model does not significantly improve on predictions for all classes. The mean accuracy of this model is 0.643, which matches the average. The Matthews correlation coefficient computed for this model is 0.1963, indicating that the model prediction is also relatively close to random, as was the case with the correlation of 0.154 from model *G30P1_1000_1* in Fig. 8.

Table 3: **G30P1 Model 4 Performance Metrics**

Class	**Precision**	**Recall**	**Fscore**
Noncontact	0.9748	0.64686	0.7777
Interprotein contact	0.0059	0.3737	0.0117
Intraprotein contact	0.1176	0.59965	0.2740

Our highest performing model for predicting interprotein contacts, the model with both highest recall and precision for this class (across all CV groupings, given the minimum test set size condition mentioned above) is *G50P1_1000_8*, with metrics presented in Table. 4. Once again, the mean accuracy is not above average for this model, and the Matthews correlation coefficient is just slightly above the random case, at 0.124. With a high recall for interprotein contacts, the model was able to correctly identify almost all (221/264) interprotein contacts, but with a relatively low precision for the same class and notably lower recall values for the other two classes, meaning that the model tended to predict pairs as being interprotein contacts. It is dubious then, whether this model was particularly good at predicting interprotein contacts, or just had an overall bias towards predicting everything as being an interprotein contact, and the latter option seems more likely. In fact, in all models of the *G30P1*, *G50P1*, *G80P1* groupings that exhibited unusually high recall for interprotein contacts (> 0.7), we find a slight decrease in precision of this particular class, and more notable decrease in the recall of the other two classes.

Table 4: **G50P1 Model 8 Performance Metrics**

Class	**Precision**	**Recall**	**Fscore**
Noncontact	0.9761	0.4630	0.6283
Interprotein contact	0.0040	0.8371	0.0080
Intraprotein contact	0.1542	0.2507	0.2297

The main pitfall of the current way of identifying our highest performing models relates to prioritising one performance metric over the other, which may not truly be the best to achieve our research objective. For example, currently the best model for prediction of interprotein contacts prioritised recall for this class, over any other metric, because we want to find a model which can as accurately as possible, pick out all the interprotein contacts. However, as we immediately see in comparing the lowered recalls of the other classes, as well as the low precision of the interprotein predictions, this best model is far from ideal, and we currently have no sound explanation for this tendency to predict all residue pairs as interprotein contacts. In fact, we want to prioritise recall and precision for contacts, and then for interprotein contacts in particular. For the overall model, the selection procedure is more straightforward, as a simple (descending) sorting of models based on all precision, and recall values easily returns the same models in a top 5 or top 10 set of models for a particular CV grouping. However, as briefly mentioned above, whether this is justifiably

the best overall model given that the predictive performance is so similar to the average model can also reasonably be challenged. Based on the minimal improvements brought about by tuning parameters such as the $train : test$ ratios or number of trees, perhaps the concept 'best' does not exist in this problem, and average performance is what we're stuck with.

4.2.5 Feature importance of high performing models

To better understand how much of an effect our chosen set of features have on the prediction of our classifier models, particularly those of our higher performing models, we visualise the feature importance values for the *G30P1_1000_4* model with a barplot in Fig 12. The colouring of the bars corresponds to the feature set from which it belongs (labelled with the column number from the input datamatrix), with the colour scheme corresponding to the legend. All importance values are relatively low, and the coevolutionary value has the most effect on predictive power of the classifier. The high influence of the coevolutionary value is not limited to *G30P1_1000_4* - in fact, it is the feature with the highest importance across all models. In order of decreasing importance after the coevolutionary value - the predicted solvent accessibilities of each residue in a pair and the conservation of the residues are most influential, followed by secondary structure predictions, and finally, the amino acid identity percentages. Although importance values are globally very low, each feature set does tend to group together in the importance ranking.

To compare the rankings of feature importance for our two high-performing models, we compute the correlation between their feature importance values, which is extremely high (0.9993). When we repeat this for randomly selected models across all CV groupings, we find that the correlation stays high (> 0.95) between their feature importance values, indicating that the overall ranking of features is stable, and although specific features may shuffle orders, they mainly remain within their defined block (colored section corresponding to range of importance values of the Fig. 12 barplot) with other related features.

