

HMI-PRED: Design and Implementation of a Webserver for Host-Microbe Interactions Prediction

by

Asma Hakouz

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computer Science and Engineering



**KOÇ
ÜNİVERSİTESİ**

August 1, 2019

**HMI-PRED: Design and Implementation of a Webserver
for Host-Microbe Interactions Prediction**

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

Asma Omar Hakouz

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Prof. Attila Gürsoy (Advisor)

Prof. Özlem Keskin [Co-Advisor]

Prof. Deniz Yüret

Assoc. Prof. Nurcan Tunçbağ

Date: _____



ABSTRACT

HMI-PRED: Design and Implementation of a Webserver for Host-Microbe Interactions Prediction

Asma Hakouz

Master of Science in Computer Science and Engineering

August 1, 2019

Microbes, commensals and pathogens, control numerous functions in host cells. They can alter host signaling and modulate immune surveillance by interacting with host proteins through mimicking the host protein-protein interfaces. To shed light on the contribution of microbes to health and disease, it is vital to discern how microbial proteins rewire host signaling and through which host proteins. Current host-microbe interaction data is a long way from complete, and experimental methods for large-scale identification of HMIs is challenging. Most of the currently available methods for HMI prediction are based on global sequences or structural similarity. On the other hand, there is only one available webserver for these methods, which limits the usage of these tools by the researches and scientists. To address both issues, we developed Host-Microbe Interaction PREDictor (HMI-PRED), a user-friendly webserver for template-based structural prediction of protein-protein interactions (PPIs) between host (i.e., human) and any microbial species, including bacteria, viruses, fungi, and protozoa. HMI-PRED relies on “interface mimicry” through which the microbial proteins hijack host binding surfaces. HMI-PRED server was optimized by clustering the template interface set, which is used as bases for predictions. Given the 3D structure of a microbial protein of interest, HMI-PRED will return detailed 3D structural models of potential host-microbe interaction (HMI) complexes, the list of host endogenous and exogenous PPIs that can be disrupted by the microbe protein, and the functional annotation and tissue expression of the microbe-targeted host proteins. Also, the server offers 3D visualizations of the predicted structures with highlighted contact residues, as well as visualization of the predicted structure superimposed on the original template interface. The server also allows users to upload homology models of microbial proteins. The prediction results are stored in a repository for the community access. Users can examine and search the accumulated results. HMI-PRED is available for public at <https://interactome.ku.edu.tr/hmi>. We also introduce a ranking method for the predicted interactions using a deep learning model based on 3D structures which is trained to identify valid protein-protein interfaces.

ÖZETÇE

HMI-PRED: Konak-Mikrop Protein Etkileşiminin Tahmini için Web

Sunucusu Tasarımı ve Geliştirilmesi

Asma Hakouz

Bilgisayar Bilimleri ve Mühendisliği, Yüksek Lisans

1 Ağustos 2019

Mikroplar konak hücrelerde sayısız işlevi kontrol eder. Mikroplar, konakçı hücredeki sinyal mekanizmalarını değiştirebilir ve protein-protein ara yüzlerini taklit edip, konakçı proteinlerle etkileşime girerek bağışıklık sisteminin denetlenmesini modüle edebilir. Mikropların sağlığa ve hastalığa olan katkısına ışık tutmak için, mikrobiyal proteinlerin konak hücrelerdeki sinyal mekanizmalarını nasıl değiştirdiğini ve bu durumun hangi konak proteinlerini içerdiğini anlamak çok önemlidir. Mevcut konak-mikrop proteinlerinin etkileşimi hakkındaki veriler henüz çok yetersizdir ve bu etkileşimlerin tanımlanmasında deneysel yöntemlerin kullanımı oldukça zordur. Konak-mikrop etkileşimlerinin tahmini için kullanılan mevcut yöntemlerin çoğu, global hizalama veya yapısal benzerliğe dayanmaktadır. Öte yandan, bu yöntemlerin araştırmacılar tarafından kullanımını sınırlayan sebeplerden biri de bu amaç için kullanılan yalnızca bir web sunucusu olmasıdır. Her iki sorunu da çözmek için, kullanıcı dostu bir web sunucusu olan Host-Microbe Interaction PREDictor (HMI-PRED)'i geliştirdik. Bu web sunucusu konakçı (insan) ve herhangi bir mikrobiyal tür (bakteri, virüs, mantar ve protozoa) arasındaki protein-protein etkileşimlerinin şablon bazlı yapı tahmini için kullanılabilir. HMI-PRED'in çalışma prensibi, mikrobiyal proteinlerin konakçı proteinlerin bağlanma yüzeylerini taklit etmesine dayanmaktadır. HMI-PRED sunucusu, tahminler için temel olarak kullanılan şablon ara yüz seti kümelenecek optimize edildi. HMI-PRED, ilgilenilen bir mikrobiyal proteinin 3B yapısı göz önüne alındığında, potansiyel konak-mikrop etkileşiminin detaylı 3B yapısal modellerini, mikrop proteini tarafından bozulabilen konak endojen ve eksojen protein-protein etkileşimlerinin listesini, mikrop hedefli konakçı proteinlerin fonksiyonel açıklamasını ve bu proteinlerin ekspresyonunun gerçekleştiği bütün dokuların listesini vermektedir. Buna ek olarak, sunucu, vurgulanmış etkileşim kalıntılarını, öngörülen yapıların 3B görselleştirmelerini ayrıca orijinal şablon ara yüzü ile üst üste gösterilen tahmini yapının şeklini sunmaktadır. Sunucu ayrıca, kullanıcıların mikrobiyal proteinlerin homoloji modellerini de girdi olarak kullanılmasına olanak tanır. Tahmini sonuçlar, kullanıcıların gelecek erişimleri için bir arşivde depolanır ve bu sayede kullanıcılar bu arşivdeki sonuçları inceleyip arayabilirler. HMI-PRED, <https://interactome.ku.edu.tr/hmi> adresinde halka açıktır. Ayrıca, öngörülen protein-protein ara yüzlerinin biyolojik geçerliliğini tanımlamak için derin öğrenme modeli kullanarak eğitilmiş bir sıralama yöntemi de tarafımızdan sunulmaktadır.

ACKNOWLEDGEMENTS

First and foremost, I offer my sincerest gratitude to my advisors, Attila Gürsoy and Özlem Keskin for giving me the chance of being a member of their research group, and providing a productive and comfortable research environment.

I cannot express enough thanks to Emine Güven-Maiorov, Ruth Nussinov, and Chung-Jung Tsai from NIH for their guidance and sharing their endless experience with me throughout my journey in Koç.

I would also like to extend my thanks to my thesis committee, Deniz Yüret and Nurcan Tunçbağ for offering their valuable time and expert opinion.

I am endlessly grateful for all members of COSBI family and all my friends in Koç. The completion of this project could not have been accomplished without their continuous and unconditional support.

Finally, to my caring, loving, and supportive family: my deepest gratitude. I am extremely grateful for their love, prayers, caring and sacrifices for educating and preparing me for my future.

TABLE OF CONTENTS

List of Tables	x
List of Figures	xi
Abbreviations	xiii
Chapter 1: Introduction	1
Chapter 2: Literature Review	3
2.1 Host-Microbe Interactions	3
2.2 Protein-Protein Interaction interfaces	4
2.3 Template-Based PPI prediction	5
2.4 Ranking PPI Interfaces with Deep Learning.....	6
2.4.1 PPI4DOCK dataset	7
2.4.2 DOCKGROUND Dataset.....	9
2.4.3 PIFACE Dataset.....	9
Chapter 3: Materials and Methods	11
3.1 HMI-PRED protocol.....	11
3.1.1 Preprocessing and Surface Extraction	13
3.1.2 Structural Alignment	13
3.1.3 Transformation and Filtering	14
3.1.4 Flexible Refinement.....	15
3.1.5 HMI-PRED Output.....	15
3.2 Template Interface Set	17
3.2.1 Template Set Preparation.....	18
3.2.2 Clustering Heuristics	21
3.3 Predicted PPI Validation/Ranking with DeepInterface	27
3.3.1 Datasets	27
3.3.2 Model.....	28
3.4 HMI-PRED Backend Design and Implementation.....	29

3.4.1	Design Principles	30
3.4.2	Operating System and Server Setup	32
3.4.3	Job Scheduling	32
3.4.4	Database Design	32
3.5	HMI-PRED Frontend Design and Implementation	40
3.5.1	Responsive Design.....	40
3.5.2	3D structures visualization	41
Chapter 4: RESULTS AND DISCUSSION		43
4.1	HMI-PRED Webserver Usage	43
4.1.1	HMI-PRED Main Page	43
4.1.2	Predictions Page.....	45
4.1.3	Template Page	46
4.1.4	Downloadable Results Folder for Each Target.....	47
4.2	Accuracy of Structural Alignment and Scoring Function.....	49
4.3	Clustering Optimization.....	50
4.4	Case Studies	52
4.4.1	Pathogenic bacteria protein with resolved 3D structure as an input..	52
4.4.2	Commensal bacteria protein with homology model as an input.....	52
4.5	DeepInterface Performance Analysis.....	54
4.5.1	Data Redundancy Removal	57
Conclusion		58
BIBLIOGRAPHY		59
Appendix A.....		66
Enumerating Maximal Cliques (All-Clique) Algorithms		66
Appendix B		72
Detailed Accuracy Test Results for Known HMIs in The Default template Set		72
Appendix C		Error! Bookmark not defined.

Case Study Results.....	Error! Bookmark not defined.
Appendix D.....	Error! Bookmark not defined.
Details of Known HMI's Incorrect Alignment	Error! Bookmark not defined.



LIST OF TABLES

Table 3.1: DeepInterface training, validation and testing datasets resources	27
Table 3.2: HMI-PRED Modules and their functionality	31
Table 3.3: List of HMI-PRED DB Tables	33
Table 3.4: Description of 'Job' Table in HMI-PRED DB	34
Table 3.5: Description of 'Jobs' Table	35
Table 3.6: Description of 'ip_addr' Table	35
Table 3.7: Description of 'alignment_results' table	36
Table 3.8: Description of 'target_info' Table	36
Table 3.9: Description of 'templates_hotspots' Table	36
Table 3.10: Description of 'templates' Table	37
Table 3.11: Description of 'template_rosetta_score' Table	37
Table 3.12: Description of 'protein_info' Table	38
Table 3.13: Description of the 'results' Table	39
Table 4.1: Comparison between clustering algorithms on the full template set.	52
Table 4.2: Evaluation of DeepInterface model	56
Table 4.3: Ranking of true interface among decoys of the corresponding complex.	56
Table 4.4: DeepInterface datasets after redundancy removal	57
Table 4.5: Evaluation of DeepInterface model after removing redundancy	57

LIST OF FIGURES

Figure 2.1: '1am4AD' PPI structure with highlighted interface and hotspot residues	5
Figure 3.1: Abstract Overview of HMI-PRED protocol.....	12
Figure 3.2: Detailed Overview of HMI-PRED Protocol Stages	12
Figure 3.3: Structural Alignment Step	14
Figure 3.4: HMI-PRED Results Page Details and Functionality.....	17
Figure 3.5: Functional annotation of microbe-targeted host proteins	17
Figure 3.6: Preparation of template interface set.	18
Figure 3.7 The frequency distribution of PPI groups size.	20
Figure 3.8: An example of the input and output of clustering algorithm.	22
Figure 3.9: Interface clustering steps.	23
Figure 3.10: Example of the execution of CLUSTER_V1 algorithm.	24
Figure 3.11: Pseudocode of CLUSTER_V2 algorithm.	25
Figure 3.12: Pseudocode for Picking Representatives in CLUSTER_V2	26
Figure 3.13: Example of the execution of CLUSTER_V2 algorithm.	26
Figure 3.14: DeepInterface input data format.	29
Figure 3.15: DeepInterface Model architecture	29
Figure 3.16: HMI-PRED webserver architecture	30
Figure 3.17: HMI-PRED Database Schema (Jobs-related data, and intermediate results)	33
Figure 3.18: HMI-PRED Database Schema (Results-related data).	34
Figure 3.19: MySQL function to calculate the predicted complex 'weight'	39
Figure 3.20: MySQL Functions to calculate Scores 1-3 for the predicted complexes. ..	40
Figure 3.21: Different 3D Visualization Options Offered by HMI-PRED.....	42
Figure 4.1: HMI-PRED main page	44
Figure 4.2: “Predictions” tab, a repository for previous results.	46
Figure 4.3: “Templates” tab.	47
Figure 4.4: Accuracy of Structural Alignment and Scoring Functions	49
Figure 4.5: Examples of multiprotein complexes in a crystal	50
Figure 4.6: Optimality Comparison of Clustering Heuristics	51
Figure 4.7: Case studies.	54
Figure 4.8: Sample distributions of DeepInterface scores for two complexes	55
Figure A.1: A graph which achieves Moon & Moser’s bound	66

Figure A.2: Bron-Kerbosch algorithm (version 1) Pseudocode	67
Figure A.3: An example of the execution of Bron-Kerbosch algorithm (v1).....	69
Figure A.4: Bron-Kerbosch algorithm (version 2) Pseudocode.	70
Figure A.5: Execution trees for Bron-Kerbosch v1(b) and v2 (c) on the graph in (a)....	71



ABBREVIATIONS

CAPRI	Critical Assessment of Predicted Interactions
Fnat	Fraction of native contacts
HMI-PRED	Host Microbe Interaction PREDictor
iRMSD	interface Root-Mean-Square Deviation
lRMSD	ligand Root-Mean-Square Deviation
PPI	Protein-Protein Interaction
PRISM	Protein Interactions by Structural Matching
RMSD	Root-Mean-Square Deviation
NPV	Negative Predictive Value

Chapter 1:

INTRODUCTION

Proteins seldom act on their own, instead they interact with each other to carry out their functions. Protein-Protein Interactions (PPI)s play vital roles in all living organisms. Microbial proteins can modulate host (e.g., Human) immune surveillance by exploiting complex mechanisms. One of these mechanisms is mimicking the host PPI interfaces, which can rewire cellular pathways by allowing microbes to compete with the host proteins. Limited Host-Microbe Interaction (HMI) data is because of the difficulty of large-scale experimental identification. Therefore, computational methods for HMI prediction became more needed. However, most of the available HMI predictors are based on global sequences or structural similarity. This rationale does not take into account that many microbial proteins do not have structural or sequence homologs in host (i.e., human) which makes it difficult to predict HMIs for every microbe with such methods.

In this work, we discuss the design, development and optimization of HMI-PRED, an interface-based HMI-prediction web server. To date, our method is the only one that relies solely on interface mimicry. The rationale is that even if proteins have different global structures and/or sequences, they might share common interface architectures. The server is expected to become a structural Host-Microbe Interactions (HMI) repository, which grows with user inputs and predictions and eventually will become a rich source for the study of HMI networks and the microbiome.

This thesis covers the prediction protocol followed by the server and explains optimization measures done to speed up and improve predictions such as structural clustering of the template interfaces set. In this thesis, we also suggest an additional ranking method for the server predictions using deep learning.

In chapter 2, we present background information and related work for HMI prediction and the deep learning model used for ranking predictions. Chapter 3 discusses the design and optimization of the web server. First, we offer a detailed overview of the prediction protocol implemented in the server. Then, we explain how we optimized the template interface set by structural clustering to enhance the server's performance. Also, we suggest a deep learning model to rank the predictions offered by the server. Finally,

the chapter covers backend and frontend design including database design, jobs scheduling, software design principles, responsive UI design, and 3D visualization of the predicted complexes. /

In Chapter 4, we present our results and analysis. First, we briefly describe the usage and functionality of the web server. Then, we evaluate the performance of each of the steps followed by our prediction protocol and present two case studies, one with a pathogenic and another with a commensal microbial protein, with and without resolved 3D structures respectively. Moreover, we analyze the performance of DeepInterface; the deep learning model for ranking predictions.

Finally, we offer a conclusion which gives a summary of the work done, and future work related to this thesis.

Chapter 2:

LITERATURE REVIEW

2.1 *Host-Microbe Interactions*

Microbes modulate host physiological processes in health and can induce diverse diseases, including cancer. Exactly how they initiate and maintain the disease phenotype is often still a mystery. Microbes can crosstalk with the host through proteins, metabolites, small molecules, and nucleic acids. They evolved efficient molecular mimicry strategies to usurp host immune surveillance by repurposing host signaling mainly through protein-protein interactions (PPIs) with the host. Microbes hijack host proteins at four different levels: sequence, structure, motif, and interface [1-4]. Interface mimicry is the local structural similarity that allows proteins with distinct global structures to interact in similar ways [5, 6]. It appears to be the most common molecular mimicry type and is observed frequently in endogenous (intra-species) [7, 8] and exogenous (inter-species) interactions [9]. Similarity in interface architectures enables microbial proteins to compete with host proteins to bind to a common target, and thereby alter host physiological signaling. A microbial protein that hijacks a host interface grapples with all host endogenous PPIs sharing this architecture. Elucidation of the complete set of host-microbe interactions (HMIs), along with their complex (bound) structures, will provide insights into microbes' virulence strategies and action on host signaling. Revealing the map of structural host-microbe crosstalk can also foster more potent therapeutic development.

Experimental HMI data, including structural details, are extremely limited and large-scale experimental detection is challenging. Yeast-2-hybrid (Y2H), protein arrays, and mass-spectrometry followed by affinity purification (AP-MS) or crosslinking are examples of high throughput experimental methods to determine HMIs [10]. Each method has downsides, such as Y2H and AP-MS have high false-negative [11] and false-positive rates [12], respectively. Structural determination is even more time consuming, expensive and difficult to apply on a large scale. There is an escalating demand for robust and efficient computational approaches to complete the HMI space and resolve the complex structures. Although predicting endogenous PPIs is common, modeling exogenous interactions – i.e. HMIs – is new. Still, several computational approaches have

been put forward to identify HMIs [13, 14], most of which rely on sequence homology [15-20]. Sequence-based methods are often successful only if the homology is high. However, not all microbial proteins have human homologs and they may still hijack host protein structure fully or partially [21] and impede host signal transduction. There are also methods that consider sequence motifs [22, 23], global structural similarity combined with sequence similarity [24-28], and structural motif-domain interactions [29, 30]. To date, the only method that relies solely on interface mimicry is ours [2, 3]. Interface-based approaches have also been successful in predicting endogenous host PPIs [31-36]. Since interface mimicry appears the most prevalent type, it holds promise in identifying more HMIs compared to other methods and may absolve the need for sequence homology. In addition, most of the available HMI predictors give the list of potential HMIs, but not the structures of the complexes. The availability of the complex structures of both modeled HMIs and endogenous host PPIs is key to deduce the potential consequences of the HMIs – whether they are likely to inhibit or activate the host pathway.

Although several methods have been published on predicting HMIs, only one web server is available [37]. This server requires the user to provide the sequences of both the microbial protein and its potential interactor in the host. However, the potential host partners are often unknown, and the main purpose of the predictors is to identify them.

2.2 Protein-Protein Interaction interfaces

Proteins seldom act on their own, instead they interact with each other to carry out their functions. Protein-Protein Interactions (PPIs) play a vital role in all living organisms. Protein-protein interaction interfaces can be extracted using multiple approaches such as Voronoi diagrams, accessible surface areas, and atomic distances.

Using atomic distances, interfaces are defined as the interacting and nearby residues. If the distance between two residues – any atoms – from each partner protein is less than the sum of their van der Waals radii plus 0.5 Å, these residues are defined to be interacting. The residues next to an interacting residue on the same protein – C α 's within a 6 Å radius – are assigned as 'nearby residues' [31]. Evolutionary conserved interface residues or hotspots are predicted according to computational hotspots. Hotspots are interface residues which have the highest contribution towards the binding energy; thus,

they are the primary targets of therapeutic agents. Hotspots can be identified experimentally by alanine scanning mutagenesis in which a residue is labeled as a hotspot if its contribution to the binding energy is higher than 2.0 kcal/mol [38]. Figure 2.1 presents the 3D structure of '1am4AD' protein-protein complex; interface and hotspot residues are highlighted in both interacting partners.

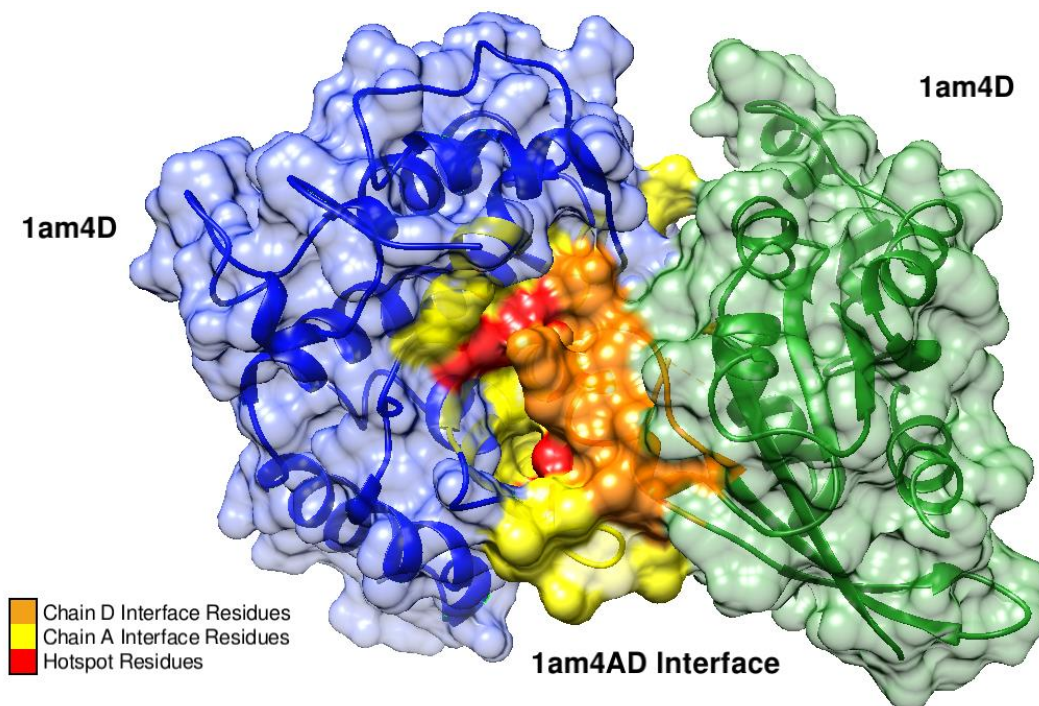


Figure 2.1: '1am4AD' PPI structure with highlighted interface and hotspot residues

2.3 Template-Based PPI prediction

Predicting PPIs is critical for scientists and researchers. Many structural-based computational methods exist for PPI prediction [39]. One of these methods is docking techniques which are designed as a complement to experimental methods. ZDOCK [40], RosettaDock [41] and HADDOCK [42] are examples of this method. However, docking techniques are computationally expensive and they rely on experimental knowledge to determine the interactions. On the other hand, homology-based methods are widely used for large-scale PPI predictions. PRISM [31] is one of the first and widely used template interface-based PPI prediction tools. PRISM exploits the observation that protein interface regions are evolutionarily more conserved compared to the rest of the protein surface. HMI-PRED's protocol was inspired by PRISM's protocol; however, several

changes were made to accommodate for the HMI prediction requirements, as well as using different structural alignment, refinement and scoring tools.

2.4 Ranking PPI Interfaces with Deep Learning

Proteins interact with each other through their interfaces [43]. Defining a PPI interface requires knowledge of the 3D structure of the protein-protein complex it belongs to. Nevertheless, many of the PPIs do not have structural information. Computational PPI prediction and docking techniques are used to provide predicted PPI structures [44]. Docking a protein pair generates hundreds of decoys. It is crucial to rank these decoys and identify the native-like structures among them.

There are several interface properties which can be used to identify interface residues among other surface residues. They include physicochemical properties of residues such as their polar/nonpolar nature, phosphorylation and methylation status, and number of H-bond acceptors and H-bond donors. In addition, hydrogen bonds, salt bridges, hotspots, and secondary structure types found in the interface region help in the interface identification [45-48]. These properties can be considered in different stages of prediction process; either at the beginning to help make the prediction [31, 32, 49, 50], or at the end to rank/score predicted complexes [51, 52].

There are few available methods currently for scoring protein-protein complexes using their interface properties. CIPS (Combined Interface Propensity for decoy Scoring) by Nadalin and Carbone considers the composition of the interface together with the contact preference among residues [52]. Popov and Grudinin showed that possibility of predicting binding candidates by statistical learning from the knowledge of the native protein-protein interfaces only [53]. Chen and Zhou developed a cons-PPISP (consensus neural network protein-protein interaction site prediction) that predicts binding residues using the protein structures as an input [54].

The rapid expansion of structural protein data in protein data bank allowed the accumulation of protein-protein docking results. These results can be used to form protein-protein interface structural datasets. ProtinDB [55], and ProtCID [56] datasets list all possible interfaces. On the other hand, PIFACE [57] dataset clusters the available interfaces based on structural similarity.

Large sequence or structure-based datasets can be used in machine learning and deep learning models for PPI prediction, protein folding, and protein fold classification. Deep learning has the advantage of direct usage of raw data compared to shallow machine learning algorithms which need data with high level features.

DeepInterface [58], the model suggested in this study as a ranking tool, was developed by Tugrul et al as a classifier of predicted protein-protein interface into native or non-native. The data used for training and testing of the model is composed of protein-protein interface atomic coordinates and their residue types.