If the feature importance were at all relatable to biological importance in determining whether a residue pair interacts or not, and if interacting - as an interprotein or intraprotein contact, based on the consistency of feature importance ordering across all of our models, we would have reason to say that coevolutionary value, solvent accessibility, and secondary structure have the most influence in this classification decision, and in this order. However, considering the overall performance of our classifier models, as well as the generally low feature importance score differences relative to each other, making such a conclusion is not possible. After all, feature importance is a measure based on how much error is introduced into the classifier performance after a permutation of the feature at hand, where low error means low feature importance, and vice versa. Particularly for random forests implemented through SciKit Learn, the feature importance represents an average over the decrease in Gini impurity values (how stably a node is classified to differing classes, where a lower impurity value is a higher accuracy to classify the node to a certain class) introduced by a feature [22]. Certain characteristics inherent in our feature set can highly influence this importance. Therefore, to evaluate with more certainty which features are truly important for our model and their relation to each other, we need to implement a more rigorous quality check of our feature set characteristics - by which we mainly mean an identification of highly correlated features. For example, since a random forest randomly selects a set of features per split in the decision tree, a chance selection of one of, say, two highly correlated features, may lead to an increase of the importance of the selected feature but a decrease in the reported importance of

the other correlated features, which is misleading because in reality their importance should be relatively equal. Additionally, to test the reported importances of our features, we would need to regrow random forests based on holding out highly important features to evaluate the classification performance. These are all crucial future steps in obtaining the correct evaluations of feature importance and can have a significant influence on optimising our model to truly predict based on biologically-sound, informative features.