The data of DeepInterface model comes from four resources: Protein Data Bank (PDB) [59], PIFACE [57], DOCKGROUND [60], and PPI4DOCK [61]. The first two are the sources of positive samples, while the last two are the sources of the negative samples in the datasets we used for our deep learning model. More details on the model, and data composition and size can be found in Section **Error! Reference source not found.**

2.4.1 PPI4DOCK dataset

The standard PPI4DOCK dataset has 1,417 non-redundant docking targets in their unbound state where targets share less than 70% sequence identity. Each of these models was used to generate up to 54,000 decoys. The initial data which was used to create PPI4DOCK dataset contained 5,641 heterodimeric interfaces obtained from InterEvol [62] database. These interfaces represent interactions between two single-chain proteins with less than 70% sequence similarity. The generation of the standard PPI4DOCK dataset went through five main steps.

First, Yu and Guerois filtered the initial 5,641 interfaces based on the following criteria which defines the rules of a “good-quality” homodimer:

- i. X-ray resolution ≤ 3.25 Å.
- ii. Contains ≤ 20 chains in the biological assembly.
- iii. The interface should be predicted by NOXclass [63] as biological as opposed to crystal.
- iv. The interface size > 300 Å².
- v. Does not contain interfacial ligands.

In the second step, Yu et al applied HHsearch [64] on each side of the 3,157 interfaces to search for candidate homologs. Pools of candidates were collected such that a candidate

has higher than 50% HHsearch probability, higher than 20% sequence similarity with the reference interface chain, and more than 20 aligned columns. Pools of candidates had at most 60 templates by the end of this step.

In step 3, first Yu et al filtered “good” candidates from the pool of all candidates. They define a “good” candidate as the one satisfying the criteria:

- i. X-ray resolution $< 4\text{\AA}$.
- ii. Contains < 20 chains in the biological assembly.
- iii. Covers at least 40% of the interfacial residues (CAPRI’s [65] definition of interface residues is used; where interacting interface residues are two non-hydrogen atoms in different chains separated by a distance $\leq 5\text{\AA}$).
- iv. Has ≥ 0.6 TM-score provided by running TM-align [66] on the candidate model and its reference protein. Such score indicates that both aligned structures belong to the same family/fold.
- v. Represents a proper unbound conformation, which was assessed using iAlign’s [67] confidence scores to ensure distinct interfaces.

Then, the “best” template candidate was selected as the one with the highest TM-score among all candidates.

In the fourth step, Yu et al generated a homology model using RosettaCM [68] for the heterodimeric interfaces, for which valid templates were assigned to each of the interface sides. The dataset after this step contained 1,531 modeled interactions.

The last step was models’ evaluation. First, in case multiple spatially separated sub-regions existed within one chain in a model, Yu et al stripped all but one region which has the most contacts to the other chain to emphasize the interface area. If stripping the model reduced the interface size by 50% or more, the model was discarded. Models with less than 30 residues after stripping were also discarded. The stripped models were aligned with the x-ray structures of the corresponding target subunits using TM-align. Models with both TM-score < 0.5 and sequence similarity $< 50\%$ were discarded. Then, Yu et al used TM-align to superimpose the remaining homology models onto their reference bound structures to form a “superimposed decoy”. The cases where there exist less than 5 residues in contact or the interface size is less than $300\text{\AA}$ were removed.

The final PPI4DOCK dataset contained 1,417 models, which had the same chain IDs, sequences, and residue numbers as the reference structures. Up to 54,000 decoys were generated for each target using ZDOCK 3.0.2 [40] which uses a fast Fourier transform

(FFT)-based docking algorithm. All decoys were evaluated based on the CAPRI criteria and categorized into ‘Incorrect’, ‘Acceptable’, ‘Medium’, or ‘High’. CAPRI scores of all decoys with $F_{\text{nat}} > 10\%$ are stored and can be obtained from PPI4DOCK dataset.

2.4.2 DOCKGROUND Dataset

DOCKGROUND [60] source presents several docking benchmark sets and docking decoys sets (scoring benchmarks), each of which can be used for different tasks. Docking benchmarks include sets of unbound, simulated unbound, and model structures. Docking decoy sets are used to test scoring/ranking functions; therefore, the dataset offers several poses of each protein-protein complex one of which should have near-native structure while the remaining poses are decoys (i.e., false positives). The decoy set has two subsets; the first set contains 101 structures (at least 1 of which is near-native) for 61 unbound complexes from the docking benchmarks in DOCKGROUND, the second subset forms a larger set of 100 decoys (at least 1 of which is near-native) of 351 unbound structures. The decoys were generated by an FFT-based tool GRAMM-X [69]. In addition to the structures, DOCKGROUND resource provides lists of IRMSD, LRMSD, and F_{nat} values of each structure. Although CAPRI [65] scores are not offered by DOCKGROUND, CAPRI classifications can be calculated using the given details (i.e., IRMSD, LRMSD, and F_{nat}).

2.4.3 PIFACE Dataset

PIFACE [57] dataset consists of PPI interfaces extracted from protein structures deposited in PDB before 2012. Cukuroglu et al. used networks algorithms and structural alignment tools to cluster the interfaces based on their structural similarity into unique sets/clusters.

First, Cukuroglu and his colleagues fetched all multiple-chain protein structures deposited in PDB as of January 14th, 2012 (45,491 structures were found). Then, they generated a list of all possible binary interactions between the chains within each structure (622,321 such interactions were found). However, not every pair in a structure is physically interacting. To avoid such cases and determine the correct number of binary interactions (i.e., interfaces), Cukuroglu et al. used NACCESS [70] to calculate the accessible surface area (ASA) of the protein structures. They suggest that if a significant

difference exist between the ASA of a chain/protein as a monomer and the ASA of the chain/protein in a complex, then the two chains are considered as interacting partners. This filtering reduced the number of possible interactions/interfaces to 130,209.

Cukuroglu et al. calculated pair-wise structural alignment using the sequence-independent structural alignment tool Multiprot [71]. Then, they defined an interface similarity measure as in equation 2.1. To speedup similarity calculations, interface pairs having highly unbalanced lengths were discarded.

$$S_{1,2} = \frac{M}{\min(N_1, N_2)} \quad (2.1)$$

M: the number of matched residues by Multiprot's alignment

N_i: The number of residues in the *i*th interface

Once pair-wise similarity measures are calculated, interfaces were modeled as a graph; each interface represents a node in the graph, and each edge between a pair of nodes represents the similarity between the corresponding interfaces pair. Cukuroglu et al. clustered the graph using Girvan and Newman method [72] which finds communities (i.e., groups of densely connected nodes) within a graph. The algorithm outputs a set of communities/clusters/groups of structurally similar interfaces within the set of all interfaces.

Chapter 3:

MATERIALS AND METHODS

3.1 *HMI-PRED protocol*

The rationale behind HMI-PRED is that if the microbial target protein has a patch of surface that is structurally similar to one face of the template interface with similar evolutionarily conserved “hot spots”, it can interact with the host protein on the complementary face. Structural alignment of the microbial protein with all known human interfaces allows determining the potential HMI pairs, which are later scored with Rosetta docking [73] to filter out the energetically unfavorable ones. HMI-PRED performs online calculations and the results are stored for community access. Our applications revealed that HMI-PRED generates promising HMI models that can explain the molecular mechanisms of how microbes evade the host immune, alter cytoskeletal dynamics, and modulate cell cycle regulation [2, 3].

HMI-PRED protocol can be described as a four-step algorithm, which involves extracting the surface from the input target protein structure, aligning the target to both faces of the template interfaces in the template set, transforming the predicted structures and filtering the results, and finally, refining and scoring the predicted structures. Figure 3.1 presents an abstract overview of the flow of the protocol; The microbial protein structure is the only user input to HMI-PRED. If it has not been resolved yet, homology model can also be used. We first extract the surface of the microbial protein, then structurally align each template interface with the microbial protein surface and find some matches. If the microbial protein is aligned with the B-face of the interface, we assume that it can interact with the complementary face, A. We generate the potential HMI pairs based on this assumption. However, structural complementarity may not always guarantee electrochemical complementarity. The next several sections offer a more detailed explanation for each of the steps. Figure 3.2 illustrates a detailed overview of the protocol steps.

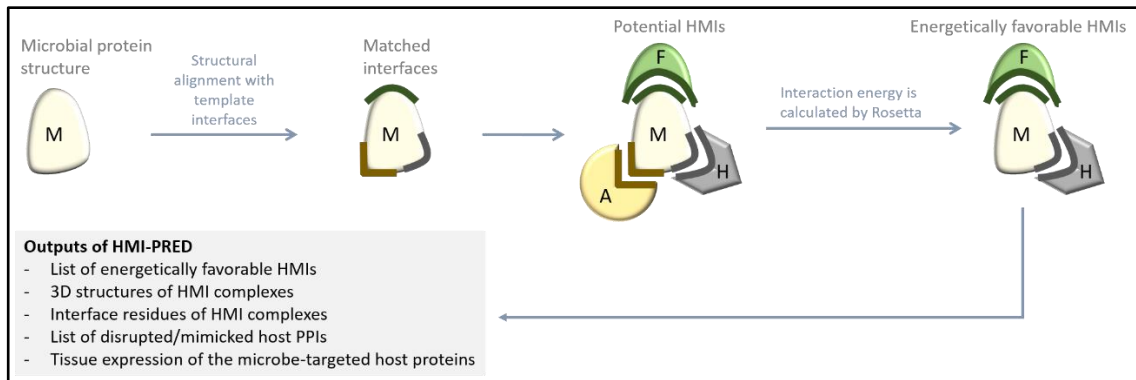


Figure 3.1: Abstract Overview of HMI-PRED protocol

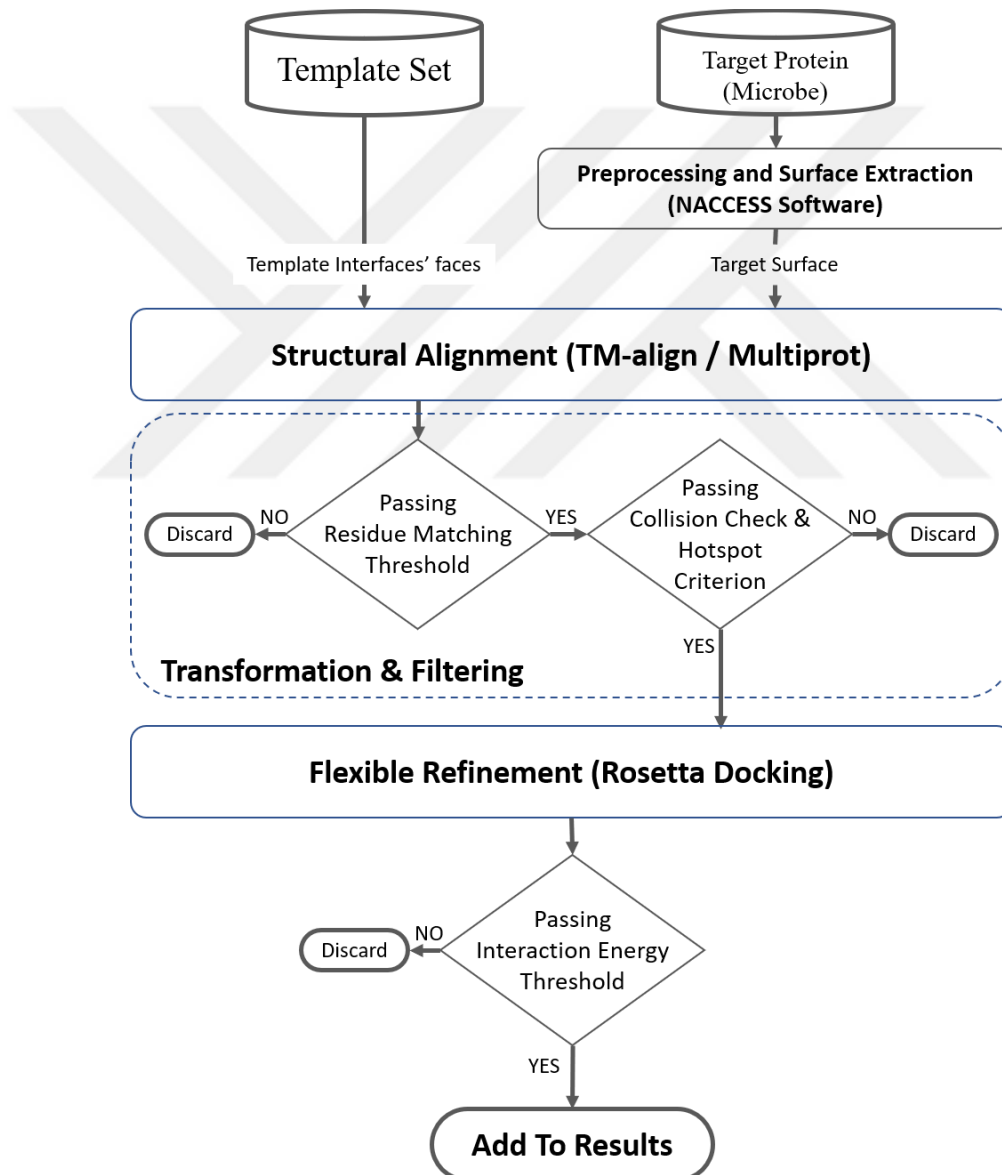


Figure 3.2: Detailed Overview of HMI-PRED Protocol Stages

3.1.1 *Preprocessing and Surface Extraction*

The first step of HMI-PRED is preprocessing and surface extraction. The steps performed in this stage are:

- 1- Extract the microbial target organism and Uniprot [74] entry name from the PDB file.
- 2- Check if the target chain contains DNA or RNA structures. Such structures are discarded in the algorithm.
- 3- Run Naccess [70] software on the target protein. Naccess simulates a solvent probe which is rolled on the target molecule to calculate its accessible surface area. The program can handle targets containing up to 20,000 atoms and it allows for user-defined prob size and user-defined atomic van der Waal radii. The default radii and default probe size of 1.4 Å are used in HMI-PRED.
- 4- Extract surface residues based on the relative accessibilities of each residues calculated by Naccess. A protein surface is defined as the set of residues having a relative accessibility > s15% (default). In this step, only carbon alpha atoms are extracted from each surface residue.
- 5- Extract the structural scaffold of the target surface. The scaffold is constructed from nearby residues which have their corresponding carbon alpha atoms within 5.0 Å distance from the surface residues.

3.1.2 *Structural Alignment*

In this step, all the interfaces in the template set are split into their two corresponding faces. Then, the extracted surface of the input target from the first step is structurally aligned to both faces of each interface. The structures are aligned using either Multiprot [71] or TM-align [66] (default) sequence-independent structural alignment tools based on the user choice. In the case of exogenous template interfaces, the target is aligned with the non-human side of the interface only (i.e., the possible interaction is the target with the human side of the interface). Also, if the template interface is a symmetric homodimer, the target is aligned with only one side of the interface to avoid redundant results. We determined the symmetric homodimer interfaces. 3,617 of the 17,351 representative interfaces in our default template set are symmetric homodimers (TM-threshold for symmetry is 0.80).

This step aims to find structural match between one (or both) of the interface faces and the extracted surface of the input target protein. For a structural match, the default threshold for TM-score is 0.25, but the user has an option to change it, in the “Advanced Options”. TM-score ranges between 0 and 1: higher thresholds allow more stringent and lower thresholds are for more relaxed alignment. The alignment threshold for Multiprot is 2 Å. Figure 3.3 illustrates the alignment process starting with two split sides of the interface template I_L and I_R , and the extracted surface of the target protein T . The algorithm aligns the target to both interface faces. When a structural compatibility is found (i.e., T is structurally compatible with I_L), HMI-PRED passes the resulting complex to the next step which includes both sides of each complex with the target’s corresponding transformation matrix which is calculated by the structural alignment tool of choice.

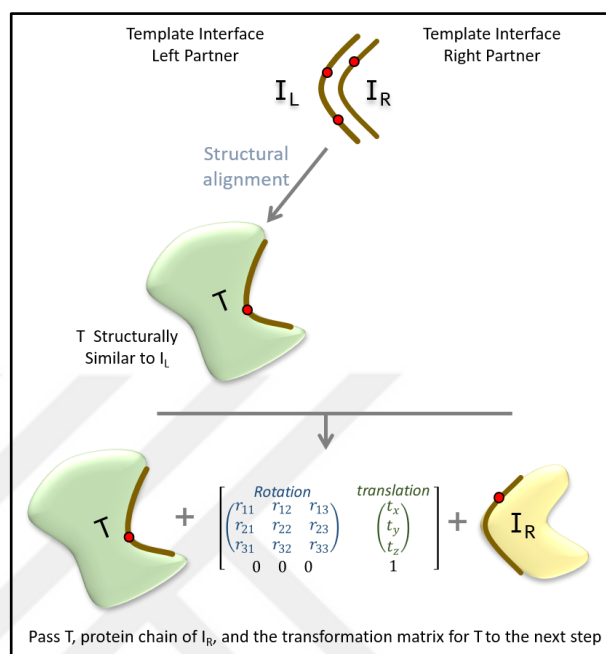


Figure 3.3: Structural Alignment Step

3.1.3 Transformation and Filtering

In this step, a set of predicted complexes is first filtered based on structural similarity and hotspots matching. To pass the first filtering step, at least 50% structural matching should be found for the template interface chains which are shorter than 50 residues, and at least 30% for the longer chains with more than 50 residues. However, in both cases there should be at least 15 residue matches and 1 hotspot residue of the template interface chain should geometrically match with a residue of the same amino acid type in the microbial protein surface. The use of hotspots in protein-protein interactions prediction provides evolutionary insights into the predictions.

Once a structural match is found and passed through initial filtering, the full structure (not only surface residues) of the microbial protein is transformed onto the aligned interface using the transformation matrix from the previous step. The aligned protein of

the template PPI is replaced with the transformed microbial protein, while the other template face is replaced with its full structure (instead of only interface residues), creating the putative HMI pair between the microbial protein and the un-aligned face of the template.

Then, a second filtering is applied where HMI-PRED checks the complex for clashes between the interacting partners. A clash is detected if an atom from one of the interacting chains is within a clashing distance of 3Å from an atom in the other chain. Predictions with 5 clashes or more are discarded. The remaining potential HMI complexes have structural (geometrical) complementarity, but they may not have electrochemical complementarity, i.e. favorable interaction energy. Thus, the remaining predictions are passed to the next step to identify the energetically favorable HMI complexes.

3.1.4 Flexible Refinement

To identify the energetically favorable HMI complexes, HMI-PRED exploits Rosetta (local refinement protocol, Rosetta 2018.09.60072) [73]. HMIs with Rosetta interface scores (I_{sc}) smaller than -5 and total energy scores smaller than zero are assigned as energetically favorable. I_{sc} of template PPIs are also calculated to assess whether the microbial proteins will outcompete the bona fide partners of the microbe-targeted host proteins.

3.1.5 HMI-PRED Output

HMI-PRED outputs a list of predicted energetically favorable HMI complexes with their Rosetta interface scores (I_{sc}). Also, to help users in evaluating the HMI models further, HMI-PRED also provide details, such as:

- i. Percent match (ratio of the number of aligned residues to the total number of residues in the aligned face of the interface).
- ii. The assigned weight of the interface which depends on the interface size. The larger the size of the interface, the bigger its “weight” is: weight is 0.5 for interfaces with less than 30 residues ($n < 30$), 1 for $30 < n < 50$, 1.5 for $50 < n < 80$, and 2 for $n > 80$.

- iii. The probability of template interfaces being real biological interfaces or crystal artifacts. EPPIC (Evolutionary Protein-Protein Interface Classifier) [75] server was exploited to obtain the probability of interfaces being biological interfaces.
- iv. Tissue expression of the host proteins. This data is obtained from the Human Protein Atlas [76, 77].

Figure 3.4 and Figure 3.5 show a sample for the results page offered by HMI-PRED and its functionality. In Figure 3.4a, the page shows the results table of a sample HMI-PRED job. The first column is the microbial protein PDB (the user input); second column is the microbial organism; third is the microbial protein name; fourth is the human protein that is targeted by the microbial organism; fifth is the human protein name; sixth is the mimicked human PPI, i.e. template PPI; seventh and eight are the interface scores (I_{sc}) for the HMI model and the template PPI, respectively. If the I_{sc} of the HMI is lower than the template PPI, it suggests that microbial protein has higher affinity to the human protein than the human protein's bona fide partner in the template PPI. User can sort the columns that have PDB IDs and I_{sc} values. Figure 3.4b Shows that clicking on the "View" button in the last column opens a pop-up page for visualizing the HMI model interactively. The user can rotate, change colors, download the structure in PDB format ("Download Structure" button). The user can also visualize the predicted structure with highlighted contact residues, Figure 3.4c illustrates that the user can retrieve the interface residues of the predicted HMI, Figure 3.4d and Figure 3.4e show the visualization of the HMI structure with highlighted interface residues, and the superimposed structures of the HMI on the template PPI respectively. The superimposed structure shows how well the microbial protein hijacks the human interface on the human protein (although they have different global structures).

The functional annotation box shown at the top of the results page in Figure 3.5a can be clicked to direct the user to DAVID website [78, 79] show in Figure 3.5b, where the KEGG pathways that are enriched with the microbe-targeted host proteins are seen. Figure 3.5c shows the option for the user to download the table at the DAVID website, (s)he can see which microbe-targeted human proteins are involved in a given KEGG pathway under "Genes" column. "Count" column gives the number of microbe-targeted host proteins and "Pop Hits" column gives the total number of human proteins involved in the KEGG pathway.

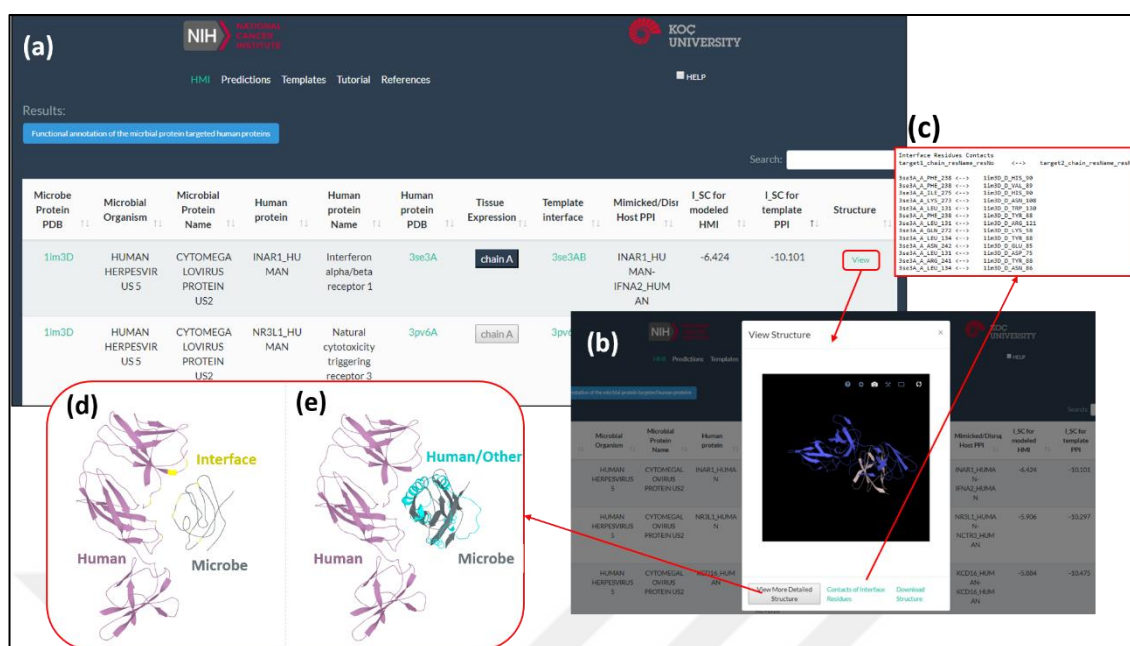


Figure 3.4: HMI-PRED Results Page Details and Functionality.

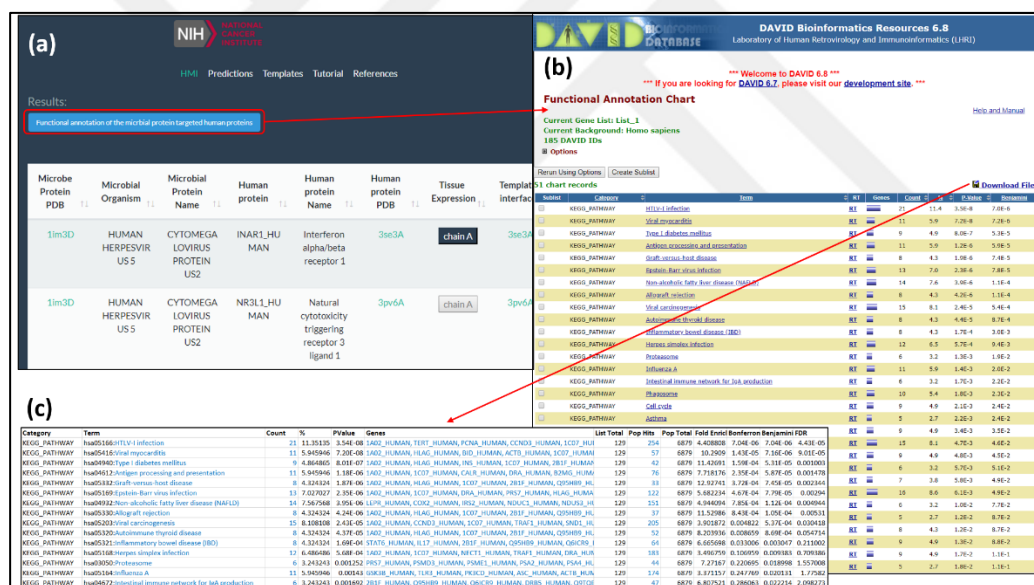


Figure 3.5: Functional annotation of microbe-targeted host proteins

3.2 Template Interface Set

We extracted the interfaces from all cocrystal endogenous and exogenous human PPI structures in PDB that are deposited as of January 3rd, 2019. Figure 3.6 illustrates the process of extracting all human interfaces from PDB. Some PDB entries have huge protein assemblies and are not available in PDB format. We excluded them and chimeric proteins at template generation step. Each interface has 2 faces (partners/chains/sides), one from each interacting partner in the host PPI. In each interface in our set, at least one

face is from human. If both faces are from human, it is an endogenous interface. If one face is from human and the other face from another organism, it is an exogenous interface. Our non-redundant template interface set has 15,762 endogenous and 1,589 exogenous interfaces. Interfaces are composed of interacting and nearby residues. If the distance between two residues – any atoms – from each partner protein is less than the sum of their van der Waals radii plus 0.5 Å, these residues are defined to be interacting. The residues next to an interacting residue on the same protein – C α 's within a 6 Å radius – are assigned as 'nearby residues'. Evolutionary conserved interface residues or hotspots are predicted according to computational hotspots [38, 80]. All interfaces in our template set have at least 1 hotspot residue and more than 15 residues in at least one side for endogenous interfaces and in the non-human side of the exogenous interfaces. The next section covers the details of preparing and clustering the template set.

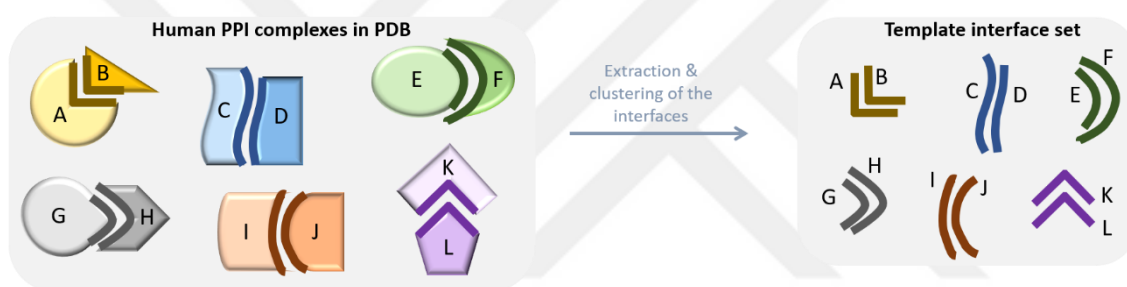


Figure 3.6: Preparation of template interface set.

3.2.1 Template Set Preparation

The preparation of the template interface set for HMI-PRED server followed multiple steps.

- i. The list of all PDB IDs of the deposited human proteins in PDB as of January 3rd, 2019 has been obtained,
- ii. All possible binary interacting partners where at least one side is a human protein has been generated from the previous list. For example, “10gsAB” means that there is a possible interaction between the proteins in chains A and B in the structure with the PDB ID “10gs”. We found 197,145 possible binary interactions.
- iii. Template PPI interfaces were extracted from the list obtained in step 2. Each interface has 2 faces (partners/chains/sides), one from each interacting partner in the host PPI. Interfaces are composed of interacting and nearby residues. If the distance between two residues – any atoms within the residues– from each partner

protein is less than the sum of their van der Waals radii plus 0.5 Å, these residues are defined to be interacting. The residues next to an interacting residue on the same protein – Cα's within a 6 Å radius – are assigned as 'nearby residues'. Both interacting residues and 'nearby residues' form the template interface structure. Some PDB entries have huge protein assemblies and are not available in PDB format. We excluded them and chimeric proteins at template generation step.

- iv. After interface generation was done, a list of all interfaces generated successfully was obtained, in addition to their 3D structure, contact residues, and predicted hotspots. The output list from this step contained all cocrystal endogenous and exogenous human PPI structures in PDB that are deposited as of January 3rd, 2019. Evolutionary conserved interface residues or hotspots are predicted according to computational hotspots [38, 80].
- v. Evolutionary Protein-Protein Interface Classifier (EPPIC) [75] server was utilized to obtain the probabilities of the template interfaces being valid biological interactions.
- vi. All interfaces satisfying one of the following conditions were discarded:
 - a. Having no hotspots in both sides of the interface.
 - b. Being an exogenous human PPI and having no hotspots in the non-human side of the interface.
 - c. Having less than 15 residues in both sides of the interface.
 - d. Being an exogenous human PPI and having less than 15 residues in the non-human side of the interface regardless of the other side.
 - e. Having less than 5 residues in one side of the interface with EPPIC probability of 0 for the template interface.

By the end of this step, we had 53,339 redundant human PPI interfaces in our template set.
- vii. We mapped the template interfaces in our set to the Uniprot [74] entry names corresponding to their two interacting partners. For example, the template '4hxdAB' is mapped to (UBC_HUMAN – L_DUGBA), since it represents the interaction interface between the protein '4hxdA' (UBC_HUMAN) and the protein '4hxdB' (L_DUGBA).
- viii. We formed template PPI groups, where a PPI group represents the set of all template interfaces of a specific protein-protein interaction. For example, the

PPI group UBC_HUMAN – L_DUGBA : [4hxdAB, 4hxdCD], represents the template interfaces for the interaction between the protein UBC_HUMAN and the protein L_DUGBA where these proteins' names are the proteins' Uniprot entry names. This group has two members: 4hxdAB and 4hxdCD. Within the full template interfaces set, there was 7,681 distinct PPI groups; 2,313 of them have only a single member, and the remaining 5,368 groups have multiple templates as members. Figure 3.7 illustrates the frequency distribution of the PPI groups sizes. It can be noted that almost all PPI groups have less than 500 members while the majority have less than 100 members.

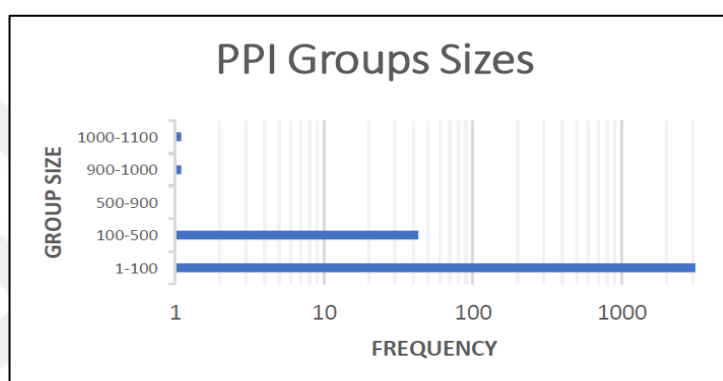


Figure 3.7 The frequency distribution of PPI groups size.

For each PPI group, we clustered the interfaces within the group and selected representatives of each cluster, through the following steps:

- ix. A pairwise structural alignment between every possible pair within the PPI group has been run using TM-align [66]. TM-align takes two protein structures as an input, structurally aligns them, and outputs the TM-score between 0 and 1 for the alignment; a TM-score of 1 means that the two input structures had been aligned exactly. This score can be used as a structural similarity measure between the input complexes.
- x. All the calculated pairwise TM-scores were used to construct a similarity matrix as in the following example:

PPI group: UBC_HUMAN – L_DUGBA

templates: 4hxdAB 4hxdCD

1	0.932115
0.932115	1

The entry 0.932115 in the 1st row, 2nd column means that the TM-score from aligning the 1st template (4hxdAB) with the 2nd template (4hxdCD) was 0.932115

- xi. Similarity matrices were used to construct adjacency matrices by following the rule:

IF

similarity score ≥ 0.8

THEN

set the corresponding adjacency matrix element to 1

ELSE

set the element to 0

- xii. A graph with nodes/vertices and edges $G(V, E)$ was constructed for each PPI group. A node in the graph represents a template interface, and an edge represents a structural similarity between 2 template interfaces connected by the edge. These graphs were used as inputs to our clustering algorithm which is explained in detail in the next section. The purpose of clustering is to reduce structural redundancy in the template interface set used in HMI-PRED which will speed up the computations done by HMI-PRED jobs and increase the quality of the predicted HMI complexes by avoiding redundant results. It is important to note that we only avoid structural redundancy within the same PPI group; thus, in case of having similar 3D structures of two predicted HMIs mimicking two templates of different protein pairs, both predictions should be included in the results since it is important for researches and scientists to know about both possibilities.

3.2.2 Clustering Heuristics

HMI-PRED server uses a set of experimentally known 3D-structures of human PPI interfaces for prediction. Due to the large size of our initial template interfaces set (53,339 templates) and the structural redundancy between various template interfaces within it, the results produced by the server take a very long time to be computed on one hand. On the other hand, the results have many redundant predictions which reduces their quality. To address both problems, the template set size should be reduced in a way that preserves its structural variation while decreasing redundancy. This reduction can be achieved by clustering the templates set into structurally similar clusters and picking a representative of each cluster to be in the reduced set.

In other words, the clustering algorithm should output a set of representatives R , where every vertex in G which is not picked as a representative should be connected to at least one of the representative vertices in R . The main goal of the algorithm is to **minimize** the size of R to remove redundancy and speedup predictions. This problem is known in literature as the minimum dominating set problem (a special type of graph cover).

Since finding optimal clustering/graph-cover is an NP-Hard problem, there is no known exact polynomial-time algorithm to solve it. Therefore, we suggested two different heuristics to achieve the required job, a comparison between the results of both approaches can be found in the next chapter. Both algorithms take as an input a graph G (V , E) representing a set of templates (one of the PPI groups) and their similarity matrix. The output is a set of nodes (i.e., template interfaces) which provides a minimum structural representation for the whole graph G . Figure 3.8 represents an example of the clustering algorithm's input and output.

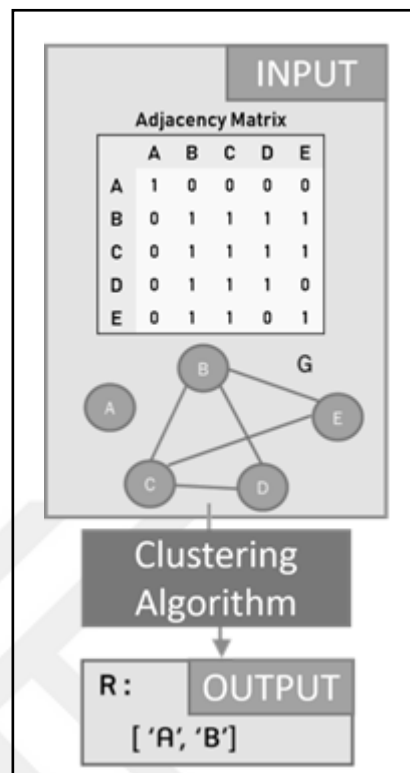


Figure 3.8: An example of the input and output of clustering algorithm.

After the full template list is grouped by PPI groups (as mentioned in the previous section), each PPI group is input to the clustering algorithm and all representatives are collected to obtain the minimum representatives set for the full template set.

Both suggested heuristics for the clustering algorithm build upon the concept of finding a maximal-clique cover of a graph. A clique is defined as a fully connected subgraph. A “maximal-clique” is a clique, to which no other node can be added while preserving its full connectivity. Figure 3.9(b-d) illustrates the distinction between a non-clique subgraph (b), a non-maximal clique (c) and a maximal-clique (d). It is also important to note the distinction between a maximal-clique and the maximum-clique of a graph; the latter is the maximal-clique with the largest size in a specific graph. Figure 3.9(e-f) illustrates the distinction; all maximal-cliques are highlighted in the graph in (e), while only the maximum-clique is highlighted in (f).

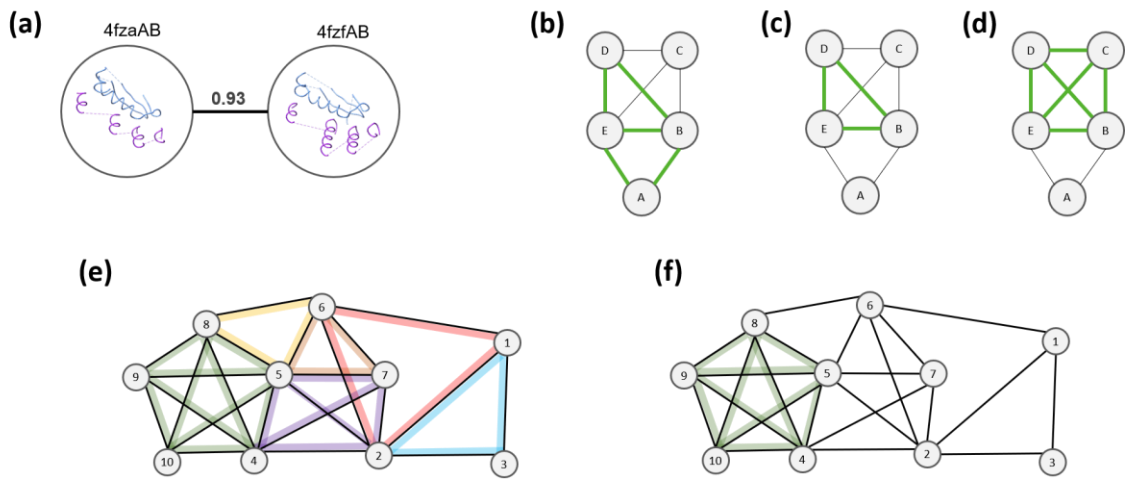


Figure 3.9: Interface clustering steps. (a) Shows the structural similarity between interfaces that correspond to the same PPI with a node-and-edge representation, where nodes are interfaces and the edges are the structural similarity between them. The edge weight represents the similarity score (TM-score) between a pair of nodes and we connect 2 nodes with an edge only if their TM-score is above 0.8. Here, each face of an interface is regarded as a single entity and structurally aligned with other interfaces with TM-align. (b-d) A “maximal-clique” definition (b) Nodes (A, B, D, E) do not form a clique because they are not fully connected, A-D edge is missing. (c) Nodes (B, D, E) form a clique, all nodes are interconnected; however, it is not a maximal-clique. (d) Nodes (B, C, D, E) form a maximal-clique, no more nodes can be added while preserving the full connectivity. (e) shows an example of all maximal in a graph of 10 nodes, each maximal clique has a distinct color. (f) shows the maximum-clique in the same graph.

First Heuristic CLUSTER_V1

The first heuristic we suggested ‘CLUSTER_V1’ utilizes Tomita et al.’s variation of Bron-Kerbosch algorithm (Check Appendix A). First, CLUSTER_V1 finds the maximum-clique in the graph G , picks a representative node from it, remove its nodes from G , then repeat the steps until G is empty.

Figure 3.10 illustrates the execution of CLUSTER_V1 algorithm on a sample graph. Starting from the graph G in Figure 3.10a, the maximum-clique is first found (highlighted in green in (b)), and node {5} (highlighted in red) is picked as the first representative. In Figure 3.10c, the nodes of the previous clique {4,5,8,9,10} were discarded from G , and

the maximum-clique within the remaining nodes is found $\{1,2,3\}$, node $\{1\}$ is picked as the second representative. Figure 3.10d shows again removing the nodes, finding the next max-clique and picking $\{6\}$ as the last representative. Figure 3.10e is the list of representatives (i.e., the output of the algorithm).

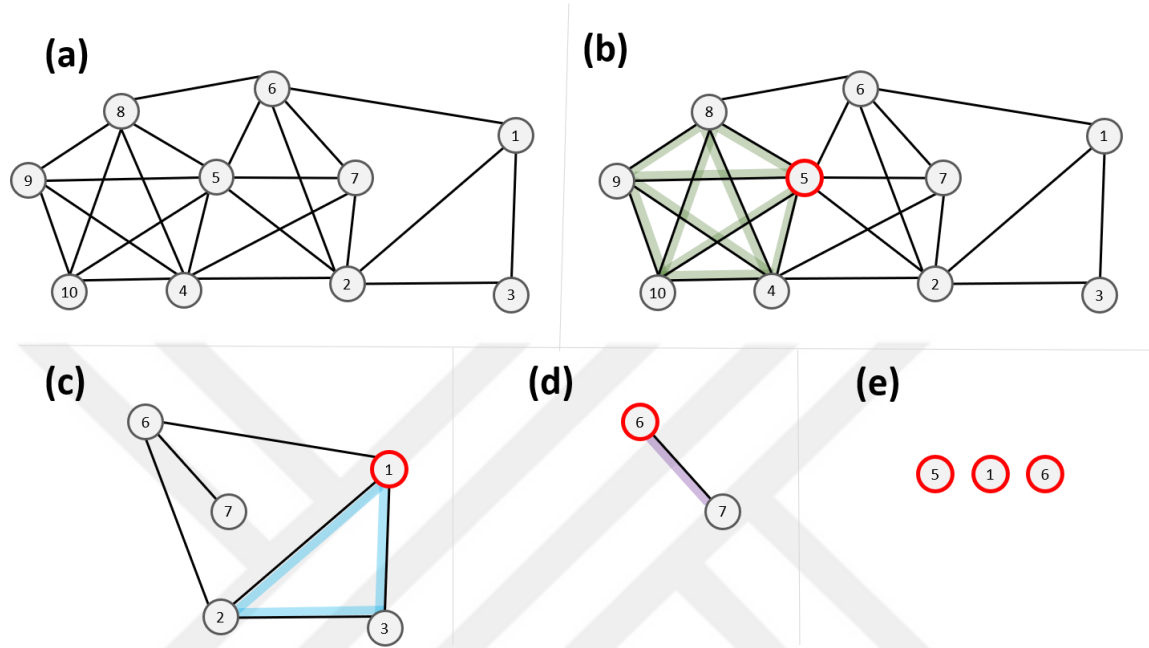


Figure 3.10: Example of the execution of CLUSTER_V1 algorithm.

Second Heuristic CLUSTER_V2

Our second heuristic aims to find a near-optimal maximal-clique cover to the graph utilizing the Tomita et al.'s variation of the Bron-Kerbosch algorithm (Check Appendix A). The pseudocode of the algorithm is shown in Figure 3.11. Given a graph g , CLUSTER_V2 first finds the connected components in g 'connectedComponents', since the graph is not guaranteed to be connected. Then, for each subgraph/component (g') in 'connectedComponents', the algorithm enumerates all maximal cliques using bronKerbosch_v2 (Check Figure A.4)**Error! Reference source not found.** From the list of all maximal cliques 'cliques', the algorithm picks representatives using a heuristic based on the number of cliques each vertex is a member in.

To find the connected components in the graph, either Breadth First Search BFS or Depth First Search DFS can be exploited to report the connected components using a stack or a queue respectively. The latter option was used in CLUSTER_V2.

```

1- CLUSTER_V2(g):
2-   // Since g does not have to be necessarily connected,
3-   // connected components of g are found first
4-   connectedComponents <- calculateConnectedComponents(g)
5-   // This list is used to store all picked representatives
6-   allRepresentatives = []
7-   for g' in connectedComponents:
8-     cliques <- enumerateMaximalCliques(g')
9-     rep <- pickRepresentatives(cliques)
10-    allRepresentatives.add(rep) // store the representative
11-
12-   return allRepresentatives

```

Figure 3.11: Pseudocode of CLUSTER_V2 algorithm.

Picking the representatives in CLUSTER_V2 is a minimization problem, where we need to pick the minimum number of representatives R from the list of all maximal cliques such that every vertex in $V \setminus R$ (i.e., nodes in V but not in R) is connected to at least one vertex in R . Since the number of cliques in a graph of size n is $O(3^{(n/3)})$, using a brute force algorithm would not be feasible. Therefore, we applied a heuristic (Pseudocode in Figure 3.12) where from a list of all possible candidates (initially, it contains all vertices), we pick the vertex included in the highest number of cliques (i.e., the vertex having the highest ‘cliqueCount’) as a representative, then add it and all the vertices contained in the clusters where it exists to the ‘verticesCovered’ list (vertex in this list should not be considered as possible candidate in the next iterations), then repeat the steps until no vertices are left as possible candidates (i.e., all vertices are in ‘verticesCovered’). Figure 3.13 illustrates an example of the algorithm’s execution on a sample input.

Different solutions can be seen when comparing the results of executing CLUSTER_V1 and CLUSTER_V2 on the sample input in Figure 3.10 and Figure 3.13 respectively. Having different solutions is due to using a heuristic to obtain a near optimal solution instead of using an exact algorithm. After comparing both suggested heuristics (comparison results and details can be found in Chapter 4:), we picked the second heuristic CLUSTER_V2 to use in our algorithm due to its more optimal performance.

```

1. pickRepresentative(cliques, G(V, E)) {
2.   representatives <- []
3.   // for each vertex in G, count the number of cliques that it is a
4.   // member in. 'templatesCliquesCount' has <vertex, cliquesCount>
5.   // pairs for all vertices in G
6.   templatesCliquesCount = countVertexCliquesMembership(cliques)
7.
8.   // push all the pairs from templatesCliquesCount into a priority queue
9.   // pq which orders the elements by their 'cliquesCount'
10.  for p in templatesCliquesCount
11.    pq.push(p)
12.
13.  verticesCovered = []
14.  while (pq is not empty)
15.    top_v <- pq.top
16.    if top_v is not in verticesCovered
17.      representatives.add(top_v)
18.      for all cliques containing top_v
19.        verticesCovered.add(clique.vertices)
20.
21.  return representatives

```

Figure 3.12: Pseudocode for Picking Representatives in CLUSTER_V2

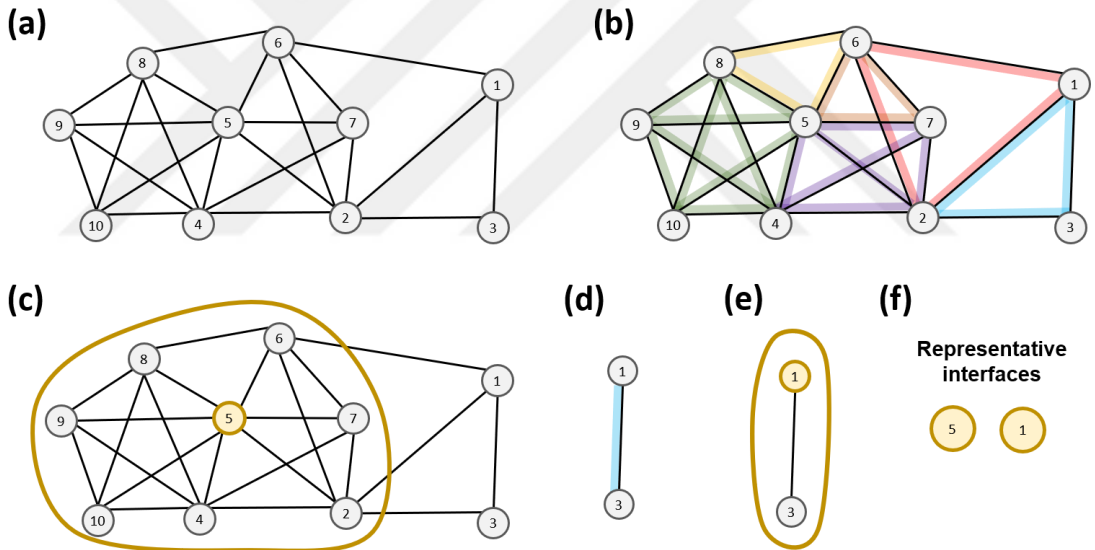


Figure 3.13: Example of the execution of CLUSTER_V2 algorithm.

Starting from the graph G in (a), all maximal-cliques are first found (highlighted in distinct colors in (b)). In (c) node {5} (highlighted in yellow) is picked as the first representative since it is a member of 4 maximal cliques (max. cliqueCount). In (d), the vertices of the representative and the cliques it is member in {2,4,5,6,7,8,9,10} were discarded from G , and the maximal-cliques within the remaining vertices are found. (e) node {1} is picked as the second representative (member in 1 clique, same as 3). (f) is the list of representatives (i.e., the output of the algorithm)

3.3 Predicted PPI Validation/Ranking with DeepInterface

We suggest an extra step to validate and rank the predicted results from HMI-PRED server using our DeepInterface model [58]. DeepInterface was developed as a deep learning model to identify a docked complex as a valid protein-protein interaction interface or just a result of a mere structural coincidence.

3.3.1 Datasets

DeepInterface model is trained, validated and tested using datasets constructed from several resources including PIFACE [57], and interface structures from PDB for the positive dataset. On the other hand, docking decoy sets such as PPI4DOCK [61], and DOCKGROUND [60] were used for the negative dataset. To show how well the model generalize, the training set was constructed from different sources than validation and testing sets. The details of which resources were used for which datasets are shown in Table 3.1.

Table 3.1: DeepInterface training, validation and testing datasets resources

	PIFACE	PPI4DOCK	DOCKGROUND	PDB2012	Total
Training	110,248	110,332	-	-	220,580
Validation	17,542	17,458	-	-	35,000
Test	-	-	8,125	8,125	16,250
Total	127,790	127,790	8,125	8,125	271,830

The input data to DeepInterface is a preprocessed PPI interface structure. To prepare the input, first interfaces are extracted from the PPI structure. Then, to ensure that the interface structures are rotation & translation independent, the steps below were followed:

- Translate the midpoint between centroids of the two interface sides to the origin.
- Rotate the interface to align the line connecting the centroids to the x-axis.
- Rotate the interface until the furthest atom from origin points towards the positive z-direction.
- Construct a 40 x (25x25x25) matrix; 40 channels, each a 3D 25x25x25 cube. Every cell in the cubes represent a 2x2x2Å³ space. There are 20 channels for each

side of the interface; 19 are for storing the locations of the side chains of each residue type (the side chain is represented as a single $C\beta$ atom at the side chain's centroid), and the last channel is for each residue's $C\alpha$'s locations. The amino acids locations are considered as 3D Gaussian distributions with the exact coordinates as the mean while the standard deviations are derived from the van der Waals radii of each amino acid type. Gaussian distributions were used instead of crisp values to simulate the dynamic nature of proteins and to smooth out noise in data. Distributions are quantized to fill the cells of the cube.