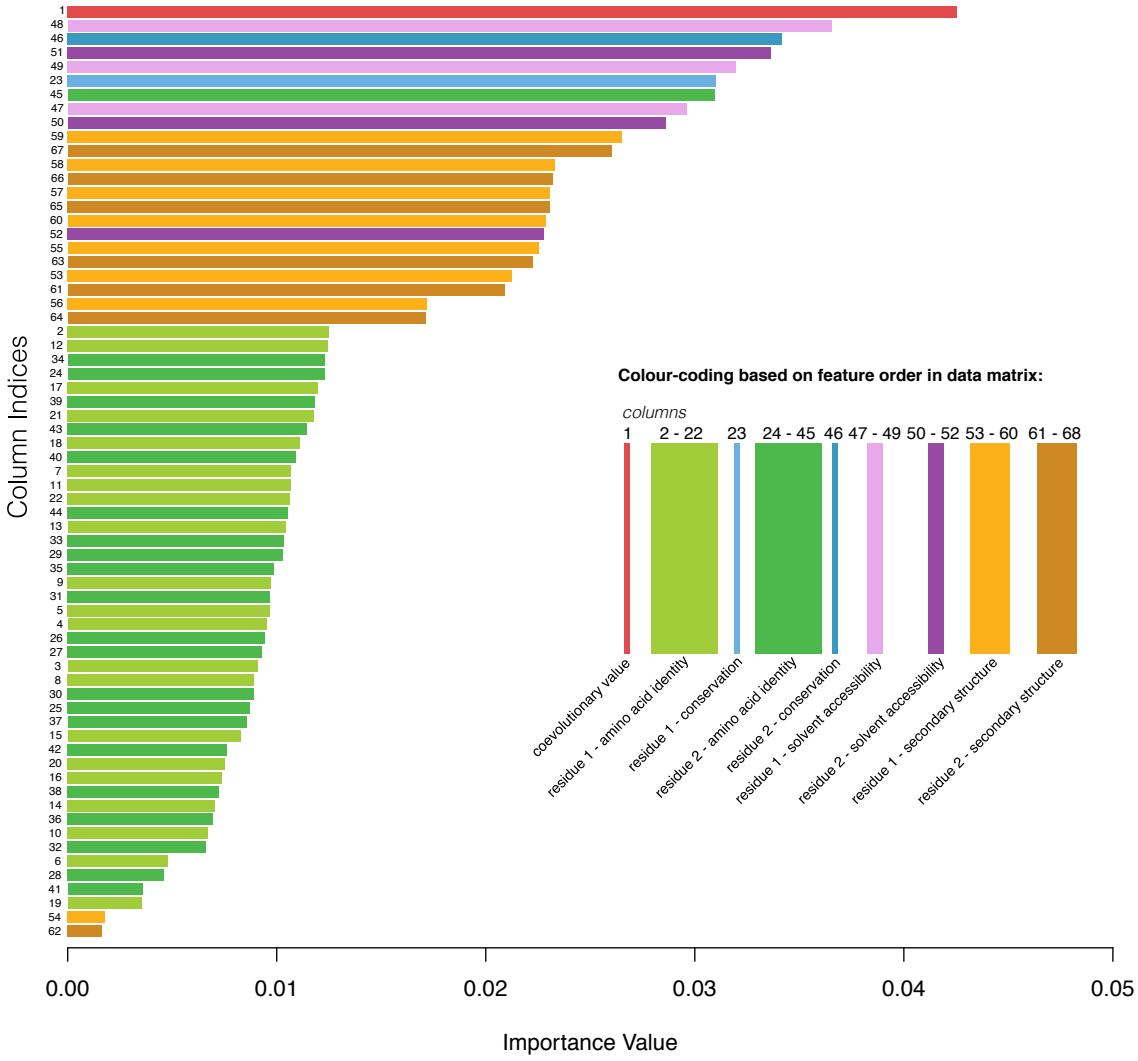


Figure 12: **Feature importance plot of *G30P1_1000_4***: Colour-coded bar plot of feature importances for our best overall model, with colours corresponding to a particular feature as given by the order it is embedded within the input data matrix to the classifier. Highest feature importance goes to the coevolutionary value, followed by predicted solvent accessibility and conservation for both residues of a pair, predicted secondary structure for both residues, and the percentage identities of the MSA positions of both residues. We see a clear division between the types of contacts.

4.3 Preliminary Patterns in Features and Predictions

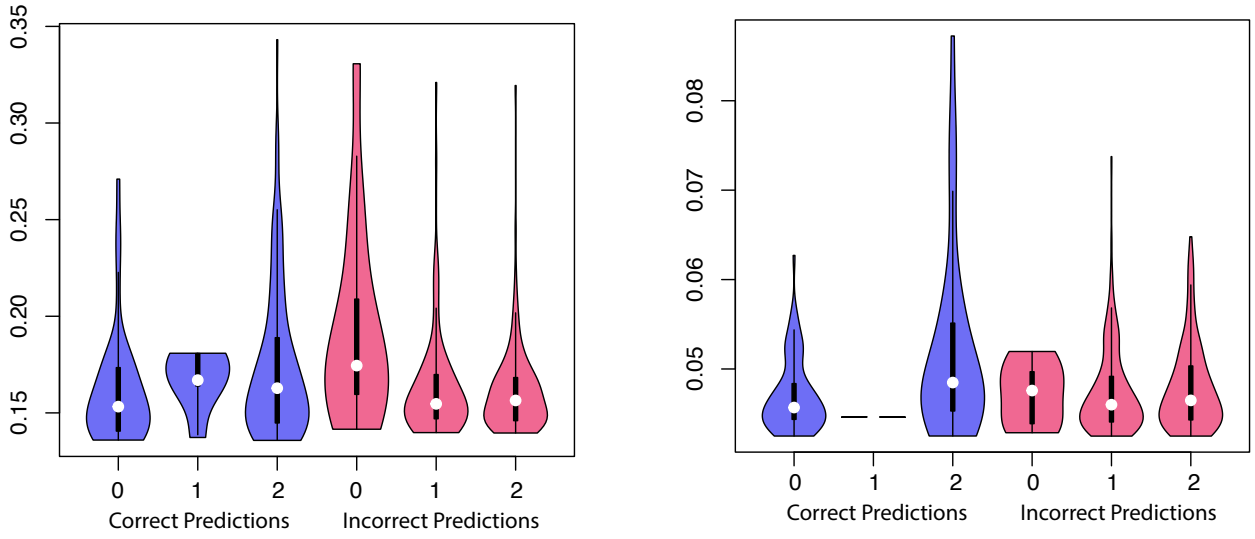
Given that the features with the highest feature importance have the most impact on the classification, we investigate whether we can deduce a predictive pattern associated with residue pairs

with high or low values of a certain feature, based on a biological understanding of how the feature should relate to all three types of classes. For example, starting with the feature with highest importance, the coevolutionary value, we hypothesise that - since a higher coevolutionary signal between two residues translates to a stronger mutational interdependency between the residues, we expect residue pairs with high coevolutionary signal to be interacting, and therefore be in contact. Whether the exact type of contact is interprotein or intraprotein is difficult to judge, because to do this we would need to know the common mutational rates of interacting residues within a protein and between two proteins, compared to how conserved these interacting residues generally are, since coevolutionary methods do not readily pick up signals from conserved contacts.