Figure 3.14 **Error! Reference source not found.** illustrates the structure of the data after preprocessing, this data will be the input of DeepInterface. A matrix containing 40 channels/cubes of size (25x25x25) is created for every interface as shown in Figure 3.14a. As shown in Figure 3.14b, For each interface side, there are 20 'cubes', one to store $C\alpha$ atoms locations, and the remaining 19 'cubes' store the locations of $C\beta$ atoms of the residues. One cube is reserved for each residue type (except Glycine). Each $C\alpha$ atom and amino acid $C\beta$ atom location is expressed as a 3D Gaussian distribution as in Figure 3.14c.

3.3.2 Model

The model is a deep Convolutional Neural Network (CNN) with 4 convolutional layers followed by 2 fully connected layer. All convolutional layers are followed by batch normalization [81] and a rectifier linear unit (ReLU) [82] activation. The fourth convolutional layer is followed by a global average pooling to improve generalization. The last fully connected layer is followed by a softmax function which outputs two score values for both positive and negative classes. A dropout with 0.7 probability was applied to avoid overfitting. Figure 3.15 represents DeepInterface model architecture. The model was developed using a deep learning framework called "Knet" [83], which is implemented in Julia programming language [84].

DeepInterface model was trained using Adam [85] optimizer and a learning rate of 0.001. The data was divided into batches of size 32 and was shuffled in each epoch of training. We applied a learning rate decay with a factor of 2. Model weights were initialized using Kaiming [86] & Xavier [87] for the convolutional layers and the fully

connected layers, respectively. More details about the model's evaluation and training can be found in section (4.5).

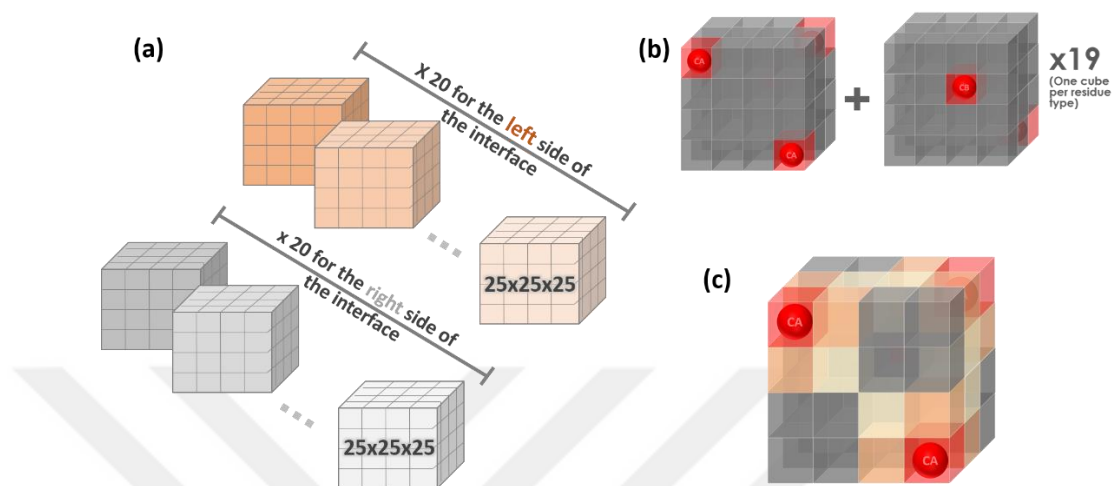


Figure 3.14: DeepInterface input data format.

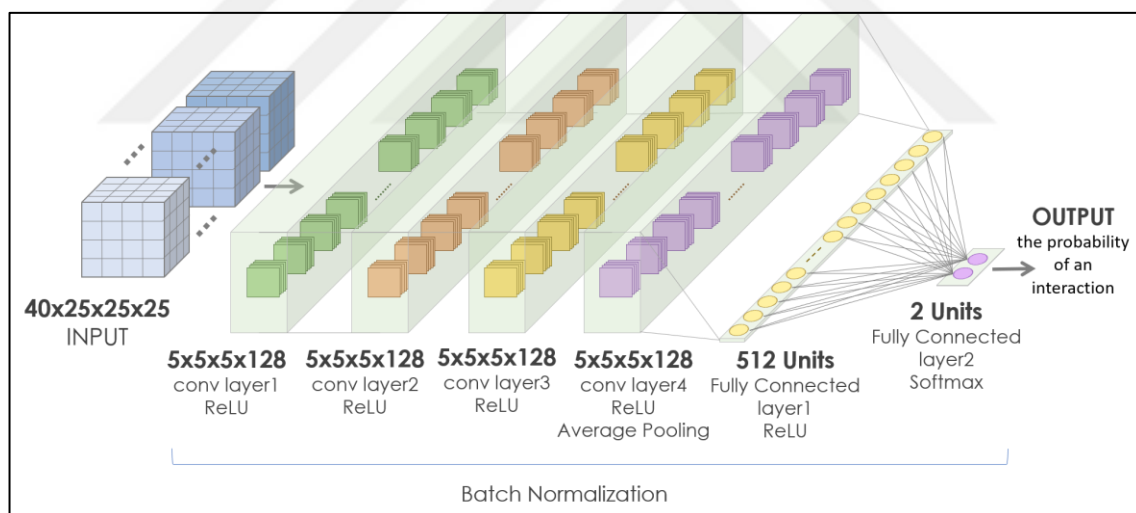


Figure 3.15: DeepInterface Model architecture

3.4 HMI-PRED Backend Design and Implementation

Software design principles and best practices were considered while implementing HMI-PRED webserver. Figure 3.16 summarizes the architecture and design of HMI-PRED webserver including frontend and backend parts.

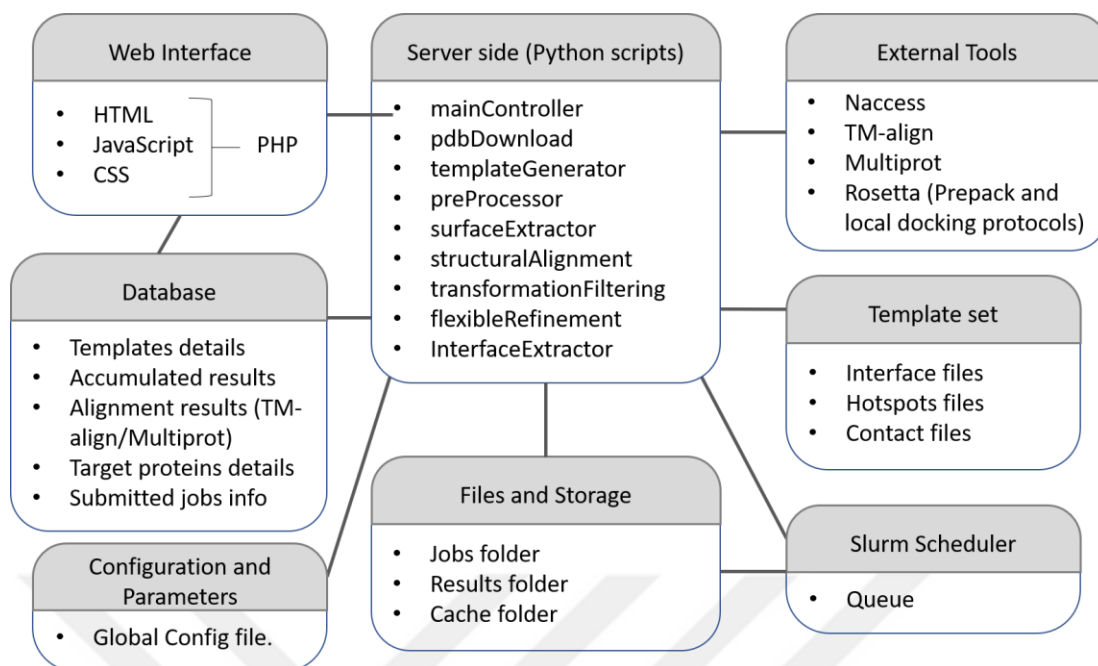


Figure 3.16: HMI-PRED webserver architecture

3.4.1 Design Principles

The webserver applies code-modularity by having several modules with low-coupling in between. Figure 3.16 outlines HMI-PRED server architecture. Server-side scripts are written in python, and they represent the different steps in HMI-PRED. Table 3.2 lists HMI-PRED's modules and a concise description of their functionality, input(s), and output(s).

The server uses a global configuration file (hmi.ini) to easily configure global parameters such as external tools paths and parameters, thresholds, and other parameters used to control job submission and results' storage.

HMI-PRED temporarily saves the intermediate results between modules to be able to recover from errors and restart the job from the point it reached before the error happened. Intermediate results are saved under different directories inside the job folder, which is named by the unique job ID. These temporary results are removed once the job done successfully.

Table 3.2: HMI-PRED Modules and their functionality

Protocol Step	Python Module(s)	Functionality	Inputs/Outputs
Preprocessing & Surface Extraction	pdbDownload	Checks validity of PDB and chain IDs provided by user and downloads the required PDB files.	Input: target PDB and chain ID. And the template list. Modify: Create target PDB file. Output: Validated template and target lists (invalid template codes removed)
	templateGenerator	Generates user-provided template interfaces (including their hotspots and contact files) in case they are not in the default template set.	Input: template list Modify: Create template interface, hotspot, and contact files. Output: a list of successfully generated template interfaces plus the templates which are already generated.
	preProcessor	Extracts target protein info (organism, Uniprot entry name) and saves it to DB. Extract the user-input target chain from the target pdb file.	Input: target protein (PDB and chain ID) Modify: Extracts the target chain only from target PDB. Output: target info (organism and Uniprot entry name).
	surfaceExtractor	Extract target's surface using Naccess.	Input: target protein (PDB and chain ID) Modify: Creates a PDB file of the extracted surface from target chain.
Structural Alignment	structuralAlignment	Performs one-to-all structural alignment between the target and template interfaces sides using TM-align or Multiprot based on the user choice. The alignment results are saved to DB	Input: target and template list Modify: saves one-to-all structural alignment results to database (including match count, matched residues, and alignment method and score). Output: A list of predicted interacting partners with their alignment results
Transformation & Filtering	transformationFiltering	Transforms the target to form the predicted complexes. And filters the predictions based on the TM-score (in case TM-align is used), match count, and hotspot matching criteria.	Input: A list of predicted interacting partners with their alignment results Modify: create a list of filtered predicted complexes with transformed target.
Flexible Refinement & Output Preparation	flexibleRefinement	Runs Rosetta's prepack and local docking protocols to refine and score the predictions. Then, it filters out the energetically unfavorable ones (i.e., having total energy > 0 or Rosetta I_sc > -5.0)	Input: A list of predicted interaction complexes. Output: A list of energetically favorable interactions with their details and energy scores.
	interfaceExtractor	Extract contact residues from the predicted complexes (can be downloaded by user). And prepares PDBs and JSON config files for 3D structures visualization.	Input: A list of predicted interaction complexes. Modify: Create a contact residues file. Create a PDB file of the superimposed structure, and create JSON config for 3D visualization

HMI-PRED is a template-based prediction tool, thus all templates data is stored in ‘template’ directory including interfaces, hotspots and contact residues, which are stored in ‘interfaces’, ‘hotspot’, and ‘contact’ subdirectories respectively.

3.4.2 Operating System and Server Setup

HMI-PRED is deployed as a LAMP stack on a CentOS 7 Linux server [88]. A **LAMP** stack is a webserver model where the server is set up using the original four open-source components: **L**inux OS, **A**pache HTTP Server [89], **M**ySQL relational database management system (RDBMS), and **P**HP programming language. However, these components are interchangeable and not limited to the original ones. In our server, we use MariaDB as our RDBMS, which is an open-source fork of MySQL developed by the community. The server is a high-performance computer (HPC) with 2 sockets and 14 cores per socket, resulting in 28 CPUs in total.

3.4.3 Job Scheduling

We used a workload manager ‘Slurm’ [90] to be able to serve multiple users simultaneously and schedule jobs while preserving fair resource allocation. Our server can handle up to 28 jobs running simultaneously, any job requests coming while the server is running in full capacity will be queued waiting for a CPU to be freed. Each job in the queue have the same priority and is independent of the other jobs. We also defined a job submission limit of 5 jobs per IP-address to give the chance to more users to run their jobs and ensure that our server will not be busy with single user’s jobs.

3.4.4 Database Design

HMI-PRED database is constructed to store the accumulated data from the jobs run with our server’s default template set and parameters. The database also keeps track of all submitted jobs related information and stores the intermediate results of the structural alignment done by TM-align (our default alignment tool) and Multiprot. Structural alignment results are serialized and stored in binary large object (BLOB) format. HMI-PRED database consists of 10 tables shown in Table 3.3. The database schema can be seen in Figure 3.17 and Figure 3.18. Following will be a detailed explanation of each table as well as the stored functions in the database.

Table 3.3: List of HMI-PRED DB Tables

Tables_in_hmiDatabase	
alignment_result	
ip_addr	
job	
jobs	
protein_info	
results	
target_info	
template_rosetta_score	
templates	
templates_hotspots	

jobs		job	
target	varchar(4)	job_id	varchar(30)
alignment_method	varchar(8)	target	varchar(4)
alignment_threshold	double	email	varchar(30)
alignment_result		ip_addr	varchar(45)
interface	varchar(6)	date_created	timestamp
chain	varchar(1)	ip_addr	
target	varchar(5)	ip	varchar(45)
alignment_method	varchar(8)	job_id	varchar(30)
content	blob		

Figure 3.17: HMI-PRED Database Schema (Jobs-related data, and intermediate results)

* The diagram was drawn using <https://dbdiagram.io>

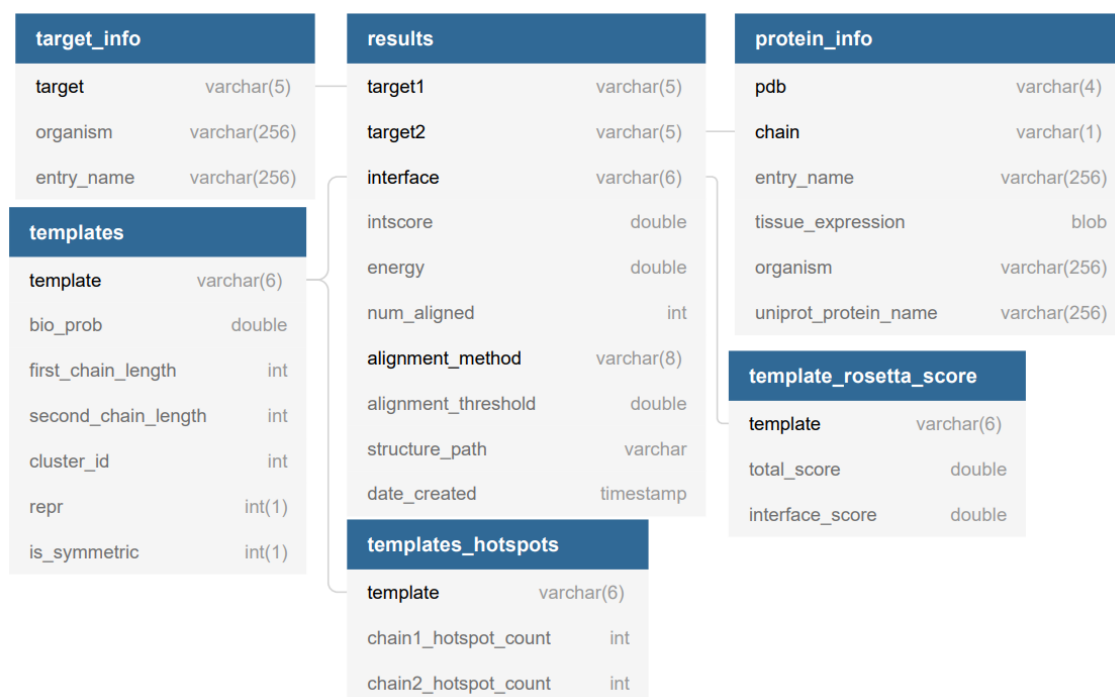


Figure 3.18: HMI-PRED Database Schema (Results-related data).

* The diagram was drawn using <https://dbdiagram.io>

Job Table

This table was created to keep track of all the jobs that are submitted to HMI-PRED server regardless of the user choices and input. Table 3.4 Describes ‘Job’ table’s fields and their datatypes. The primary key of the table is the ‘jobid’ field which stores the unique job ID of the submitted jobs. It also stores the input target and the email provided by the user (if any). The ‘date_column’ is automatically set to the current timestamp.

Table 3.4: Description of ‘Job’ Table in HMI-PRED DB

Field	Type	Null	Key	Default	Extra
jobid	varchar(30)	NO	PRI		
target	varchar(5)	NO			
email	varchar(30)	NO			
date_column	timestamp	NO		CURRENT_TIMESTAMP	
ip_addr	varchar(45)	NO			

Jobs Table

The ‘Jobs’ table (described in Table 3.5) stores the details of the successfully-done jobs which uses the default template set and the default TM-align threshold (in case TM-

align was picked as the aligner). The table records the input target and the alignment tool which was used.

Table 3.5: Description of 'Jobs' Table

Field	Type	Null	Key	Default	Extra
target	varchar(30)	NO	PRI	NULL	
alignment_method	varchar(30)	NO		TM-align	

Ip_addr Table

The purpose of this table is to restrict the maximum number of jobs which can be submitted by a single user (i.e., requests coming from the same IP address). The limit we use in HMI-PRED is 5 simultaneous jobs per user. The table temporarily stores the job ID and IP address for the currently running jobs. Once a job finishes, its entry is deleted from the table. Table 3.6 describes the columns of the table and their data types.

Table 3.6: Description of 'ip_addr' Table

Field	Type	Null	Key	Default	Extra
ip	varchar(45)	NO		NULL	
job_id	varchar(30)	NO	PRI	NULL	

Alignment_results Table

This table stores the intermediate results from the structural alignment stage. The table's primary key is composite of multiple fields (interface, chain, target, method) as shown in Table 3.7. The chain field here indicated which side/chain of the interface was aligned with the target for that entry's result. The content field is a binary BLOB field which stores the results of the alignment after being parsed and serialized by the structuralAlignment python module. The content field includes:

- i. The number of matched residues in the alignment.
- ii. The transformation matrix for the target protein.
- iii. Matching residues from both complexes.
- iv. TM-score (in case TM-align was used).

Table 3.7: Description of 'alignment_results' table

Field	Type	Null	Key	Default	Extra
interface	varchar(6)	NO	PRI	NULL	
chain	varchar(1)	NO	PRI	NULL	
target	varchar(5)	NO	PRI	NULL	
content	blob	NO		NULL	
method	varchar(30)	NO	PRI	NULL	

Target_info Table

This table stores the details related to the input target (see Table 3.8). The 'target' field is 5-letters length containing 4-letters PDB ID and the last letter is the chain ID of the target protein. The stored details include the target's organism and UniProt [74] entry name. These details are extracted from the PDB file of the target protein.

Table 3.8: Description of 'target_info' Table

Field	Type	Null	Key	Default	Extra
target	varchar(5)	NO	PRI	NULL	
organism	varchar(256)	NO		N/A	
entry_name	varchar(256)	NO		N/A	

Templates_hotspots Table

This table stores the hotspots count in each face of template interfaces. As shown in Table 3.9, each template has an entry in the table where the 6-letters template code is stored along with the hotspot count in both sides of the template interface.

Table 3.9: Description of 'templates_hotspots' Table

Field	Type	Null	Key	Default	Extra
template	varchar(6)	NO	PRI	NULL	
chain1_hotspot_count	int	NO		0	
chain2_hotspot_count	int	NO		0	

Templates Table

Details related to the template interfaces in the server's default template set is stored in the 'templates' table. Table 3.10 lists the fields of this table and their corresponding data types. Each entry has a unique 6-letters template code, EPPIC[75] biological relevance probability 'bio_prob', the interface's interacting chains' lengths in 'first_chain_length' and 'second_chain_length', a flag 'repr' to indicate whether the template is a representative of its cluster with the ID 'cluster_id', and a flag 'is_symmetric' to indicate whether a template is a symmetric homodimer (i.e., a symmetric homodimer has both sides of the interaction coming from the same protein and has a symmetrical structure around the interface center).

Table 3.10: Description of 'templates' Table

Field	Type	Null	Key	Default	Extra
template	varchar(6)	NO	PRI	NULL	
bio_prob	double	NO		0	
first_chain_length	int	NO		NULL	
second_chain_length	int	NO		NULL	
cluster_id	int	YES		-1	
repr	int(1)	YES		-1	
is_symmetric	int(1)	YES		0	

Template_rosetta_score Table

Rosetta interface score (I_sc) for the template PPIs are calculated to assess whether the microbial proteins will outcompete the bona fide partners of the microbe-targeted host proteins. These scores are stored in the 'template_rosetta_score' table (see Table 3.11). The total score represents the total interaction energy obtained from Rosetta, and 'interface_score' is Rosetta's I_sc.

Table 3.11: Description of 'template_rosetta_score' Table

Field	Type	Null	Key	Default	Extra
template	varchar(6)	NO	PRI	NULL	
total_score	double	NO		100000	
interface_score	double	NO		100000	

Protein_info Table

This table stores protein details about the individual faces/chains/partners of the template interfaces. The stored details include the proteins Uniprot entry name and protein name, its tissue expression, and the protein's organism. Each protein_info entry is identified by the PDB ID and the chain ID in which the protein's structure can be found. The description of the 'protein_info' table can be seen in Table 3.12

Table 3.12: Description of 'protein_info' Table

Field	Type	Null	Key	Default	Extra
pdb	varchar(4)	NO	PRI	NULL	
chain	varchar(5)	NO	PRI	NULL	
entry_name	varchar(256)	YES		NULL	
tissue_expression	blob	NO		NULL	
organism	varchar(256)	YES		NULL	
uniprot_protein_name	varchar(256)	YES		NULL	

Results Table

The 'results' table is the main table in HMI-PRED database. The table stores the accumulated results from various runs on the server. Only runs with default template set and default parameters are stored in the database. Each entry in the table has the following fields:

- i. 'target1' → The microbial part of the predicted interaction (i.e., input target)
- ii. 'target2' → The human (host) part of the predicted interaction.
- iii. 'interface' → The template interface which was used to predict the HMI.
- iv. 'energy' → Rosetta's total interaction energy of the predicted complex.
- v. 'structure path' → The path on the server where the predicted structure PDB file is stored (including the name of the file)
- vi. 'date_created' → The timestamp when the result has been obtained.
- vii. 'intscore' → Rosetta's I_{sc} for the modeled HMI.
- viii. 'num_aligned' → The number of matched/aligned residues during the structural alignment stage.
- ix. 'alignment_method' → The alignment tool used to obtain this result (currently, it should be either 'TM-align' or 'Multiprot').

- x. 'alignment_theshold' → This field is only relevant in the case of using TM-align as the aligner, and it stores the TM-score threshold.

Table 3.13: Description of the 'results' Table

Field	Type	Null	Key	Default	Extra
target1	varchar(5)	NO	PRI	NULL	
target2	varchar(5)	NO	PRI	NULL	
interface	varchar(6)	NO	PRI	NULL	
energy	double	NO		NULL	
structure_path	varchar(200)	NO		NULL	
date_created	timestamp	NO		CURRENT_TIMESTAMP	
intscore	double	NO		100000	
num_aligned	int	NO		0	
alignment_method	varchar(8)	NO	PRI	TM-align	
alignment_theshold	double	NO		0.25	

Stored Functions

HMI-PRED offers more results details to the user which can be calculated from already stored details. Therefore, these details were not stored directly to reduce the amount of consumed memory space. Instead, several functions were written to calculate these details upon request. Some examples of these details and their functions are:

- Weight**, which depends on the size of the aligned side of the template interface. The larger the size of the interface, the bigger its weight is: weight is 0.5 for interfaces with less than 30 residues ($n < 30$), 1 for $30 < n < 50$, 1.5 for $50 < n < 80$, and 2 for $n > 80$. A MySQL function 'calculate_weight' (see Figure 3.19) was written to perform this calculation.

```
CREATE FUNCTION `calculate_weight` (interfacelength INT)
RETURNS float
BEGIN
    DECLARE w FLOAT;
    IF interfacelength < 30 THEN
        SET w = 0.5;
    ELSEIF interfacelength < 50 THEN
        SET w = 1.0;
    ELSEIF interfacelength < 80 THEN
        SET w = 1.5;
    ELSE
        SET w = 2.0;
    END IF;
    RETURN w;
END
```

Figure 3.19: MySQL function to calculate the predicted complex 'weight'

- ii. **Scores 1-3**; *Score 1* is calculated by the multiplication of the prediction weight by the percentage of matched residues during structural alignment. *Score 2* represents the multiplication of *Score 1* with the EPPIC "Probability of being a valid biological interface". *Score 3* can be calculated by multiplying *Score 2* with the Rosetta I_sc score of the modeled HMI. The lower *Score 3* is, and the higher *Score 2* is, the more confident we are that the HMI model is likely to occur because it offers a better mimic of the template endogenous biological interfaces. Figure 3.20 lists the functions used to calculate these scores upon request.