We take a subset of our classifier predictions corresponding to the residue pairs with the top 1000 highest coevolutionary values, and plot the predicted classes as violin plots in Fig. 13. For each plot, the leftmost three curves (blue) show the number of correctly predicted classes (0 - noncontact, 1 - interprotein, 2 - intraprotein), and the remaining three (red) show the incorrectly predicted classes. We do this for both of our highest performing models, *G30P1_1000_4* and *G50P1_1000_8*. Fig. 13 (a) and (b) show that the highest coevolutionary values generally correspond to correct intraprotein contacts, but that the distribution of pairs is most concentrated still at the lower end of the coevolutionary values, around 0.15. There is a slight tendency in (a) to (incorrectly) classify residue pairs with high coevolutionary signal as noncontacts, compared to the other two classes. Surprisingly, in the subset of residue pairs with the thousand highest coevolutionary values for the *G50P1_1000_8* model, there is only one interprotein contact. If we extend this analysis to other models from other CV groupings, we see a similar pattern of a few intraprotein contacts identifiable by having the highest coevolutionary values, whereas the bulk of intraprotein contacts do have relatively high coevolutionary values compared to the rest of the residue pairs, but generally lower within this range. The next group present with high co-evolutionary values are the non noncontacts, and no interprotein contact achieves the same range of coevolutionary value as the other classes, so violin plots of the three class predictions for most models are similar to Fig. 13 (a). Based on this analysis, we cannot conclude that high coevolutionary value corresponds to a higher certainty of a residue pair to be in contact.

If we were to extend this analysis to the second highest-ranking feature in terms of importance value (label nr. 48), corresponding to the probability of the first residue in a pair to be exposed (has a relative solvent accessibility, $pACC$ of between 40%-100%). High-scoring residue pairs are most likely to be exposed in the solvent, which could indicate that they are not in contact, or interacting between proteins as an interprotein contact. However, for the highest-scoring residue pairs for this feature (3000 in total, effectively covering the entire spectrum of nonzero probabilities for this feature), there are no correctly predicted interprotein contacts, and the majority of residue pairs correctly predicted are fairly evenly distributed between noncontacts and intraprotein contacts, whereas incorrectly predicted residues have almost no bearing as to which class they belong. Once again, not much can be said about the predictive patterns for this feature.

This particular type of prediction analysis, though potentially able to identify very broad patterns in prediction, is limited in its usefulness, as the conclusions we can make are dubious. For instance, the number of data points with a high feature value we would select (1000, 3000), is relatively arbitrary, and we cannot know beforehand that this number will properly represent the prediction patterns associated with the high feature value. Instead, we need to take into consideration the full range of the feature value, and extract any predictive patterns from there. Even if a pattern were



(a) G30P1 Model 4

(b) G50P1 Model 8

Figure 13: **Predictions Corresponding to High Coevolutionary Value**: Violin plots of our two best models showing the spread and density of correct (blue) and incorrect (red) predicted classes for the residue pairs with the highest coevolutionary values (top 1000 only). Class labels are: 0 - noncontact, 1 - interprotein contact, 2 - intraprotein contact. From these plots, higher coevolutionary values do not necessarily indicate higher instances of features.

identifiable, the cause would still be uncertain. Whereas we know what a high coevolutionary value should represent, the classifier does not, and the data do not affirm this expectation potentially because the sensitivity of our coevolutionary method was low, and therefore the values given by our CCMpred contact prediction could be inaccurate. Another possibility is that the structural extraction of the classes was erroneous, but this is more or less akin to opening a Pandora's box if it were true, at least for the integrity of our classification validation. Finally, isolating one feature, and looking at residue pairs scoring in either extremity of the range of that feature's value, is inherently excluding lots of information which the classifier used to make the prediction in the first place. Further analysis of the prediction patterns is very warranted, however, since this is where the connection between biology and classification can be strengthened. For instance, some directions this could take would be to investigate if low solvent accessibility of residue pairs corresponds to higher number of contacts, but specifically intraprotein contacts, as those tend to be buried within a protein. Additionally, we could investigate whether a highly conserved residue pair indeed has very little to no coevolutionary signal, and whether these residues are then incorrectly classified as noncontacts.