3.5 HMI-PRED Frontend Design and Implementation

3.5.1 Responsive Design

HMI-PRED is a novel webserver for HMI 3D structural-based prediction and is expected to be used widely from scientists and researchers in the field. Therefore, we developed a publicly-available user-friendly webserver. The user interface of the server enables the user to submit their jobs with clear and easy-to-follow directions. In addition, users can check the details of the default template set we use in the server as well as being able to search and fetch the accumulated results in the database in a neat and organized way.

```
CREATE FUNCTION `calculate_score1` (interfacelength INT, matchpercentage FLOAT)
RETURNS float
BEGIN
    DECLARE s1 FLOAT;
    SET s1 = matchpercentage / 100.0 * calculate_weight(interfacelength);
    RETURN s1;

CREATE FUNCTION `calculate_score2` (interfacelength INT, matchpercentage FLOAT,
                                   bioprob FLOAT)
RETURNS float
BEGIN
    DECLARE s1 FLOAT;
    DECLARE s2 FLOAT;
    SET s1 = calculate_score1(interfacelength, matchpercentage);
    SET s2 = s1 * bioprob;
    RETURN s2;
END

CREATE FUNCTION `calculate_score3` (interfacelength INT, matchpercentage FLOAT,
                                   bioprob FLOAT, intscore FLOAT)
RETURNS float
BEGIN
    DECLARE s2 FLOAT;
    DECLARE s3 FLOAT;
    SET s2 = calculate_score2(interfacelength, matchpercentage, bioprob);
    SET s3 = s2 * intscore;
    RETURN s3;
END
```

Figure 3.20: MySQL Functions to calculate Scores 1-3 for the predicted complexes.

To ensure a responsive design of the web interface for HMI-PRED, we used Bootstrap 4.0 framework [91]. Bootstrap is a famous (HTML, CSS, JavaScript) framework for developing responsive and flexible web pages in less effort and time. The framework was developed initially at Twitter to promote consistency across their internal tools. Later, Bootstrap was made available to public as an open-source framework which is compatible with all modern browsers including Firefox, Chrome, Internet Explorer, Safari, ...etc. Bootstrap offers (HTML and CSS)-based design templates for common user interface components such as forms, navigations, buttons, dropdowns, modals, alerts, tooltips, ...etc.

In our design of HMI-PRED's web interface, we made use of most of the template components offered by Bootstrap to ensure a responsive design. A responsive user-interface ensures that all the interface components of the webpage automatically adapts to the screen size and orientation, which provides the user with the best experience.

The webserver also offers a HELP (on/off) switch at the top of the page, to allow the user to enable and disable popup dialog windows that guide the user through his/her through job submission. Also, user input is validated before submission and informative alerts are shown to the user in case (s)he enters invalid input.

For the users to keep track of the jobs they submit, they can (optionally) provide their email address to receive notifications once their jobs are submitted and completed. However, this is an optional input, users can submit jobs anonymously and track their job status from the link provided to them once they submit the job.

3.5.2 3D structures visualization

HMI-PRED provides the user with three visualization options of the predicted structures:

- i. Visualization of the original predicted structure (see Figure 3.21a).
- ii. Visualization of the predicted structure with highlighted interface residues; two residues are marked as interface residues when they are each from a different face/chain of the interaction and are within a cutoff distance (the sum of their van der Waals radii + 0.5 Å) from each other (see Figure 3.21b).
- iii. Visualization of the predicted structure with the mimicked template interface protein superimposed on it. For example, if we have the predicted interaction

between the target ‘1y6mL’ and the host protein ‘3a5B’ mimicking the template interface ‘3a5AB’, we superimpose the mimicked protein ‘3a5A’ on the predicted structure to allow the user to examine how well the microbial protein hijacks the interface on the human protein (although they have different global structures) (see Figure 3.21c).

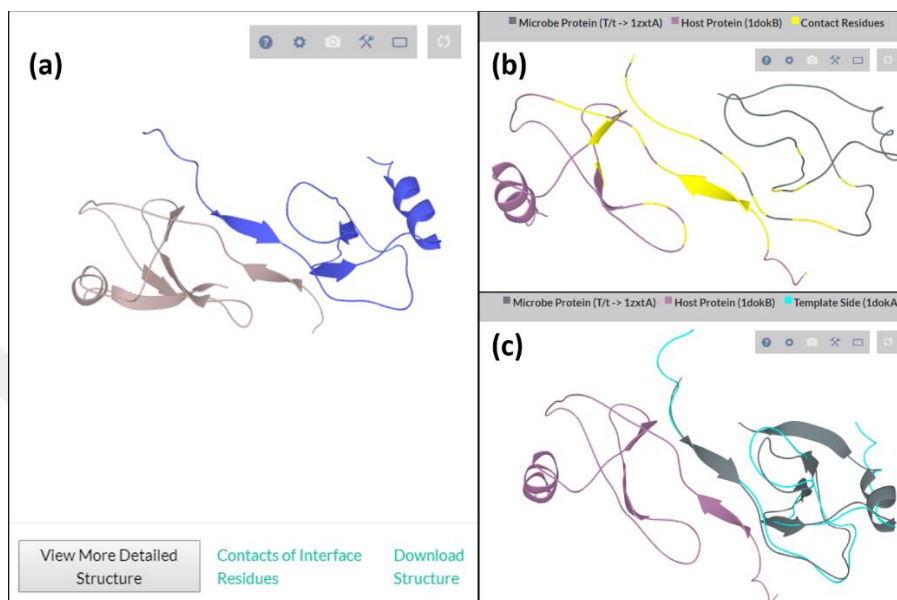


Figure 3.21: Different 3D Visualization Options Offered by HMI-PRED

These visualizations were constructed using PDB-LiteMol [92]; a customized AngularJs [93] Web-Component for LiteMol [94] implemented by Mandar Deshpande. LiteMol is a plugin written in TypeScript [95] and compiled to JavaScript. It allows the user to visualize 3D structures of molecules interactively, where (s)he can for example rotate, change colors, and change the visualization type of the structure (c- α trace, Balls and Sticks, Surface, ...).

In order to customize the visualization and apply accumulative formatting on the 3D structure, a custom theme for PDB LiteMol was used [96]. This customized theme requires a JSON configuration file to be provided which contains the formatting details for the visualized molecule's residues.

Chapter 4:

RESULTS AND DISCUSSION

4.1 *HMI-PRED Webserver Usage*

HMI-PRED server's input is the user-provided PDB and chain IDs of the microbial (target) protein or user-uploaded 3D structure (model) of the microbial protein in a PDB format. The target protein should be a monomer (i.e., single chain). The user can also choose to run the protocol with a single template interface or a subset of up to 100 template interfaces instead of using the whole default template set. A template interface has a 6-alphanumeric digits code where the first four characters represent the PDB ID of the complex from which the interface was extracted, and the last two digits represent the chain IDs of the interacting partners (e.g.: '3d2uAB' is the code of the interaction interface between chain A and chain B in the structure with the PDB ID '3d2u').

HMI-PRED has five main pages:

1. The main "HMI" page, where users submit their jobs.
2. The "Predictions" page where users can access the results of previous analyses.
3. The "Templates" page where the details of each template can be found including the organism name, protein name and tissue expression data for each constituent face of the interface.
4. The "Tutorial" page, where the detailed instructions on how to use the server can be found.
5. The "References" page, where relevant publications and "how to cite" information can be found.

4.1.1 *HMI-PRED Main Page*

This is the query submission page (Figure 4.1). If the microbial protein has a resolved structure available in PDB, user enters its PDB and chain IDs. If it doesn't have available structure, user can upload the homology model in PDB format by clicking on the "Upload Target" button. In the default settings, our template interface set and TM-align will be used. If the user wants to use the default template set and settings, s/he doesn't need to change anything in "Template" and "Structural Alignment Method" parts; just entering the email address and clicking the "Submit" button is enough. The query will be placed

in the HMI-PRED server queue. Users will be provided with a URL to track their job status. If users provide their emails, they will be notified upon start and completion of the job. When the run is completed, the user will receive a notification email with a URL which will take him/her to the results page.

If the user wants to test only a subset of interfaces or an interface that is missing in our template set, such as an interface deposited to PDB after January 3rd 2019, a list of up to 100 interfaces (PDB IDs with two chains, e.g. 1a0nAB) can be provided. HMI-PRED automatically downloads the corresponding PDBs and extracts the interfaces and hotspots. In addition, if the user wants to try only a few interfaces instead of aligning 17,351 interfaces (i.e., the entire default template set), s/he can give up to 100 interfaces. The default structural alignment tool is TM-align and the default alignment threshold is 0.25. The user can change the threshold: Higher thresholds will result in more stringent alignment. User can also switch from TM-align to Multiprot.

Depending on the size of the target protein and number of matched interfaces, computation time can take several hours, but on average the results are returned to the user within 12-24 hours. If the target protein was analyzed before, the results from the previous query will be already in the database (i.e. “Predictions” page) and HMI-PRED will instantly direct the user to these, instead of re-modeling the HMIs.

The screenshot shows the HMI-PRED web interface. At the top, there are logos for NIH National Cancer Institute and KOC UNIVERSITY, and the title 'HMI-PRED'. Below the title is a navigation bar with links: HMI, Predictions, Templates, Tutorial, References, and a HELP icon. The main form is titled 'Input the Microbe Protein:' and has a 'Target:' input field. To the right of the 'Target:' field is a section for 'OR Upload Target:' with a 'Choose File' button and a 'No file chosen' status. Below this is an 'Optional Inputs:' section. It includes a 'Template (optional):' section with radio buttons for 'Use the default template set' and 'Choose a template manually'. Under 'Choose a template manually' is an 'Upload a list of template interfaces' button and a 'Choose File' button. There is also a 'Structural Alignment Method:' section with radio buttons for 'TM-align' (selected) and 'Multiprot'. Below this is an 'Advanced Options' button. Further down is a 'TM-align Score Threshold (0 to 1):' input field with the value '0.25'. At the bottom of the form is an 'Email(optional):' input field. At the very bottom are 'Submit' and 'Reset' buttons. Red callout boxes with arrows point to various parts of the form: 'Enter PDB & chain IDs of microbial protein Example: 1m3D' points to the 'Target:' field; 'User can give up to 100 interfaces Example: 3se3AB' points to the 'Upload a list of template interfaces' button; 'User can change the alignment tool or the alignment threshold' points to the 'Structural Alignment Method:' radio buttons and the 'TM-align Score Threshold' field; 'When job is started & completed, user will receive an email with a URL of the results' points to the 'Email(optional):' field; and 'If microbial protein has no PDB, upload the homology model' points to the 'Choose File' button in the 'Upload Target' section.

Figure 4.1: HMI-PRED main page

4.1.2 Predictions Page

The results of all queries with the default settings and PDB identifiers (non-homology model targets) as an input are stored by the server for future access. However, the results of the homology models as targets or the non-default settings will not be stored. Results obtained through TM-align and Multiprot are stored separately. The default page shows TM-align results, but the user can switch to Multiprot by clicking on the “Multiprot” box.

These results can be accessed in the Predictions page (Figure 4.2). The page in Figure 4.2a shows a typical results table; For each prediction, the PDB and chain IDs of the target, the microbial organism to which this target protein belongs, microbial protein name, its predicted interactor in the host, mimicked/disrupted human PPI (template PPI), I_{sc} for the HMI and I_{sc} of the template PPI, and tissue expression of the host protein are listed. If the I_{sc} of the HMI is lower than the template PPI, it suggests that the microbial protein has higher affinity to the human protein than the human protein's bona fide partner in the template PPI. Users can sort and search the results by microbial organism, PDB_IDs of the microbial/target protein and template interface. Clicking on the “View” button in the last column of the results tables opens a pop-up page for visualizing the HMI model interactively as in Figure 4.2b. The user can rotate, change colors, download the structure in PDB format (“Download Structure” button), retrieve the interface residues of the HMI (Figure 4.2c), visualize the HMI structure with highlighted interface residues (Figure 4.2d), and visualize the superimposed structures of the HMI and the template PPI (Figure 4.2e). Superimposed structure will show how well the microbial protein hijacks the interface on the human protein (although they have different global structures).

For a given target, the Results table is also downloadable. There are additional details in the downloaded file, such as percent match, weight of the interface, and probability of template interfaces being biological interfaces. Moreover, the results table of each microbial protein has “Functional annotation of microbial protein targeted human proteins” box (Figure 3.5). The KEGG pathways enriched with the targeted host proteins will provide the user a broad picture of the major human pathways impacted by the microbe.

The View button allows users to interactively visualize the HMI complex structures with a pop-up window. The structure and interface residues of the HMI model can also

be downloaded through “Download Structure” and “Contacts of Interface Residues” links in the visualization window. “View More Detailed Structure” button opens another page, where the user can visualize the structures of the HMI model with highlighted interface residues and superimposed HMI and the corresponding template PPI structures. Both the HMI and the superimposed structures are downloadable.

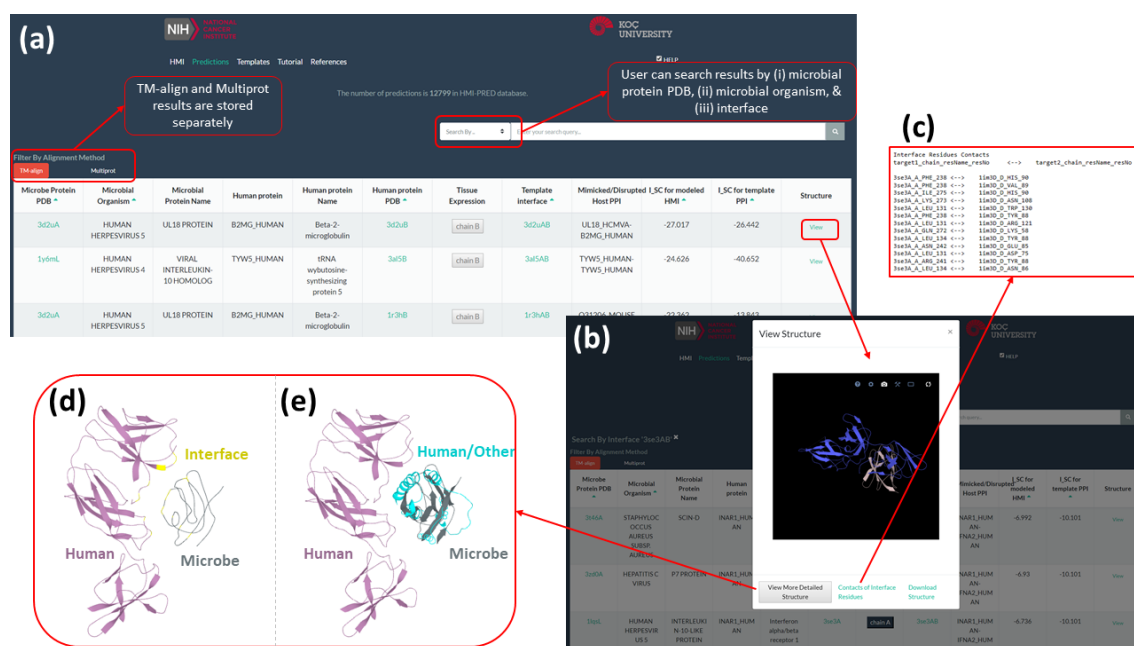


Figure 4.2: “Predictions” tab, a repository for previous results.

4.1.3 Template Page

The default template set and the details of each protein in the interface can be browsed and searched in this page (Figure 4.3). The first column shows the representative interface of the cluster that has structurally similar interfaces corresponding to the same human PPI. Second, third, and fourth columns show the organism, protein name, and the tissue expression of the constituent faces of the interface. Last column shows the number of members in the corresponding interface cluster. Clicking on the “Members” column link will display all members of this cluster. Interacting residues of the interface can be accessed by clicking on the interface ID.

NIH NATIONAL CANCER INSTITUTE

HMI Predictions **Templates** Tutorial References

KOC UNIVERSITY

HELP

The number of template clusters is 17351 in HMI-PRED database.

User can search a particular interface Example: 3se3AB

Search By... Enter your search query...

We clustered the redundant interfaces corresponding to the same human PPI based on structural similarity. This link shows the members of a given interface cluster

Representative	Organisms	Protein Names	Tissue Expression	Members
13gsAB	Homo sapiens - Homo sapiens	GSTP1_HUMAN - GSTP1_HUMAN	chain A - chain B	71
1a02FJ	Homo sapiens - Homo sapiens	FOS_HUMAN - JUN_HUMAN	chain F - chain J	4
1a0IBC	Homo sapiens - Homo sapiens	TRYB2_HUMAN - TRYB2_HUMAN	chain B - chain C	16
1a0nAB	Homo sapiens - Homo sapiens	P85A_HUMAN - FYN_HUMAN	chain A - chain B	1
1a12AB	Homo sapiens - Homo sapiens	RCC1_HUMAN - RCC1_HUMAN	chain A - chain B	2
1a12AC	Homo sapiens - Homo sapiens	RCC1_HUMAN - RCC1_HUMAN	chain A - chain C	1
1a15AB	Homo sapiens - Homo sapiens	SDF1_HUMAN - SDF1_HUMAN	chain A - chain B	5
1a1aAB	Homo sapiens - Homo sapiens	SRC_HUMAN - SRC_HUMAN	chain A - chain B	7
1a1mAB	Homo sapiens - Homo sapiens	1B53_HUMAN - B2MG_HUMAN	chain A - chain B	2
1a1nAB	Homo sapiens - Homo sapiens	1B35_HUMAN - B2MG_HUMAN	chain A - chain B	51
1a1uAC	Homo sapiens - Homo sapiens	P53_HUMAN - P53_HUMAN	chain A - chain C	1
1a22AB	Homo sapiens - Homo sapiens	SOMA_HUMAN - GHR_HUMAN	chain A - chain B	7

Figure 4.3: “Templates” tab.

4.1.4 Downloadable Results Folder for Each Target

When a job is completed, the user will receive an email with a URL which will take him/her to the results page. At the bottom of the results page, there is “Download Prediction Results” box. This box can be seen also for each individual microbial protein searched in the “Predictions” page. Clicking on this box will download a folder which has:

1. A table, “results.txt”, listing the details of all HMI models for a given target. The details include:
 - a. Microbe Protein PDB
 - b. Microbial Organism
 - c. Microbial Protein Name
 - d. Human protein
 - e. Human protein Name
 - f. Human protein PDB
 - g. Template interface
 - h. Mimicked/Disrupted Host PPI
 - i. I_SC for modeled HMI
 - j. I_SC for template PPI
 - k. Number of aligned residues

- l. Length of template interface: Number of residues on the aligned face of the interface.
 - m. %Match: Ratio of the number of aligned residues to the total number of residues in the aligned face of the interface.
 - n. Human Protein Tissue Expression: Tissue expression of the un-aligned face of the interface, i.e. the partner of the microbial protein. Tissue expression information is obtained from the Human Protein Atlas [5, 6].
 - o. Weight: Each face of the template interface has a weight depending on their sizes, such that bigger faces have larger weights. The given weight information in this table is the weight of aligned face of the interface. If the aligned face has less than 30 residues ($n < 30$), the weight is 0.5; if $30 < n < 50$, weight is 1; if $50 < n < 80$ weight is 1.5; and if $n > 80$, the weight is 2.
 - p. Probability of being a biological interface: Obtained from EPPIC (Evolutionary Protein-Protein Interface Classifier) [1].
 - q. Score1: Calculated by the multiplication of "% Match" with the "Weight".
 - r. Score2: Calculated by the multiplication of "Score 1" with the "Probability of being a biological interface".
 - s. Score3: Calculated by the multiplication of "Score 2" with the "Rosetta I_{sc} for the modeled HMI". The lower the Score3 is, the more confidence we have that the HMI model would have favorable interaction energy and be likely to occur since they are better mimics of the real biological endogenous interfaces.
2. Structures of HMI complexes
- E.g. "1awbAB_2f68X_0_1awbB_0.rosetta_0001_0001.pdb": Here, the microbial protein is "2f68X.pdb" and its binding partner in the HMI complex is "1awbB.pdb". This HMI is predicted through the template interface "1awbAB". The microbial protein is aligned with the "A" face of the interface and interacts with the un-aligned "B" face.
3. Structures of superimposed HMI and the template PPI
- E.g. "1awbAB_2f68X_0_1awbB_0.rosetta_0001_0001.pdb_superimposed.pdb"
4. Interface residues of the HMI models
- E.g. "1awbAB_2f68X_0_1awbB_0.rosetta.intRes.txt"

4.2 Accuracy of Structural Alignment and Scoring Function

Due to limited experimental structural HMI data, it is difficult to calculate the exact false positive and false negative rates of our predictions. To demonstrate the accuracy of the structural alignment and scoring function that we employ in our method, we calculated the number of true positives by testing the recovery of all known HMIs (606 of them) in our template set. It is important to note that these HMI interfaces may not be all biological interfaces, but rather crystal artifacts. In fact, 290 of them either have EPPIC score smaller than 0.3 or no EPPIC score, meaning that their probability of being real biological interfaces is very low [75].

As Figure 4.4 illustrates, we were able to recapitulate 94% of the known HMIs (570 of 606) through structural alignment and 81% of the correctly aligned HMI cases (462 of 570) through scoring function, Rosetta (Check Appendix B for detailed results of the accuracy test which we ran). TM-align failed to align 36 of the interfaces correctly (i.e., did not achieve exact alignment with TM-score of 1.0) and they are not passed to Rosetta for scoring. Rosetta gave unfavorable I_{sc} (i.e., $I_{sc} > -5.0$) for 107 cases, 89 of which are non-biological interfaces according to EPPIC [75]. So, it is reasonable that Rosetta identified these HMIs as unfavorable. One case (template interface ID: 1jew1R) failed due the clashing residues: HMI-PRED tolerates up to 5 clashes, but in this case, the original interface has 7 clashes. In addition, some of the failed HMIs are multi-protein complexes in the asymmetric unit of the crystal and only a binary interface only may not be sufficient to stabilize the interaction. This could also be the reason for unfavorable I_{sc} . Figure 4.5 demonstrates examples for such multiprotein complexes and binary interfaces of these complexes.

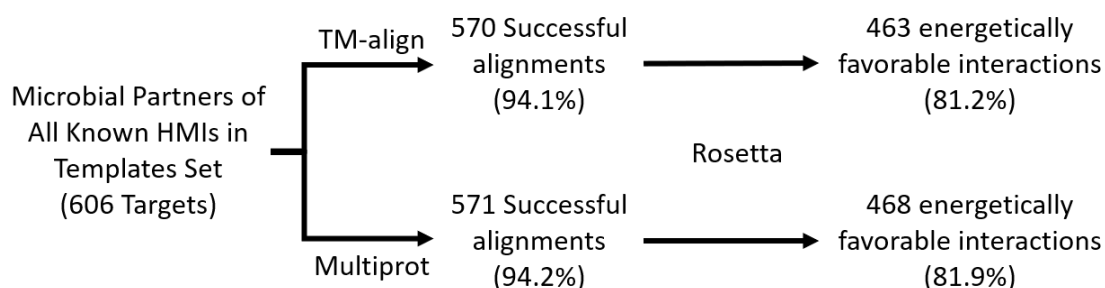


Figure 4.4: Accuracy of Structural Alignment and Scoring Functions

We also tested known HMIs by utilizing “Multiprot” [71] which is an alternative structural alignment tool available in HMI-PRED. Multiprot correctly aligned 571 of 606

cases and failed at 35 cases, 11 of which are common to TM-align failures. Details of Multiprot and TM-align failures are given in Appendix D. Multiprot tends to fail for smaller interfaces. TM-align and Multiprot complements each other: for majority of the cases, they correctly align the interfaces and fail at few. Although they agree in most of the cases, they also differ at some. So, users may want to try both alignments to double-check the HMI results or predict additional HMIs through a second alignment method.

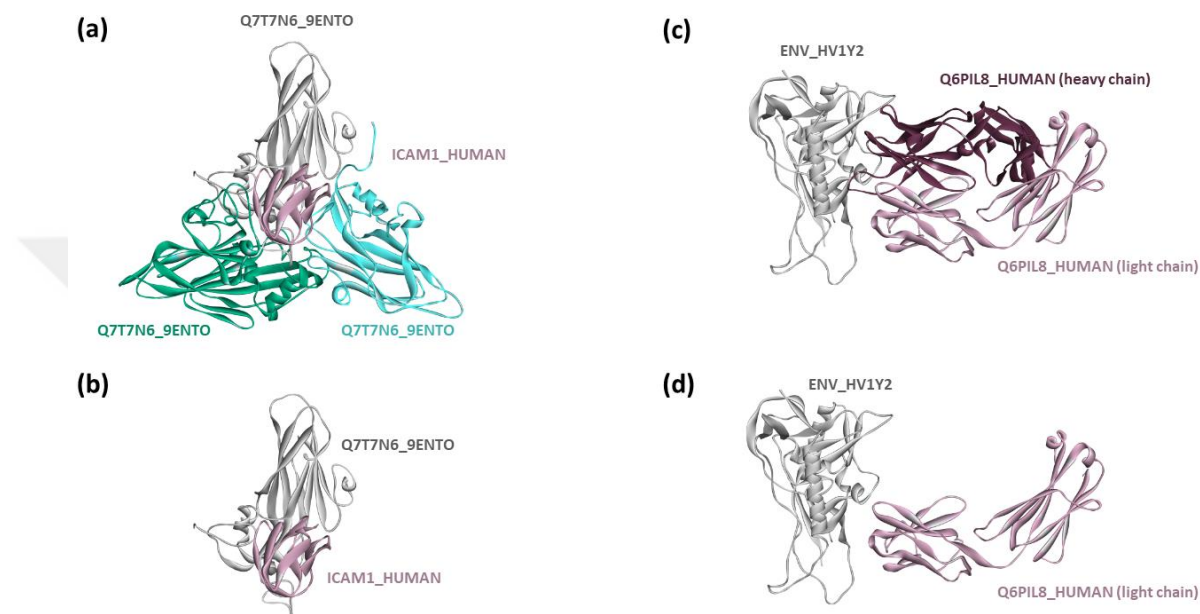


Figure 4.5: Examples of multiprotein complexes in a crystal

(a) Human ICAM1 is in contact with three human-coxsackievirus A21 proteins in the asymmetric unit of a crystal (PDB_ID: 1z7z, chains 1, 2, 3, & l). (b) The binary HMI interface in our template set (interface ID: 1z7z1l). Since this interface is small compared to the multiprotein interface, it is not enough to hold the constituent proteins together and Rosetta gives unfavorable I_{sc} for this interaction. (c) Human immunodeficiency virus envelope glycoprotein gp120 is in contact with two human proteins – light & heavy chains of F105 antibody in the asymmetric unit of the crystal (PDB_ID: 3hi1, chains A, J, & B). (d) Binary interaction between gp120 and light chain of F105 does not have a favorable I_{sc} (interface ID: 3hi1AJ).

4.3 Clustering Optimization

As mentioned before, structural clustering was applied on the template set to reduce redundancy and speed up predictions. In Chapter 3, two clustering heuristics were discussed: CLUSTER_V1 and CLUSTER_V2. We compared the performance and the optimality of both heuristics and notice that CLUSTER_V2 offers a more optimal solution; thus, it was used as our clustering algorithm. First, we compared manually some

small samples of our graphs (i.e., PPI groups) as shown in Figure 4.6. The figure shows that in all the sample cases, CLUSTER_V2 performs as good as the optimal solution unlike CLUSTER_V1. Table 4.1Figure 4.6 lists the size of the initial template set and the reduced sizes after applying both heuristics. CLUSTER_V2 shows a more optimal result (i.e., more minimization in the representatives set).

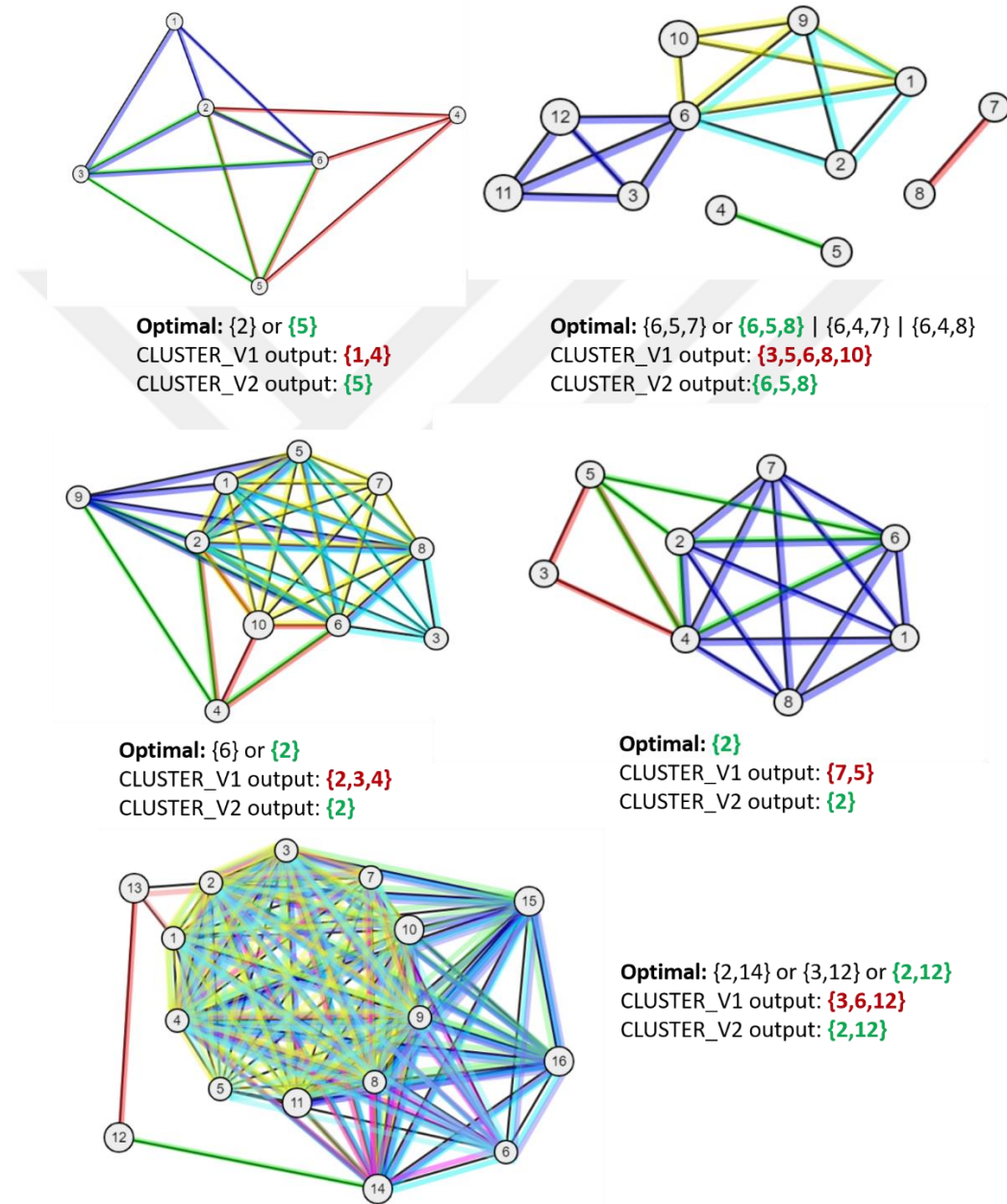


Figure 4.6: Optimality Comparison of Clustering Heuristics

Table 4.1: Comparison between clustering algorithms on the full template set.

	CLUSTER_V1	CLUSTER_V2
Full templates list size	54,897	
# of PPI groups/graphs	5,368	
# of representatives	18,428	17,186

4.4 Case Studies

4.4.1 Pathogenic bacteria protein with resolved 3D structure as an input

Mycobacterium tuberculosis secretes a toxin, “Tuberculosis necrotizing toxin” (TNT), through which it kills human macrophages by inducing necrosis with an unidentified molecular mechanism. A recent study found that TNT depletes NAD⁺ in human cells by hydrolyzing it. This catalytic function is vital for TNT-induced cell death [97]. Mycobacterium tuberculosis itself is immune to TNT because it has the antitoxin, TNT immunity factor (IFT) that binds to TNT and inhibits its enzymatic activity. The structure of TNT-IFT complex has been resolved (PDB_ID: 4qlpAB).

HMI-PRED identified 36 potential HMIs for TNT. The results can be accessed in the “Predictions” tab, by filtering according to “Target” with 4qlpB (PDB_ID of the target protein). Among the TNT-targeted host proteins, there is NADH dehydrogenase 1 beta subcomplex subunit 9 (NDUB9). NDUB9 and IFT bind to overlapping interfaces on TNT (Figure 4.7a-c). NDUB9 has important roles in electron transport chain and it catalyzes oxidation of NADH to NAD⁺. This result may suggest that TNT may deplete NAD⁺ not only through hydrolysis of NAD⁺, but also through inhibiting the NDUB9 dehydrogenase by directly interacting with it and hampering production of more NAD⁺. Identification of the potential targets in the host, along with their bound structures provide mechanistic insight into the still elusive disease mechanisms.

4.4.2 Commensal bacteria protein with homology model as an input

Microbial anti-inflammatory molecule (MAM) protein from Faecalibacterium prausnitzii – one of the most abundant commensal bacteria in healthy individuals and deficient in Crohn’s disease patients – is able to inhibit the NF- κ B pathway [98]. MAM

protein does not have a resolved structure. We homology modeled it with I-TASSER [99] and modeled its HMIs. HMI-PRED predicted 35 HMIs for this protein (**Error! Reference source not found.**). MAM-targeted host proteins include polyubiquitin-C (UBC), Ubiquitin-like modifier-activating enzyme 1 (UBA1) and Ubiquitin conjugating enzyme E2N (UBE2N, Ubc13)

Ubiquitination is a reversible posttranslational modification and is integral to activation of the NF- κ B pathway. Distinct ubiquitination modes may lead to opposing results: while K48-linked ubiquitination targets a protein for degradation, K63-linked ubiquitin chain serves as a docking platform for assembly of large protein complexes. For instance, K63-linked ubiquitin chain attached to TRAF6 recruits the downstream players, TAB2/3 and NEMO (IKK- γ), which is vital for NF- κ B activation. UBE2N-UB2V2 ubiquitin conjugating enzyme complex catalyzes the addition of K63-linked to TRAF6.

Since this job used a homology model as the input, the results are not stored in the “Predictions” tab. We found that MAM can hijack the interface on UB2V2 to bind to UBE2N and impede UBE2N-UB2V2 enzyme complex (Figure 4.7d-f). Consequently, this could result in blocking NF- κ B activation, and production of pro-inflammatory cytokines. Our HMI complex may provide the conceptual foundation of the anti-inflammatory functions of MAM protein at the atomistic level and why its absence contributes to chronic inflammation in Crohn’s disease patients. These two case studies demonstrate the ability of HMI-PRED to identify disease-related proteins or implicate new proteins in disease pathogenesis.

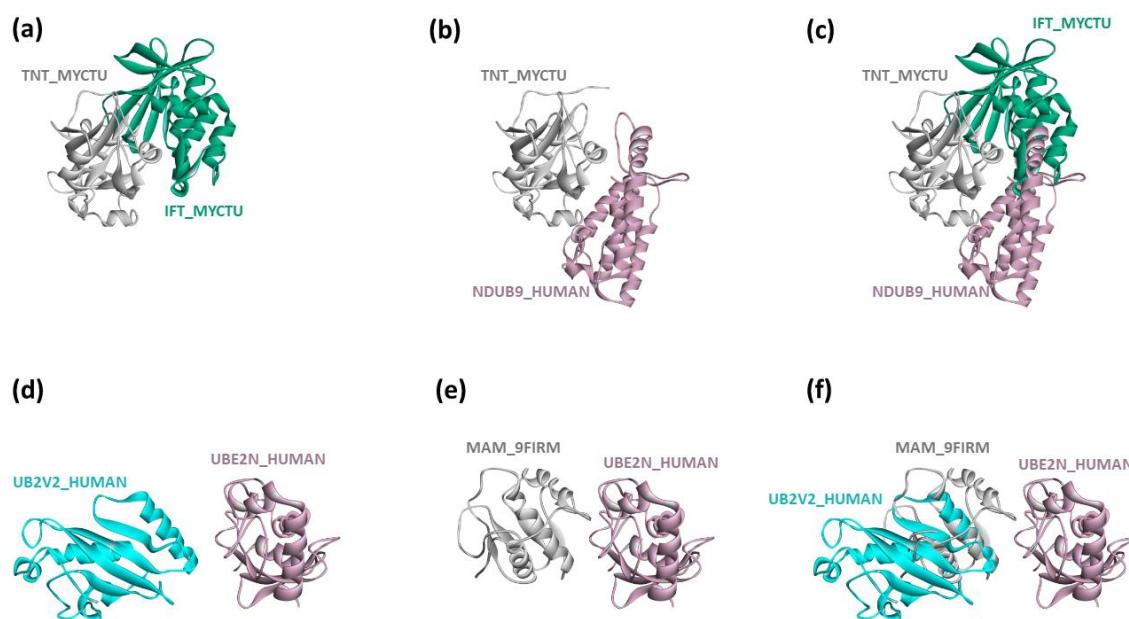


Figure 4.7: Case studies.

(a) *Mycobacterium tuberculosis* toxin, TNT, in complex with its immunity factor (IFT) (PDB_ID: 4qlpAB). (b) TNT-NDUB9 interaction complex that HMI-PRED modeled. (c) Superimposed structure of TNT-IFT and TNT-NDUB9 shows that IFT and NDUB9 bind to overlapping interfaces on TNT. (d) Human ubiquitin conjugating enzyme complex UBE2N-UB2V2 (PDB_ID: 1j7dAB), which assembles polyubiquitin chains to target proteins, such as TRAF6 in NF- κ B pathway. (e) MAM-UBE2N host-microbe interaction complex model. (f) Anti-inflammatory MAM protein can bind to UBE2N by mimicking the interface on UB2V2 and potentially disrupt UBE2N-UB2V2 interaction, impairing the ubiquitin conjugating activity.

4.5 DeepInterface Performance Analysis

As listed in Table 3.1, DeepInterface model was trained initially using structures from PIFACE dataset as the positive training samples and PPI4DOCK for negative samples. The total size of the training dataset was 255,580 interfaces. The validation dataset had 35,000 interfaces which were uniformly sampled from the training set. The validation set was used to evaluate the model during the training in different epochs to pick the model with the best performance. We trained the model for 3 epochs, after which the validation accuracy converged to 88%.

For each input (i.e., interface structure), the model outputs two scores, one for being a positive/valid prediction, and the other score is for being a negative/invalid prediction. Our decision variable is the difference between the output score (diff-score = positive_score – negative_score); when diff-score has a positive value, it means that the input represents a valid prediction.

Figure 4.8 illustrates the distribution of the output scores (positive and negative score) and the decision variable `Diff_score` of the decoys (i.e., incorrect complexes) of an interface for two cases: 2pukAB and 3c0aAB. The distribution shows that, as expected, the negative scores for most decoys are bigger than the positive scores. The black bar in both plots indicates the value of `diff-score` of the valid interface. The `diff-score` distribution (in yellow) highlights the frequency of native-like decoys/predictions when `diff-score` > 0 (i.e., false positives) or invalid interface predictions when `diff-score` < 0 (i.e., true negatives). Between these two cases, the model performed better for 3c0aAB. These cases are provided to explain how we assign a single score (`diff-score`) to a prediction.

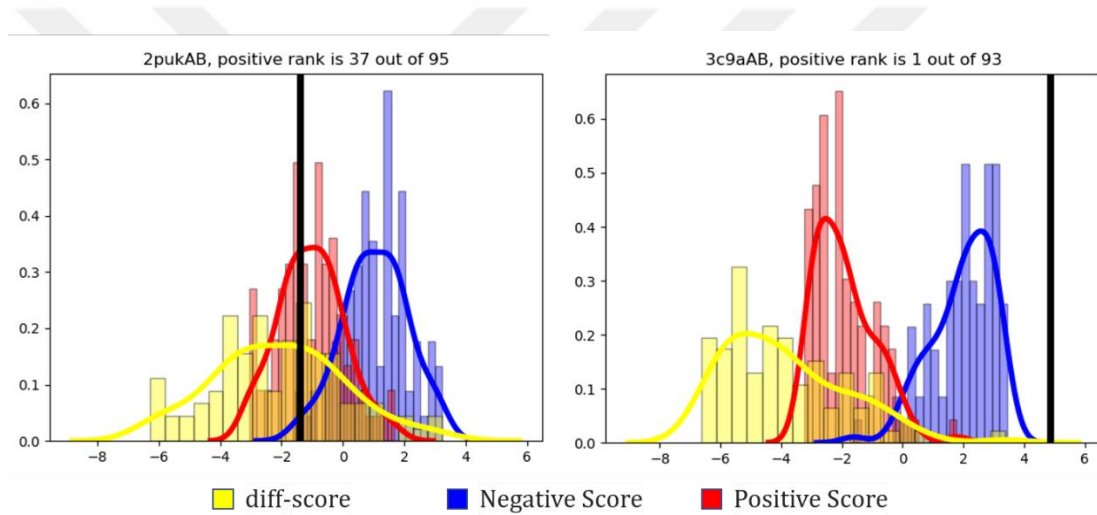


Figure 4.8: Sample distributions of DeepInterface scores for two complexes

Model evaluation details are listed in Table 4.2 and Table 4.3 provides performance metrics for the training, validation and testing data separately. Training accuracy is 0.92, while validation accuracy is 0.88, and for testing (i.e., unseen data) accuracy is 0.75.

Zhao, et al [100] offers a comparison of about 60 different protein interface prediction methods in their recent review. The accuracies range between 85.4% - 45.0%. However, it is hard to compare these methods directly since they use different datasets and measurements. Our validation and test accuracy results are both well above the average of Zhao, et al reported results. It can also be noted from Table 4.2 that the recall values are high which means that we are correctly identifying the positives. On the other hand, specificity values are low which indicates that we have a difficulty in predicting the negatives.

Table 4.2: Evaluation of DeepInterface model

	Precision	Recall	Specificity	NPV	Accuracy
Training	0.88	0.96	0.88	0.96	0.92
Validation	0.85	0.93	0.83	0.92	0.88
Test	0.61	0.87	0.68	0.90	0.75

We did further evaluation of our model by obtaining the rank of the correct interface (i.e., positive sample) of a PPI complex within the sorted list of all decoys of that complex (i.e., negative samples). We had 599 cases from the negative datasets which also have a correct interface in the positive dataset. Table 4.3 demonstrates the success rates for the top 1%, 10%, and 20% (i.e., top 1% means the correct interface is in the top 1 percentage of the ranked list). We calculated these results for two groups: “training+validation” (454 cases), and “validation+testing” (32 cases including unseen data). For the “training+validation” set, 49% of the cases (223 out of 454) have their positives in top 1%. However, a drop can be seen in the ratio for the set with unseen data (22%); however, it is still a considerably high ratio. To put this evaluation in context, we investigated the results from CIPS [52]; a study which ranks protein interfaces. CIPS scores docking decoys of the interfaces and ranks their corresponding acceptable/valid interfaces. They provided top1% and top20% rank percentages for DOCKGROUND dataset as 63% and 94% respectively. In comparison, our top 1% and top 20% results are 49% and 86% for training, and 22% and 59% for testing. It is important to emphasize that CIPS uses 10% near-native complexes in their decoys set, and if any of these complexes is ranked within the top x%, the prediction is flagged as positive. In our case, we discarded all near-native and acceptable models from the decoys; therefore, we only have one correct structure (i.e., coming from PDB) against 99-100 decoys, which corresponds to only 1% of the decoys. Consequently, our model’s performance can be considered as competitive.

Table 4.3: Ranking of true interface among decoys of the corresponding complex.

	All sets (599 cases)	Training + Validation (454 cases)	Validation + Testing (32 cases)
Top1	0.41	0.49	0.22
Top10	0.76	0.73	0.41
Top20	0.86	0.86	0.59

4.5.1 Data Redundancy Removal

The datasets for training and testing DeepInterface were extracted from different resources; however, there is redundancy between the datasets in terms of similar PPI structures and/or sequences. This redundancy needs to be removed to avoid biased results and evaluation.

To address this problem, we discarded redundant samples from the testing and validation sets. A redundant sample is detected when a sample from either testing or validation sets have higher than 40% sequence similarity with one or more samples from the training set. Redundancy in both negative and positive datasets were handled separately. Also, validation set was uniformly sampled from the same sources as the test dataset. Table 4.4 presents DeepInterface datasets sizes after removing redundant samples. Performance metrics for the training, validation and testing datasets for the model after removing redundancy are shown in Table 4.5.

Table 4.4: DeepInterface datasets after redundancy removal

	PIFACE	PPI4DOCK	DOCKGROUND	PDB2012	Total
Training	127,790	127,790	-	-	255,580
Validation	-	-	2,500	2,500	5,000
Test	-	-	6,764	6,764	13,528
Total	127,790	127,790	9,264	9,264	274,108

Table 4.5: Evaluation of DeepInterface model after removing redundancy

	Precision	Recall	Specificity	NPV	Accuracy
Training	0.945	0.860	0.950	0.871	0.905
Validation	0.744	0.804	0.723	0.786	0.763
Test	0.755	0.806	0.738	0.792	0.772

CONCLUSION

There is a paramount need for identification of HMIs and resolving their complex structures to grasp the pathogenesis mechanisms of infections at the molecular level. In this thesis, Host-Microbe Interaction PREDictor (HMI-PRED) webserver was designed and developed to facilitate this task for researchers with no programming experience. HMI-PRED is a pioneering web server that allows fast, proteome-wide, and robust prediction of structural Host-Microbe interactions through interface mimicry employing both sequence and structural conservation in protein-protein interfaces. It is applicable to any microbial protein, including proteins from pathogens and commensals.

Moreover, using the known HMI complexes in our default template set, we showed that the accuracy of HMI-PRED protocol stages are 94% and 81% for structural alignment and flexible refinement stages respectively. We also suggested a ranking method for the predictions using a deep learning ‘DeepInterface’. We analyzed and evaluated the performance of the model. The accuracy of the model on the test dataset was 77% which is a promising result compared to other available scoring methods.

We also described multiple measures which we applied to speed up predictions and increase their quality. These measures include clustering the template set to reduce structural redundancy and storing both intermediate and final results in a database to avoid recalculation once the same target is input.

As the repository of predicted complexes by the server grows, we anticipate that it will fill the gap of to-date unknown virulence strategies of microbes. With the comprehensive map of structural HMIs, researchers can take further steps to develop efficient therapies against microbial infections.

BIBLIOGRAPHY

- [1] E. Guven-Maiorov, C.-J. Tsai, and R. Nussinov, "Pathogen mimicry of host protein-protein interfaces modulates immunity," *Seminars in Cell & Developmental Biology*, vol. 58, pp. 136-145, 2016/10/01/ 2016.
- [2] E. Guven-Maiorov, C.-J. Tsai, B. Ma, and R. Nussinov, "Prediction of Host-Pathogen Interactions for *Helicobacter pylori* by Interface Mimicry and Implications to Gastric Cancer," *Journal of Molecular Biology*, vol. 429, no. 24, pp. 3925-3941, 2017/12/08/ 2017.
- [3] E. Guven-Maiorov, C.-J. Tsai, B. Ma, and R. Nussinov, "Interface-Based Structural Prediction of Novel Host-Pathogen Interactions," in *Computational Methods in Protein Evolution*, T. Sikosek, Ed. New York, NY: Springer New York, 2019, pp. 317-335.
- [4] E. Guven-Maiorov, C.-J. Tsai, and R. Nussinov, "Structural host-microbiota interaction networks," *PLOS Computational Biology*, vol. 13, no. 10, p. e1005579, 2017.
- [5] C.-J. Tsai, S. L. Lin, H. J. Wolfson, and R. Nussinov, "Protein-Protein Interfaces: Architectures and Interactions in Protein-Protein Interfaces and in Protein Cores. Their Similarities and Differences," *Critical Reviews in Biochemistry and Molecular Biology*, vol. 31, no. 2, pp. 127-152, 1996/01/01 1996.
- [6] O. Keskin and R. Nussinov, "Favorable scaffolds: proteins with different sequence, structure and function may associate in similar ways," *Protein Engineering, Design and Selection*, vol. 18, no. 1, pp. 11-24, 2005.
- [7] O. Keskin and R. Nussinov, "Similar Binding Sites and Different Partners: Implications to Shared Proteins in Cellular Pathways," *Structure*, vol. 15, no. 3, pp. 341-354, 2007.
- [8] S. Muratcioglu, E. Guven-Maiorov, Ö. Keskin, and A. Gursoy, "Advances in template-based protein docking by utilizing interfaces towards completing structural interactome," (in eng), *Current opinion in structural biology*, vol. 35, pp. 87-92, 2015/12// 2015.
- [9] E. A. Franzosa and Y. Xia, "Structural principles within the human-virus protein-protein interaction network," (in eng), *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 26, pp. 10538-10543, 2011.
- [10] C. Nicod, A. Banaei-Esfahani, and B. C. Collins, "Elucidation of host-pathogen protein-protein interactions to uncover mechanisms of host cell rewiring," (in eng), *Current opinion in microbiology*, vol. 39, pp. 7-15, 2017.

- [11] A. F. Brito and J. W. Pinney, "Protein-Protein Interactions in Virus-Host Systems," (in eng), *Frontiers in microbiology*, vol. 8, pp. 1557-1557, 2017.
- [12] D. Mellacheruvu *et al.*, "The CRAPome: a contaminant repository for affinity purification-mass spectrometry data," (in eng), *Nature methods*, vol. 10, no. 8, pp. 730-736, 2013.
- [13] E. Nourani, F. Khunjush, and S. Durmuş, "Computational approaches for prediction of pathogen-host protein-protein interactions," (in eng), *Frontiers in microbiology*, vol. 6, pp. 94-94, 2015.
- [14] R. Mariano and S. Wuchty, "Structure-based prediction of host-pathogen protein interactions," (in eng), *Current opinion in structural biology*, vol. 44, pp. 119-124, 2017/06// 2017.
- [15] M. D. Dyer, T. M. Murali, and B. W. Sobral, "Computational prediction of host-pathogen protein-protein interactions," *Bioinformatics*, vol. 23, no. 13, pp. i159-i166, 2007.
- [16] O. Krishnadev and N. Srinivasan, "Prediction of protein-protein interactions between human host and a pathogen and its application to three pathogenic bacteria," *International Journal of Biological Macromolecules*, vol. 48, no. 4, pp. 613-619, 2011/05/01/ 2011.
- [17] A. C. Doxey and B. J. McConkey, "Prediction of molecular mimicry candidates in human pathogenic bacteria," (in eng), *Virulence*, vol. 4, no. 6, pp. 453-466, 2013.
- [18] T. Huo, W. Liu, Y. Guo, C. Yang, J. Lin, and Z. Rao, "Prediction of host - pathogen protein interactions between Mycobacterium tuberculosis and Homo sapiens using sequence motifs," (in eng), *BMC bioinformatics*, vol. 16, no. 1, pp. 100-100, 2015.
- [19] F.-E. Eid, M. ElHefnawi, and L. S. Heath, "DeNovo: virus-host sequence-based protein-protein interaction prediction," *Bioinformatics*, vol. 32, no. 8, pp. 1144-1150, 2015.
- [20] S.-A. Lee *et al.*, "Ortholog-based protein-protein interaction prediction and its application to inter-species interactions," (in eng), *BMC bioinformatics*, vol. 9 Suppl 12, no. Suppl 12, pp. S11-S11, 2008.
- [21] J. Standfuss, "Viral chemokine mimicry," *Science*, vol. 347, no. 6226, p. 1071, 2015.
- [22] P. Evans, W. Dampier, L. Ungar, and A. Tozeren, "Prediction of HIV-1 virus-host protein interactions using virus and host sequence motifs," (in eng), *BMC medical genomics*, vol. 2, pp. 27-27, 2009.
- [23] A. Becerra, V. A. Bucheli, and P. A. Moreno, "Prediction of virus-host protein-protein interactions mediated by short linear motifs," (in eng), *BMC bioinformatics*, vol. 18, no. 1, pp. 163-163, 2017.
- [24] F. P. Davis, D. T. Barkan, N. Eswar, J. H. McKerrow, and A. Sali, "Host pathogen protein interactions predicted by comparative modeling," (in eng), *Protein science : a publication of the Protein Society*, vol. 16, no. 12, pp. 2585-2596, 2007.
- [25] J. M. Doolittle and S. M. Gomez, "Structural similarity-based predictions of protein interactions between HIV-1 and Homo sapiens," (in eng), *Virology journal*, vol. 7, pp. 82-82, 2010.
- [26] B. de Chassey *et al.*, "Structure homology and interaction redundancy for discovering virus-host protein interactions," (in eng), *EMBO reports*, vol. 14, no. 10, pp. 938-944, 2013.