5 Conclusions and future directions

With the current setup of our random forest classifier, we cannot achieve the initial goal of reliably differentiating between an inter and intraprotein contact in a homodimeric complex, as most of our classifications achieve low recall and precision for both contact classes, compared to the precision and recall values for noncontacts. Variation of the training and testing set sizes and the number of trees of our random forest does not significantly improve our classifier performance. Furthermore, we do not observe consistent predictive patterns from subsets of our data known to fall within a certain range of feature value - for example, high coevolutionary values do not correspond to higher instances of true or predicted interprotein contacts, but perhaps do show a slight tendency towards intraprotein contacts over noncontacts.

There are numerous factors in our methodology design and random forest building steps that may all or in part contribute to these predictive results, therefore the most immediate future direction is to return to the specific steps of our experimental design and to isolate and individually test how much of an impact they may have on the resulting predictions. One crucial step in evaluating the classifier performance is to compute training error, to see if the classifier was overfitting the data.

The first potential factor relates to the size of our final dataset and diversity of sequences present within it. 165 proteins, despite the high number of residue pairs they may contribute as input to the classifier, is very likely neither large enough of a set, nor diverse enough, as our control on this diversity is completely reliant on the UCLUST algorithm. We should thus experiment with other sequence clustering methods, and a wider range of similarity thresholds.

Another influential factor relates to the calculation of coevolutionary values - the fact that they are universally very low is perhaps due to the low N_{eff} present in our MSAs. The practical solution is to increase the dataset by increasing the 10% retention rate of our centroid sequences, so we get a larger set to input to Jackhmmer and potentially a wider range of returned sequence set sizes. We could also implement a more robust construction of a protein family for our MSAs, instead of just one Jackhmmer query search controlled by an inclusion threshold.

Once we optimise our dataset, and increase the quality of our MSAs (and therefore our coevolutionary values), we would then need to vary parameters for our random forest classifier. Due to the time restraints for this project, we kept most parameters at defaults, but trying out potentially different variables - such as the number of features randomly sampled. The feature set also needs thorough editing or at least a more thorough analysis, to better understand our input. Other classifier algorithms could also be tested.

Finally, it could be the case that the problem is just not possible to be solved, that there is no pattern to differentiate between inter and intraprotein contacts based on sequence information, regardless of any combination of features or tuning in classifier parameters, or regardless of the type of machine learning algorithm. This is yet to be determined, though, and therefore our efforts still should proceed.

References

[1] L. Burger and E. van Nimwegen. Disentangling direct from indirect co-evolution of residues in protein alignments. *PLOS Computational Biology*, 2010.

[2] D. de Juan, F. Pazos, and A. Valencia. Emerging methods in protein co-evolution. *Nature Reviews Genetics*, 14(4):249–261, April 2013.

[3] R. Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19):2460–2461, 2010.

[4] M. Ekeberg, C. Lovkvist, Y. Lan, M. Weigt, and E. Aurell. Improved contact prediction in proteins: Using pseudolikelihoods to infer potts models. *Physical Review E*, 87, 2013.

[5] O. Emanuelsson, H. Nielsen, S. Brunak, and G. von Heijne. Prediction subcellular localization of proteins based on their n-terminal amino acid sequence. *Journal of Molecular Biology*, 300(4):1005–16, July 2000.

[6] I. Ezkurdia, O. Grana, J. M. G. Izarzugaza, and M. L. Tress. Assessment of domain boundary predictions and the prediction of intramolecular contacts in casp8. *Proteins*, 77:196–209, 2009.

[7] R. D. Finn, J. Clements, and S. R. Eddy. Hmmer web server: interactive sequence similarity searching. *Nucleic Acids Research*, 39:W29–W37, July 2011.

[8] U. Gobel, C. Sander, R. Schneider, and A. Valencia. Correlated mutations and residue contacts in proteins. *Proteins: Structure, Function, and Genetics*, 18:309–317, 1994.

[9] J. Gorodkin. Comparing two k-category assignments by a k-category correlation coefficient. *Computational Biology and Chemistry*, 28(5):367–374, 2004.

[10] Q. Hou, B. E. Dutilh, M. A. Huynen, J. Heringa, and K. A. Feenstra. Sequence specificity between interacting and non-interacting homologs identifies interface residues - a homodimer and monomer use case. *BMC Bioinformatics*, 16(325), 2015.