- [27] N. Drayman *et al.*, "Pathogens Use Structural Mimicry of Native Host Ligands as a Mechanism for Host Receptor Engagement," *Cell Host & Microbe*, vol. 14, no. 1, pp. 63-73, 2013.
- [28] G. Mahajan and S. C. Mande, "Using structural knowledge in the protein data bank to inform the search for potential host-microbe protein interactions in sequence space: application to Mycobacterium tuberculosis," (in eng), *BMC bioinformatics*, vol. 18, no. 1, pp. 201-201, 2017.
- [29] A. Segura-Cabrera, C. A. García-Pérez, X. Guo, and M. A. Rodríguez-Pérez, "A viral-human interactome based on structural motif-domain interactions captures the human infectome," (in eng), *PloS one*, vol. 8, no. 8, pp. e71526-e71526, 2013.
- [30] A. Zhang, L. He, and Y. Wang, "Prediction of GCRV virus-host protein interactome based on structural motif-domain interactions," (in eng), *BMC bioinformatics*, vol. 18, no. 1, pp. 145-145, 2017.
- [31] N. Tuncbag, A. Gursoy, R. Nussinov, and O. Keskin, "Predicting protein-protein interactions on a proteome scale by matching evolutionary and structural similarities at interfaces using PRISM," *Nature Protocols*, vol. 6, p. 1341, 08/11/online 2011.
- [32] A. Baspinar, E. Cukuroglu, R. Nussinov, O. Keskin, and A. Gursoy, "PRISM: a web server and repository for prediction of protein-protein interactions and modeling their 3D complexes," (in eng), *Nucleic acids research*, vol. 42, no. Web Server issue, pp. W285-W289, 2014.
- [33] P. J. Kundrotas, I. Anishchenko, V. D. Badal, M. Das, T. Dauzhenka, and I. A. Vakser, "Modeling CAPRI targets 110-120 by template-based and free docking using contact potential and combined scoring function," (in eng), *Proteins*, vol. 86 Suppl 1, no. Suppl 1, pp. 302-310, 2018.
- [34] H. Hwang, F. Dey, D. Petrey, and B. Honig, "Structure-based prediction of ligand-protein interactions on a genome-wide scale," (in eng), *Proceedings of the National Academy of Sciences of the United States of America*, vol. 114, no. 52, pp. 13685-13690, 2017.
- [35] D. Petrey *et al.*, "Template-based prediction of protein function," *Current Opinion in Structural Biology*, vol. 32, pp. 33-38, 2015/06/01/ 2015.
- [36] R. Viswanathan, E. Fajardo, G. Steinberg, M. Haller, and A. Fiser, "Protein-protein binding supersites," (in eng), *PLoS computational biology*, vol. 15, no. 1, pp. e1006704-e1006704, 2019.
- [37] A. H. Basit, W. A. Abbasi, A. Asif, S. Gull, and F. U. A. A. Minhas, "Training host-pathogen protein-protein interaction predictors," *Journal of Bioinformatics and Computational Biology*, vol. 16, no. 04, p. 1850014, 2018/08/01 2018.
- [38] N. Tuncbag, A. Gursoy, and O. Keskin, "Identification of computational hot spots in protein interfaces: combining solvent accessibility and inter-residue potentials improves the accuracy," *Bioinformatics*, vol. 25, no. 12, pp. 1513-1520, 2009.
- [39] H. Naveed and J. J. J. Q. B. Han, "Structure-based protein-protein interaction networks and drug design," vol. 1, no. 3, pp. 183-191, 2013.
- [40] B. G. Pierce, Y. Hourai, and Z. Weng, "Accelerating protein docking in ZDOCK using an advanced 3D convolution library," (in eng), *PloS one*, vol. 6, no. 9, pp. e24657-e24657, 2011.
- [41] S. Lyskov and J. J. J. N. a. r. Gray, "The RosettaDock server for local protein-protein docking," vol. 36, no. suppl_2, pp. W233-W238, 2008.
- [42] S. J. De Vries, M. Van Dijk, and A. M. J. N. p. Bonvin, "The HADDOCK web server for data-driven biomolecular docking," vol. 5, no. 5, p. 883, 2010.

- [43] O. Keskin, A. Gursoy, B. Ma, and R. Nussinov, "Principles of protein-protein interactions: what are the preferred ways for proteins to interact?," *Chem Rev*, vol. 108, no. 4, pp. 1225-44, Apr 2008.
- [44] M. F. Lensink *et al.*, "Prediction of homoprotein and heteroprotein complexes by protein docking and template-based modeling: A CASP-CAPRI experiment," vol. 84, no. S1, pp. 323-348, 2016.
- [45] S. Jones and J. M. J. J. o. m. b. Thornton, "Analysis of protein-protein interaction sites using surface patches," vol. 272, no. 1, pp. 121-132, 1997.
- [46] O. Keskin, A. Gursoy, B. Ma, and R. J. C. r. Nussinov, "Principles of protein-protein interactions: What are the preferred ways for proteins to interact?," vol. 108, no. 4, pp. 1225-1244, 2008.
- [47] X. Li, O. Keskin, B. Ma, R. Nussinov, and J. J. J. o. m. b. Liang, "Protein-protein interactions: hot spots and structurally conserved residues often locate in complemented pockets that pre-organized in the unbound states: implications for docking," vol. 344, no. 3, pp. 781-795, 2004.
- [48] N. Tuncbag, A. Gursoy, E. Guney, R. Nussinov, and O. Keskin, "Architectures and Functional Coverage of Protein-Protein Interfaces," *Journal of Molecular Biology*, vol. 381, no. 3, pp. 785-802, 2008/09/05/ 2008.
- [49] A. S. Aytuna, A. Gursoy, and O. J. B. Keskin, "Prediction of protein-protein interactions by combining structure and sequence conservation in protein interfaces," vol. 21, no. 12, pp. 2850-2855, 2005.
- [50] U. Ogmen, O. Keskin, A. S. Aytuna, R. Nussinov, and A. Gursoy, "PRISM: protein interactions by structural matching," (in eng), *Nucleic acids research*, vol. 33, no. Web Server issue, pp. W331-W336, 2005.
- [51] S. Liang, S. O. Meroueh, G. Wang, C. Qiu, Y. J. P. S. Zhou, Function,, and Bioinformatics, "Consensus scoring for enriching near-native structures from protein-protein docking decoys," vol. 75, no. 2, pp. 397-403, 2009.
- [52] F. Nadalin and A. J. B. Carbone, "Protein-protein interaction specificity is captured by contact preferences and interface composition," vol. 34, no. 3, pp. 459-468, 2017.
- [53] P. Popov, S. J. J. o. c. i. Grudinin, and modeling, "Knowledge of native protein-protein interfaces is sufficient to construct predictive models for the selection of binding candidates," vol. 55, no. 10, pp. 2242-2255, 2015.
- [54] H. Chen, H. X. J. P. S. Zhou, Function,, and Bioinformatics, "Prediction of interface residues in protein-protein complexes by a consensus neural network method: test against NMR data," vol. 61, no. 1, pp. 21-35, 2005.
- [55] R. A. Jordan, E.-M. Yasser, D. Dobbs, and V. J. B. b. Honavar, "Predicting protein-protein interface residues using local surface structural similarity," vol. 13, no. 1, p. 41, 2012.
- [56] Q. Xu and R. L. J. N. a. r. Dunbrack, "The protein common interface database (ProtCID)—a comprehensive database of interactions of homologous proteins in multiple crystal forms," vol. 39, no. suppl_1, pp. D761-D770, 2010.
- [57] E. Cukuroglu, A. Gursoy, R. Nussinov, and O. Keskin, "Non-redundant unique interface structures as templates for modeling protein interactions," (in eng), *PloS one*, vol. 9, no. 1, pp. e86738-e86738, 2014.
- [58] A. T. Balci, "Fast Screening Algorithms for Protein Interface Structural Alignments," MSc, Computer Science and Engineering, Koç University, 2018.
- [59] H. M. Berman *et al.*, "The Protein Data Bank," *Nucleic Acids Research*, vol. 28, no. 1, pp. 235-242, 2000.

- [60] P. J. Kundrotas *et al.*, "Dockground: A comprehensive data resource for modeling of protein complexes," (in eng), *Protein science : a publication of the Protein Society*, vol. 27, no. 1, pp. 172-181, 2018.
- [61] J. Yu and R. Guerois, "PPI4DOCK: large scale assessment of the use of homology models in free docking over more than 1000 realistic targets," *Bioinformatics*, vol. 32, no. 24, pp. 3760-3767, 2016.
- [62] G. Faure, J. Andreani, and R. J. N. a. r. Guerois, "InterEvol database: exploring the structure and evolution of protein complex interfaces," vol. 40, no. D1, pp. D847-D856, 2011.
- [63] H. Zhu, F. S. Domingues, I. Sommer, and T. J. B. b. Lengauer, "NOXclass: prediction of protein-protein interaction types," vol. 7, no. 1, p. 27, 2006.
- [64] J. J. B. Söding, "Protein homology detection by HMM–HMM comparison," vol. 21, no. 7, pp. 951-960, 2004.
- [65] J. Janin *et al.*, "CAPRI: A Critical Assessment of PRedicted Interactions," vol. 52, no. 1, pp. 2-9, 2003.
- [66] Y. Zhang and J. Skolnick, "TM-align: a protein structure alignment algorithm based on the TM-score," (in eng), *Nucleic acids research*, vol. 33, no. 7, pp. 2302-2309, 2005.
- [67] M. Gao and J. J. B. Skolnick, "iAlign: a method for the structural comparison of protein–protein interfaces," vol. 26, no. 18, pp. 2259-2265, 2010.
- [68] Y. Song *et al.*, "High-Resolution Comparative Modeling with RosettaCM," *Structure*, vol. 21, no. 10, pp. 1735-1742, 2013/10/08/ 2013.
- [69] A. Tovchigrechko and I. A. J. N. a. r. Vakser, "GRAMM-X public web server for protein–protein docking," vol. 34, no. suppl_2, pp. W310-W314, 2006.
- [70] S. J. Hubbard, J. M. J. C. P. Thornton, Department of Biochemistry, and U. C. L. Molecular Biology, "Naccess," vol. 2, no. 1, 1993.
- [71] M. Shatsky, R. Nussinov, and H. J. Wolfson, "A method for simultaneous alignment of multiple protein structures," *Proteins: Structure, Function, and Bioinformatics*, vol. 56, no. 1, pp. 143-156, 2004/07/01 2004.
- [72] M. Girvan, D. S. Callaway, M. E. Newman, and S. H. J. P. R. E. Strogatz, "Simple model of epidemics with pathogen mutation," vol. 65, no. 3, p. 031915, 2002.
- [73] C. Wang, P. Bradley, and D. Baker, "Protein–Protein Docking with Backbone Flexibility," *Journal of Molecular Biology*, vol. 373, no. 2, pp. 503-519, 2007/10/19/ 2007.
- [74] T. U. Consortium, "UniProt: a worldwide hub of protein knowledge," *Nucleic Acids Research*, vol. 47, no. D1, pp. D506-D515, 2018.
- [75] J. M. Duarte, A. Srebniak, M. A. Schärer, and G. Capitani, "Protein interface classification by evolutionary analysis," (in eng), *BMC bioinformatics*, vol. 13, pp. 334-334, 2012.
- [76] M. Uhlen *et al.*, "A Human Protein Atlas for Normal and Cancer Tissues Based on Antibody Proteomics," *Molecular & Cellular Proteomics*, vol. 4, no. 12, p. 1920, 2005.
- [77] M. Uhlen *et al.*, "Proteomics. Tissue-based map of the human proteome," *Science*, vol. 347, no. 6220, p. 1260419, Jan 23 2015.
- [78] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Systematic and integrative analysis of large gene lists using DAVID bioinformatics resources," *Nature Protocols*, vol. 4, p. 44, 12/18/online 2008.

- [79] D. W. Huang, B. T. Sherman, and R. A. Lempicki, "Bioinformatics enrichment tools: paths toward the comprehensive functional analysis of large gene lists," *Nucleic Acids Research*, vol. 37, no. 1, pp. 1-13, 2008.
- [80] E. Cukuroglu, A. Gursoy, and O. Keskin, "HotRegion: a database of predicted hot spot clusters," (in eng), *Nucleic acids research*, vol. 40, no. Database issue, pp. D829-D833, 2012.
- [81] S. Ioffe and C. J. a. p. a. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015.
- [82] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807-814.
- [83] D. Yuret, "Knet: beginning deep learning with 100 lines of julia," in *Machine Learning Systems Workshop at NIPS*, 2016, vol. 2016, p. 5.
- [84] J. Bezanson, A. Edelman, S. Karpinski, and V. B. J. S. r. Shah, "Julia: A fresh approach to numerical computing," vol. 59, no. 1, pp. 65-98, 2017.
- [85] D. P. Kingma and J. J. a. p. a. Ba, "Adam: A method for stochastic optimization," 2014.
- [86] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026-1034.
- [87] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249-256.
- [88] (July 02, 2019). CentOS Documentation Home. Available: <https://docs.centos.org/en-US/docs/>
- [89] R. T. Fielding and G. J. I. I. C. Kaiser, "The Apache HTTP server project," vol. 1, no. 4, pp. 88-90, 1997.
- [90] A. B. Yoo, M. A. Jette, and M. Grondona, "Slurm: Simple linux utility for resource management," in *Workshop on Job Scheduling Strategies for Parallel Processing*, 2003, pp. 44-60: Springer.
- [91] (July 04, 2019). Bootstrap - Introduction. Available: <https://getbootstrap.com/docs/4.0/getting-started/introduction/>
- [92] M. Deshpande. (2018, July 05, 2019). PDB LiteMol. Available: <https://github.com/mandarsd/pdb-lite-mol>
- [93] (July 05, 2019). <https://docs.angularjs.org/api>. Available: <https://docs.angularjs.org/api>
- [94] D. Sehnal *et al.*, "LiteMol suite: interactive web-based visualization of large-scale macromolecular structure data," *Nature Methods*, Correspondence vol. 14, p. 1121, 11/30/online 2017.
- [95] (2019, July 05, 2019). Documentation · TypeScript. Available: <https://www.typescriptlang.org/docs/home.html>
- [96] M. Deshpande. (July 05, 2019). LiteMol Component - selection demo. Available: <https://next.plnkr.co/edit/VyvHgPpRxtMl5modDint?preview>
- [97] B. H. Penn *et al.*, "An Mtb-Human Protein-Protein Interaction Map Identifies a Switch between Host Antiviral and Antibacterial Responses," *Molecular Cell*, vol. 71, no. 4, pp. 637-648.e5, 2018.
- [98] E. Quévrain *et al.*, "Identification of an anti-inflammatory protein from *Faecalibacterium prausnitzii*, a commensal bacterium deficient in Crohn's disease," (in eng), *Gut*, vol. 65, no. 3, pp. 415-425, 2016.

- [99] A. Roy, A. Kucukural, and Y. Zhang, "I-TASSER: a unified platform for automated protein structure and function prediction," (in eng), *Nature protocols*, vol. 5, no. 4, pp. 725-738, 2010.
- [100] N. Zhao, B. Pang, C.-R. Shyu, and D. Korkin, "Feature-based classification of native and non-native protein-protein interactions: Comparing supervised and semi-supervised learning approaches," vol. 11, no. 22, pp. 4321-4330, 2011.
- [101] R. Boppana and M. M. J. B. N. M. Halldórsson, "Approximating maximum independent sets by excluding subgraphs," vol. 32, no. 2, pp. 180-196, 1992.
- [102] J. Håstad, "Clique is Hard to Approximate within," 1997.
- [103] J. W. Moon and L. J. I. j. o. M. Moser, "On cliques in graphs," vol. 3, no. 1, pp. 23-28, 1965.
- [104] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques," in *International Computing and Combinatorics Conference*, 2004, pp. 161-170: Springer.
- [105] C. Bron and J. J. C. A. Kerbosch, "Finding all cliques of an undirected graph (algorithm 457)," vol. 16, no. 9, pp. 575-576, 1973.
- [107] I. M. Bomze, M. Budinich, P. M. Pardalos, and M. J. S. A. Pelillo, "The maximum clique problem. Handbook of Combinatorial Optimization," vol. 174, 1999.

APPENDIX A

ENUMERATING MAXIMAL CLIQUES (ALL-CLIQUE) ALGORITHMS

Three main types of clique related problems appear in literature:

- i. *CLIQUE*, the decision problem of whether a graph has a clique of size at least k .
- ii. *Max-Clique (MC)*, is the optimization version which searches for the maximum clique; i.e. the largest maximal clique in graph.
- iii. *All-Clique or Maximal Clique Enumeration (MCE)*, is the problem of enumerating all maximal cliques in a graph.

The CLIQUE problem is one of the first known problems in NP-complete (Karp, 1972). While the optimization version Max-Clique is in NP-Hard. The best-known approximation algorithm for Max-Clique is $O\left(\frac{n}{(\log_2 n)^2}\right)$ [101]. It was proved that, unless $NP = ZPP^*$, for any $\varepsilon > 0$, Max-Clique cannot be efficiently approximated with a factor of $n^{1-\varepsilon}$ [102].

In both heuristics we developed, we utilized the third and most complex problem of the three; *All-Clique*, which is at least as hard as *Max-Clique*. All-Clique problem is NP-Hard because the maximum number of cliques in a graph G with n nodes is exponential in n . Moon and Moser [103] provided an upper bound $f(n)$ (Equation A.1) on the maximum number of cliques which can be found in a graph with n nodes.

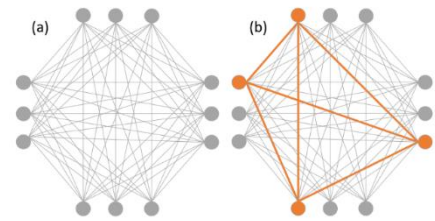


Figure A.1: A graph which achieves Moon & Moser's bound

* ZPP is the class of problems which have a Zero-errors Probabilistic Polynomial time solution. The belief of $ZPP \neq NP$ is almost as strong as $NP \neq P$

$$f(n) = \begin{cases} 3^{\frac{n}{3}}, & \text{if } n \equiv 0(\text{mod}3) \\ 4 \cdot 3^{\frac{n}{3}-1}, & \text{if } n \equiv 1(\text{mod}3), \forall n \geq 2 \\ 2 \cdot 3^{\frac{n}{3}}, & \text{if } n \equiv 2(\text{mod}3) \end{cases} \quad (\text{A.1})$$

Figure A.1 shows an example of graph G with $n = 12 \equiv 0(\text{mod}3)$ nodes that achieves Moon Moser's bound. In Figure A.1 one of the possible cliques is highlighted in orange. There are $3 \cdot 3 \cdot 3 \cdot 3 = 3^4 \equiv 3^{n/3}$ maximal cliques in G .

The algorithm used in both heuristic we suggested is based on Tomita et al.[104]'s variation of the Bron-Kerbosch algorithm BK [105] which is a recursive backtracking algorithm and it is considered as the most efficient exact algorithm known for generating all the cliques of a graph [106, 107].

Bron-Kerbosch Algorithm V1 (original)

The first version of BK algorithm is mainly presented to illustrate the method used and it is important to understand since the other version uses it with a slight change that will improve the efficiency and reduce the running time of the algorithm.

```

1. bronKerbosch_v1(R, P, X, cliques):
2.   if (P is empty & X is empty)
3.     // Report R as a maximal clique
4.     cliques.add(R)
5.
6.   for each v in P
7.     // N(v): neighbors of node v
8.     cliques = bronKerbosch_v1(R ∪ {v}, P ∩ N(v), X ∩ N(v), cliques)
9.     P = P \ {v}
10.    X = X ∪ {v};
11.
12.   return cliques

```

Figure A.2: Bron-Kerbosch algorithm (version 1) Pseudocode

Figure A.2 contains the pseudocode for BK version1 algorithm. The algorithm takes the three sets, R , P and X , as inputs. R is the set of vertices belonging to the current clique, P is the set of candidate vertices which can be added to R (i.e. $\forall v \in P: \exists (v,u) \in E, \forall u \in R$), while X is the excluded set of vertices which are not allowed to be added to C . X is used in order to avoid finding an already found clique. The initial call of the algorithm should be $\text{bronKerbosch_v1}(\{\}, V, \{\}, \{\})$.

Figure A.3 shows a part of the execution of `bronKerbosch_v1` on a sample graph. The algorithm starts with both R & X as empty, while P has all vertices. First, vertex $\{1\}$ is picked/ from P and added to R , then the algorithm is called recursively with $R = \{1\}$, $P = P \cap N(1) = \{1,2,3,4,5\} \cap \{2,3,4,5\} = \{2,3,4,5\}$, and $X = X \cap N(1) = \{\}$. The steps follow recursively until both X and P are empty, which means that R forms a maximal clique. The case where P is empty while X is not as in Figure A.3c means that the clique in R is a part of an already found clique thus the execution is stopped for the current branch.



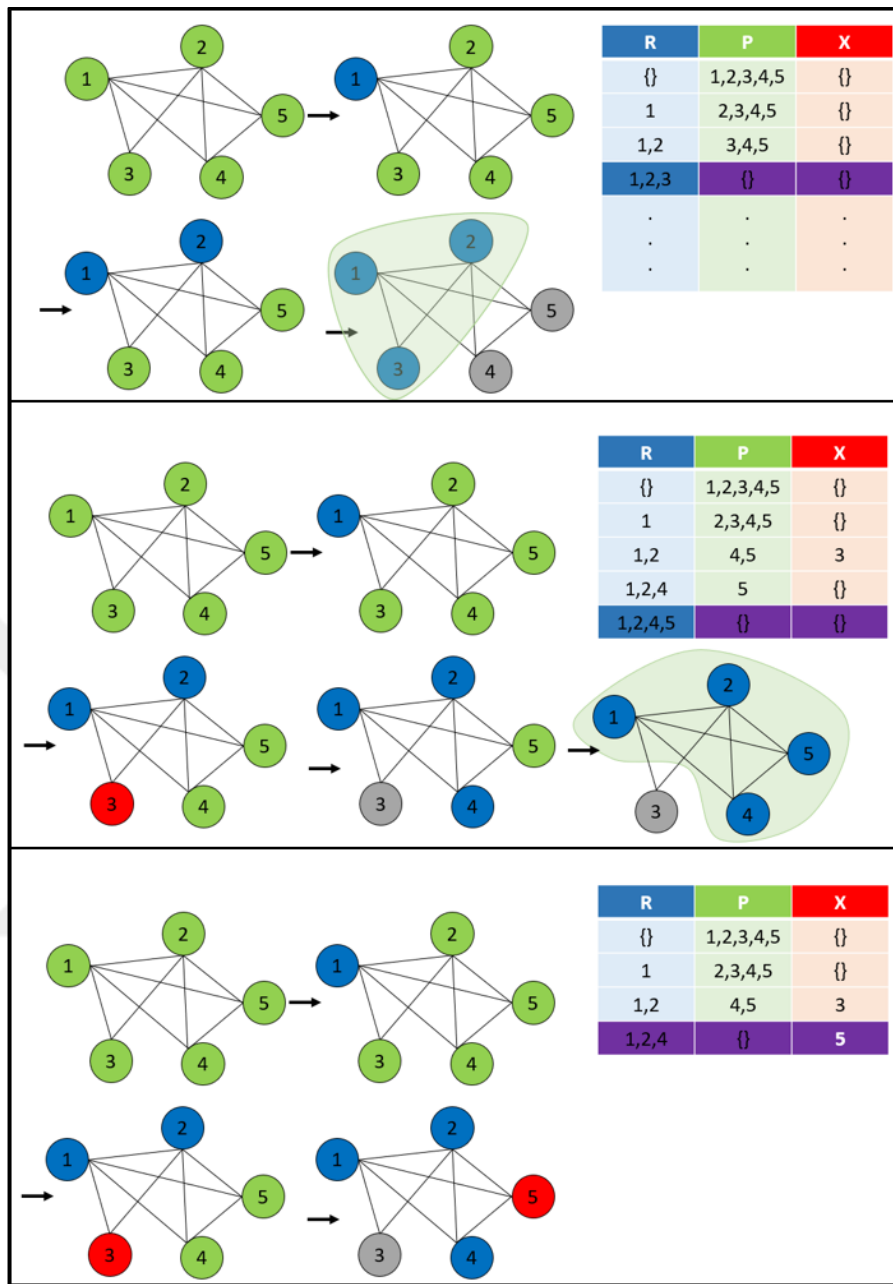


Figure A.3: An example of the execution of Bron-Kerbosch algorithm (v1). (a) shows the execution of the first recursive branch (i.e. until it finds the first clique). The table shows the sets R , P , X at each step of the execution. (b) shows the detection of the 2nd clique (2nd branch). (c) shows a bounded branch where P is empty, but X is not, which means this branch will generate a clique which was already generated thus the branch is bounded to avoid redundancy.