[11] Q. Hou, P. F. D. Geest, W. F. Franken, J. Heringa, and K. Feenstra. Seeing the trees through the forest: sequence-based homo- and heteromeric protein-protein interaction sites prediction using random forest. *Bioinformatics*, 33(10):1479–1487, January 2017.

[12] E. B. Institute. Protein interfaces, surfaces and assemblies.

[13] D. Juan, F. Pazos, and A. Valencia. Co-evolution and co-adaptation in protein networks. *FEBS Letters*, 582(8):1225–1230, 2008.

[14] M. Kallberg, H. Wang, S. Wang, J. Peng, Z. Wand, H. Lu, and J. Xu. Template-based protein structure modeling using the raptorx web server. *Nature Protocols*, 7:1511–1522, 2012.

[15] A. Lapedes, B. Giraud, L. LonChang, and G. Stormo. Correlated mutations in models of protein sequences: phylogenetic and structural effects. *ISM Lecture Notes*, 1999.

[16] N. J. Marianayagam, M. Sunde, and J. M. Matthews. The power of two: protein dimerisation in biology. *Trends in Biochemical Sciences*, 29(11), 2004.

[17] D. Marks, L. Colwell, R. Sheridan, T. Hopf, A. Pagnani, R. Zecchina, and C. Sander. Protein 3d structure computed from evolutionary sequence variation. *PLoS One*, 6(12), 2011.

[18] F. Morcos, T. Hwa, J. N. Onuchic, and M. Weigt. *Direct coupling analysis for protein contact prediction*, volume 1137 of *Protein Structure Prediction*, chapter 5. Springer Science, 2014.

[19] F. Morcos, A. Pagnani, B. Lunt, A. Bertolino, D. S. Marks, C. Sander, R. Zecchina, J. N. Onuchic, T. Hwa, and M. Weigt. Direct coupling analysis of residue coevolution captures native contacts across many protein families. *PNAS*, 108(49):E1293–E1301, 2011.

[20] J. Moult, K. Fidelis, A. Kryshtafovych, T. Schwede, and A. Tramontano. Critical assessmen of methods of protein structure prediction (casp) progress and new directions in round xi. *Proteins*, 84(1):4–14, 2016.

[21] D. Ochoa and F. Pazos. Practical aspects of protein co-evolution. *Frontiers in Cell and Developmental Biology*, 2(14), 2014.

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[23] L. Sael, M. Chitale, and D. Kihara. Structure- and sequence-based function prediction for non-homologous proteins. *J Struct Funct Genomics*, 13(2):111–123, June 2013.

[24] S. Scheuermann, B. Hambsch, L. Hesse, J. Stumm, C. Schmidt, D. Beher, T. A. Bayer, K. Beyreuther, and G. Multhaup. Homodimerization of amyloid precursor protein and its implication in the amyloidogenic pathway of alzheimer's disease. *Journal of Biological Chemistry*, 7(276), 2001.

[25] S. Seemayer, M. Gruber, and J. Soding. Ccmpred - fast and precise prediction of protein residue-residue contacts from correlation mutations. *Bioinformatics*, 30(21):3128–3130, November 2014.

[26] F. Sievers, A. Wilm, D. Dineen, T. Gibson, K. Karplus, W. Li, R. Lopez, H. McWilliam, M. Remmert, J. Soding, J. Thompson, and D. Higgens. Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular Systems Biology*, 7(539), 2011.

[27] G. Uguzzoni, S. J. Lovis, F. Oteri, A. Schug, H. Szurment, and M. Weigt. Large-scale identification of coevolution signals across homo-oligomeric protein interfaces by direct coupling analysis. *PNAS*, 114(13):E2662–E2671, March 2017.

[28] S. Vorberg, S. Seemayer, and J. Soding. Synthetic protein alignments by ccmgen quantify noise in residue-residue contact prediction. *PLOS Computational Biology*, 2018.

[29] Z. Wang, F. Zhao, J. Peng, and J. Xu. Protein 8-class secondary structure prediction using conditional neural fields. *Proteomics*, 11(19):3786–3792, 2011.

[30] Q. Wuyun, W. Zhang, Z. Peng, and J. Yang. A large-scale comparative assessment of methods for residue-residue contact prediction. *Briefings in Bioinformatics*, 19(2):219–230, March 2018.