Bron-Kerbosch Algorithm V2 (with pivot)

The second version of Bron-Kerbosch algorithm is based on the first version. This improvement was suggested by Tomita et al. [104] based on an unexploited note mentioned by Bron and Kerbosch in their original paper [105].

Figure A.4 contains the pseudo code for Bron-Kerbosch version 2. The algorithm uses a pivoting criterion to cut computational branches thus reducing the running time. In each step, instead of iterating over all vertices in P , a pivot is picked (the vertex with the highest number of neighbors included in $P \cup X$). Then, R can be expanded by either the pivot or one of its non-neighbors in P .

```

1. bronKerbosch_v2(R, P, X, cliques):
2.   if (P is empty & X is empty)
3.     // Report R as a maximal clique
4.     cliques.add(R)
5.
6.   pivot <- the node with maximum number of neighbors in P ∪ X
7.   for each v in P - N(pivot)
8.     cliques = bronKerbosch_v2(R ∪ {v}, P ∩ N(v), X ∩ N(v), cliques)
9.     P = P \ {v}
10.    X = X ∪ {v};
11.
12.   return cliques

```

Figure A.4: Bron-Kerbosch algorithm (version 2) Pseudocode.
Differences from version 1 are highlighted in yellow.

Figure A.5 shows the reduction in the execution tree which results from applying the pivoting policy suggested by Tomita et al. Figure A.5b shows the execution tree of running `bronKerbosch_v1` on G (the graph shown in Figure A.5a). The branches ending with blue nodes are the ones which reported a clique, while the branches ending with a red node are the ones that were bounded/cut to avoid redundancy. The execution tree of `bronKerbosch_v2` on G is shown in Figure A.5c. The numbers inside the tree nodes represents the pivot which was picked at each step. For example, at the first step, 1 was picked as the pivot since it has the highest number of neighbors in P ; ($|N(1) \cap P| = 4$). Thus, the algorithm iterates over $P \cap N(1) = \{1\}$ only.

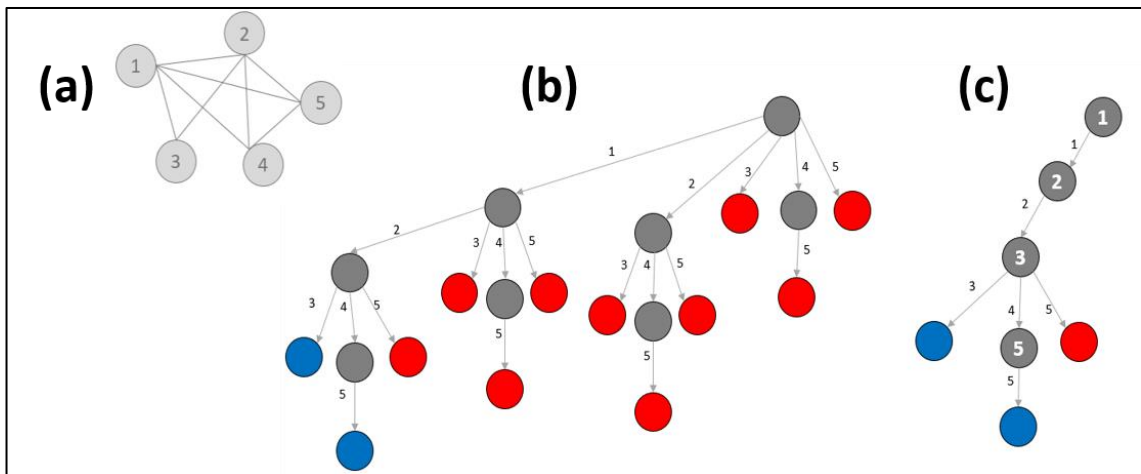


Figure A.5: Execution trees for Bron-Kerbosch v1(b) and v2 (c) on the graph in (a)

APPENDIX B

DETAILED ACCURACY TEST RESULTS FOR KNOWN HMIS IN
THE DEFAULT TEMPLATE SET

Microbial protein PDB	Template PDB	Protein1 (Microbial)	Protein2 (Host)	EPPIC score	I_sc of HMI
1k90A	1k90AD	CYAA_BACAN	CALM2_HUMAN	0.9999	-42.155
1xtgA	1xtgAB	BXA1_CLOBO	SNP25_HUMAN	0.99987	-40.942
4n9fq	4n9fkq	PEBB_HUMAN	VIF_HV1N5	0.97887	-34.872
4didB	4didAB	CDC42_HUMAN	SOPB_SALTY	0.99783	-34.523
2wwwB	2wwwAB	RAB1A_HUMAN	DRRA_LEGPH	0.99582	-31.42
4cc9B	4cc9AB	DCAF1_HUMAN	VPX_SIVSP	0.97052	-31.415
4gmxB	4gmxAB	RAN_HUMAN	YRB1_YEAST	0.97333	-31.246
3fieA	3fieAC	BXF_CLOBL	VAMP2_HUMAN	0.66298	-30.841
5k35A	5k35AB	Q5ZTL7_LEGPH	SKP1_HUMAN	0.99917	-29.877
3bg0B	3bg0AB	SEC13_HUMAN	NU145_YEAST	0.99994	-29.259
3jzaB	3jzaAB	RAB1B_HUMAN	DRRA_LEGPH	0.99793	-28.983
6bw9B	6bw9AB	IMA3_HUMAN	W_HENDH	0.99957	-28.858
5o74A	5o74AB	DRRA_LEGPN	RAB1B_HUMAN	0.99854	-28.023
6gn8C	6gn8AC	1433B_HUMAN	Q93SQ1_PSEAI	0.99976	-27.483
2b5lC	2b5lAC	DDB1_HUMAN	V_PIV5	0.56308	-27.471
1yrtA	1yrtAB	CYAA_BORPE	CALM2_HUMAN	0.98965	-27.262
3tj5B	3tj5AB	VINC_HUMAN	BOBXR4_RICRO	0.93381	-27.201
3mi9C	3mi9BC	CCNT1_HUMAN	TAT_HV1H2	0.81971	-26.937
3d2uA	3d2uAB	UL18_HCMVA	B2MG_HUMAN	0.9982	-26.764
3bg0C	3bg0CD	NU145_YEAST	SEC13_HUMAN	0.99986	-26.741
5l26B	5l26AB	GCP60_HUMAN	Q8BES6_9PICO	0.7663	-26.151
3l0iA	3l0iAB	DRRA_LEGPH	RAB1A_HUMAN	0.96424	-25.711
6gn0E	6gn0AE	1433B_HUMAN	Q51451_PSEAI	0.99976	-25.698
4f8cA	4f8cAB	A0A0H3B1Q8_YERPY	NEDD8_HUMAN	0.9955	-25.608
5oecA	5oecAB	Q6EAT3_SALCE	RAB32_HUMAN	0.9669	-24.579
4k1rA	4k1rAB	SST2_SCHPO	UBC_HUMAN	0.99396	-24.381
5l1zD	5l1zBD	CCNT1_HUMAN	TAT_HV1B1	0.81162	-24.168
4f0zC	4f0zAC	PP2BA_HUMAN	IKBL_ASFM2	0.23941	-23.469
4z8lB	4z8lAB	DCAF1_HUMAN	Q7ZB17_SIV	0.71008	-23.418
4fbjA	4fbjAB	Q7N439_PHOLL	NEDD8_HUMAN	0.98874	-23.199
1yagA	1yagAG	ACT_YEAST	GELS_HUMAN	0.49472	-23.175
3gcgB	3gcgAB	CDC42_HUMAN	Q9R8E4_ECOLX	0.99224	-23.003
3mhsA	3mhsAD	UBP8_YEAST	UBC_HUMAN	0.99887	-22.963
2gwwB	2gwwAB	VINC_HUMAN	IPAA_SHIFL	0.88245	-22.756
2hsqB	2hsqAB	VINC_HUMAN	Q6XVZ2_SHIFL	0.97206	-22.697
4gmxC	4gmxAC	RAN_HUMAN	XPO1_YEAST	0.87899	-22.61

4hcpA	4hcpAB	Q63KH5_BURPS	NEDD8_HUMAN	0.97885	-22.51
4zfrA	4zfrAB	SST2_SCHPO	UBB_HUMAN	0.98828	-22.17
1gzsB	1gzsAB	CDC42_HUMAN	SOPE_SALTM	0.98972	-22.073
4fmbA	4fmbAB	VIRA_SHIFL	RAB1A_HUMAN	0.95617	-21.988
5uzuB	5uzuAB	PERM_HUMAN	Q2G0X2_STAA8	0.99185	-21.892
1re0B	1re0AB	ARF1_HUMAN	GEA1_YEAST	0.88205	-21.787
1bmlC	1bmlAC	PLMN_HUMAN	STRP_STREQ	0.34225	-21.713
4xt1A	4xt1AB	US28_HCMVA	X3CL1_HUMAN	0.94746	-21.604
4uaeF	4uaeAF	IMA3_HUMAN	PB2_I75A3	0.9999	-21.481
4iruA	4iruAB	Q5ZSM7_LEGPH	RAB1A_HUMAN	0.97767	-21.238
4uadE	4uadAE	IMA7_HUMAN	PB2_I75A3	0.98879	-21.116
4zuxU	4zuxUX	UBP8_YEAST	UBB_HUMAN	0.99558	-21.078
6bmtB	6bmtAB	PERM_HUMAN	A0A2A4GXB5_9STAP	0.18165	-20.634
1cmxA	1cmxAB	UBL1_YEAST	UBC_HUMAN	0.94483	-20.449
4irvA	4irvAE	CAGA_HELPY	ASPP2_HUMAN	0.19467	-20.443
5u1dX	5u1dBX	TAP2_HUMAN	A0A140GKJ0_HHV1	0.84696	-20.372
6gnnC	6gnnAC	1433B_HUMAN	Q9I788_PSEAE	0.99971	-20.332
5ajjA	5ajjAB	ARBH_VAR67	BID_HUMAN	0.99528	-20.28
4uf1A	4uf1AB	DPV22_DPV83	BAK_HUMAN	0.99816	-20.201
1jowA	1jowAB	CGH2_SHV21	CDK6_HUMAN	0.9964	-20.04
5cawA	5cawAB	E0VIU9_PEDHC	UBB_HUMAN	0.92701	-19.978
4i1oD	4i1oAD	RAB1B_HUMAN	Q5ZSM7_LEGPH	0.95013	-19.974
6b7mA	6b7mAB	Q5ZSK8_LEGPH	UBC_HUMAN	0.69094	-19.948
5jk7H	5jk7CH	DCAF1_HUMAN	VPR_HV1N5	0.61134	-19.848
4b60A	4b60AC	FNBA_STAA8	FIBG_HUMAN	0.96786	-19.828
2wdtA	2wdtAD	Q8IKM8_PLAF7	UBC_HUMAN	0.89	-19.802
4i1oB	4i1oBC	Q5ZSM7_LEGPH	RAB1B_HUMAN	0.95446	-19.747
3v89A	3v89AB	TBP1_NEIMB	TRFE_HUMAN	0.11282	-19.731
4jvsA	4jvsAB	G9EPL4_9GAMM	RAB1A_HUMAN	0.96665	-19.629
1p0sE	1p0sEH	ECOT_ECOLI	FA10_HUMAN	0.99779	-19.403
6fdkA	6fdkAB	CDUB1_CHLT2	UBB_HUMAN	0.71257	-19.382
4fmcA	4fmcAB	Q7DB50_ECO57	RAB1A_HUMAN	0.97719	-19.244
2a5dB	2a5dAB	ARF6_HUMAN	CHTA_VIBCH	0.97421	-19.218
5mzaA	5mzaAB	Q8IHM0_PLAF7	ICAM1_HUMAN	0.65122	-19.136
4xs0C	4xs0AC	HBA_HUMAN	ISDH_STAA3	0.19753	-19.079
5u9mB	5u9mAB	SODC_HUMAN	CCS1_YEAST	0.97814	-18.905
5uwsC	5uwsCD	XPO1_YEAST	APBA3_HUMAN	0.9793	-18.899
2wdtC	2wdtBC	UBC_HUMAN	Q8IKM8_PLAF7	0.92	-18.73
4uf7A	4uf7AE	I0E093_9MONO	EFNB2_HUMAN	0.86568	-18.699
6avzB	6avzAB	CEAM3_HUMAN	H6A3H4_HELPX	0.08028	-18.583
4ayiD	4ayiAD	CFAH_HUMAN	Q19KF7_NEIME	0.08148	-18.533
5vmmF	5vmmAF	HBA_HUMAN	ISDB_STAAN	9.58E-02	-18.279
6gbgA	6gbgAD	B5Z8H1_HELPG	CEAM1_HUMAN	0.10425	-18.004
6aw2B	6aw2AB	CEAM1_HUMAN	H6A3H4_HELPX	0.55	-17.998

2jdqE	2jdqAE	IMA5_HUMAN	PB2_I75A3	0.99111	-17.99
4zk9A	4zk9AB	Q2F862_ORFV	CCL2_HUMAN	0.85609	-17.895
5uwjC	5uwjCD	XPO1_YEAST	FMR1_HUMAN	0.9871	-17.85
4d2mA	4d2mAB	F1_VACCA	B2L11_HUMAN	0.97889	-17.714
2wy3B	2wy3AB	MICB_HUMAN	UL16P_HCMVA	0.14345	-17.712
2rkzM	2rkzAM	FINC_HUMAN	FNBA_STAA8	0.47752	-17.617
2kjeB	2kjeAB	CBP_HUMAN	E1A_ADE05	0.81153	-17.609
2wh6A	2wh6AB	EAR_EBVA8	B2L11_HUMAN	0.99457	-17.601
5b1zC	5b1zAC	B2CL1_HUMAN	I3XMW8_HBV	0.97137	-17.239
3f62A	3f62AB	Q85319_9POXV	IL18_HUMAN	0.83531	-17.168
4f1zA	4f1zAQ	CLFB_STAAR	K1C10_HUMAN	0.77088	-17.15
5jp3A	5jp3AB	Q3BYJ5_XANC5	UBB_HUMAN	0.77389	-17.134
6e8rC	6e8rAC	SNX32_HUMAN	B7SCI5_CHLTH	0.61105	-17.125
1ughI	1ughEI	UNG_HUMAN	UNGI_BPPB2	0.9859	-17.093
3phxA	3phxAB	L_CCHFI	ISG15_HUMAN	0.96155	-17.034
2w80C	2w80BC	CFAH_HUMAN	Q9JXV4_NEIMB	0.08834	-16.947
5wtbA	5wtbAE	SDRE_STAAM	CFAH_HUMAN	0.11346	-16.833
3dvuA	3dvuAC	ARBH_MHV68	BECN1_HUMAN	0.65813	-16.706
5uwiC	5uwiCD	XPO1_YEAST	HDAC5_HUMAN	0.93454	-16.692
4n9fG	4n9fEG	ELOC_HUMAN	VIF_HV1N5	0.75678	-16.672
6cp2A	6cp2AC	Q6RCR4_LEGPN	UBB_HUMAN	0.71743	-16.66
5uwhC	5uwhCD	XPO1_YEAST	PAXI_HUMAN	0.75162	-16.649
3pt2A	3pt2AB	Q6TQF5_9VIRU	UBC_HUMAN	0.98399	-16.566
5kq1A	5kq1AC	DCP1_SCHPO	PNRC2_HUMAN	0.67137	-16.551
2jbyA	2jbyAB	Q85295_9POXV	BAK_HUMAN	0.97099	-16.512
4n9f1	4n9f1z	VIF_HV1N5	ELOC_HUMAN	0.69032	-16.509
3pcsA	3pcsAE	Q7DB50_ECO57	PAK2_HUMAN	0.33012	-16.464
4f27A	4f27AQ	CLFB_STAAR	FIBA_HUMAN	0.80185	-16.436
5ajkA	5ajkAB	ARBH_VAR67	BAK_HUMAN	0.9941	-16.389
3znhA	3znhAB	L_CCHFI	UBB_HUMAN	0.97647	-16.345
5wfiA	5wfiAC	R1AB_CVMA5	UBB_HUMAN	0.46067	-16.32
4xsgB	4xsgAB	RHOA_HUMAN	Q8KNY0_BACCE	0.49895	-16.247
1xdtT	1xdtRT	HBEGF_HUMAN	DTX_CORBE	0.5335	-16.217
3ea5B	3ea5AB	RAN_HUMAN	IMB1_YEAST	0.93901	-16.174
3by4A	3by4AB	OTU1_YEAST	UBC_HUMAN	0.17323	-16.126
4d2IA	4d2IAB	F1_VACCA	BAK_HUMAN	0.98297	-16.107
6atgC	6atgCD	O50665_BORBU	A0A024R962_HUMAN	0.42893	-16.107
4cc9B	4cc9BC	VPX_SIVSP	SAMH1_HUMAN	0.29786	-16.102
3aswA	3aswAB	CLFB_STAAN	K1C10_HUMAN	0.00938	-16.092
1he1A	1he1AC	Q51451_PSEAI	RAC1_HUMAN	0.80791	-16.09
3pseA	3pseAB	Q6TQF5_9VIRU	ISG15_HUMAN	0.82371	-16.065
5w8tA	5w8tAB	M4STU1_9BETC	ISG15_HUMAN	0.70833	-16.028
5f1bA	5f1bAC	VGP_EBOZ5	NPC1_HUMAN	0.26698	-15.992
3jrvA	3jrvAC	K7_VACCW	DDX3X_HUMAN	0.16962	-15.948
6gbhA	6gbhAB	Q8GDI6_HELPX	CEAM1_HUMAN	0.0335	-15.879

5cfaA	5cfaAC	BBP_STAAU	FIBA_HUMAN	0.97879	-15.868
3phwA	3phwAB	L_CCHFI	RS27A_HUMAN	0.98974	-15.838
3wygC	3wygCD	XPO1_YEAST	IPKA_HUMAN	0.99135	-15.828
5difC	5difCD	XPO1_YEAST	CPEB4_HUMAN	0.9747	-15.817
6atgB	6atgAB	AOA024R962_HUMAN	O50665_BORBU	0.40837	-15.72
3prmA	3prmAB	Q6TFZ7_9VIRU	UBC_HUMAN	0.98293	-15.717
4itrA	4itrAD	IBPA_HISS2	CDC42_HUMAN	0.19647	-15.705
2ny6A	2ny6AB	Q8Q7Z9_9HIV1	CD4_HUMAN	0.3296	-15.651
5zq3A	5zq3AD	Q6BBR6_LEGPN	UBC_HUMAN	0.46775	-15.629
5hplA	5hplAC	RSP5_YEAST	RL40_HUMAN	0.82263	-15.503
5wosA	5wosAB	ARBH_CNPV	B2L11_HUMAN	0.98134	-15.421
5zq5A	5zq5AB	Q6BBR6_LEGPN	UBC_HUMAN	0.09582	-15.421
4uf3A	4uf3AB	DPV22_DPV83	B2L11_HUMAN	0.99412	-15.34
4u0qA	4u0qAB	B2L3N7_PLAFA	BASI_HUMAN	0.10368	-15.317
4nzlB	4nzlAB	ELNE_HUMAN	AOA0H3KOM1_STAAM	0.98403	-15.143
2ny1A	2ny1AB	Q993A8_9HIV1	CD4_HUMAN	0.45411	-14.909
2a78B	2a78AB	RALA_HUMAN	ARC3_CBDP	0.99775	-14.877
2nxzA	2nxzAB	ENV_HV1MF	CD4_HUMAN	0.3849	-14.808
4hxdB	4hxdAB	UBC_HUMAN	L_DUGBA	0.9886	-14.687
2lnhC	2lnhAC	WASL_HUMAN	ESFU3_ECO57	0.34007	-14.672
3reaA	3reaAB	NEF_HV1A2	HCK_HUMAN	0.48295	-14.655
4eeeA	4eeeAB	Q9DHU8_YLDV	IL18_HUMAN	0.73653	-14.636
2nxyA	2nxyAB	Q49DS8_9HIV1	CD4_HUMAN	0.47752	-14.565
5lgbB	5lgbAB	CD36_HUMAN	Q25733_PLAFA	0.5575	-14.539
4c56C	4c56BC	TRBC2_HUMAN	ETXB_STAAU	0.3296	-14.537
3tnfB	3tnfAB	RAB8A_HUMAN	Q5ZWZ3_LEGPH	0.9574	-14.484
2vskA	2vskAB	GLYCP_HENDH	EFNB2_HUMAN	0.97552	-14.472
2ny5G	2ny5CG	CD4_HUMAN	ENV_HV1MF	0.3109	-14.377
4uf2A	4uf2AB	DPV22_DPV83	BAX_HUMAN	0.88117	-14.227
4f20A	4f20AQ	CLFB_STAAR	DMKN_HUMAN	0.73491	-14.161
2x19A	2x19AB	GSP1_YEAST	IPO13_HUMAN	0.99538	-14.15
6d03E	6d03BE	TFR1_HUMAN	A5K736_PLAVS	0.00108	-14.091
3lw8F	3lw8AF	RHOA_HUMAN	Q9AJW7_SHIFL	0.29896	-14.048
4r2gO	4r2gBO	CD4_HUMAN	ENV_HV1JR	0.14224	-14.013
2uzxA	2uzxAB	INLB_LISMO	MET_HUMAN	0.03498	-13.914
4mm3B	4mm3AB	UBC_HUMAN	R1A_CVHSA	0.16667	-13.896
5dh9C	5dh9CD	XPO1_YEAST	IPKA_HUMAN	0.99711	-13.813
5fw5C	5fw5AC	G3BP1_HUMAN	POLN_SFV	0.51675	-13.719
2ma9A	2ma9AC	VIF_HV1N5	ELOC_HUMAN	0.25106	-13.686
3d2uA	3d2uAC	UL18_HCMVA	ACTB_HUMAN	0.69847	-13.676
2b4cG	2b4cCG	CD4_HUMAN	Q75760_9HIV1	0.57746	-13.63
1nu7D	1nu7BD	THRB_HUMAN	Q846V4_STAAU	0.68714	-13.565
4wurA	4wurAB	R1AB_CVEMC	UBB_HUMAN	0.27178	-13.504
5a31R	5a31CR	CDC23_HUMAN	CDH1_YEAST	0.98057	-13.461
4u2xA	4u2xAD	VP24_EBOZM	IMA6_HUMAN	0.6139	-13.397

5craA	5craAD	Q6RCR0_LEGPN	UBB_HUMAN	0.56848	-13.389
4isqA	4isqAF	Q9LBR1_CLOBO	SYT1_HUMAN	0.70891	-13.377
4r2gA	4r2gAL	ENV_HV1JR	CD4_HUMAN	0.21418	-13.355
2v6qA	2v6qAB	EAR_EBVG	B2L11_HUMAN	0.99414	-13.328
4iw4C	4iw4CE	ECOT_ECOLI	MASP1_HUMAN	0.91549	-13.193
3at0A	3at0AB	CLFB_STAAN	FIBA_HUMAN	0.01707	-13.133
4zncA	4zncAE	SPA_STAAU	IGHG3_HUMAN	0.82987	-13.086
1g4uS	1g4uRS	RAC1_HUMAN	SPTP_SALTY	0.55915	-13.08
5jk7F	5jk7DF	UNG_HUMAN	VPR_HV1N5	0.86304	-13.072
6ay9A	6ay9AB	B6DRA0_9HIV1	CPSF6_HUMAN	0.56547	-12.965
1g3nC	1g3nAC	CDK6_HUMAN	Q98147_HHV8	0.90162	-12.797
4mywA	4mywAB	GD_HHV23	NECT1_HUMAN	0.84466	-12.743
4pufA	4pufAC	SLRP_SALT1	THIO_HUMAN	0.65766	-12.74
4u0aA	4u0aAB	GAG_HV1N5	CPSF6_HUMAN	0.62948	-12.714
4iumA	4iumAB	RPOA_EAVBU	UBC_HUMAN	0.48	-12.679
3kasB	3kasAB	TFR1_HUMAN	Q8AZ57_MACHU	0.03745	-12.674
3w31A	3w31AB	Q8VSD5_SHIFL	UBE2N_HUMAN	0.97394	-12.67
1xx9C	1xx9AC	FA11_HUMAN	ECOT_ECOLI	0.99446	-12.628
5ayrB	5ayrAB	UNG_HUMAN	Q936H5_STAAU	0.9951	-12.595
4orzB	4orzAB	HCK_HUMAN	NEF_HV1A2	0.50547	-12.59
2rl0G	2rl0AG	FINC_HUMAN	FNBA_STAA8	0.62948	-12.518
3v3kB	3v3kAB	CASP9_HUMAN	NLEF_ECO57	0.92079	-12.513
5etaC	5etaAC	MK14_HUMAN	B6KJB6_TOXGV	0.94795	-12.473
4xr8F	4xr8CF	P53_HUMAN	VE6_HP16	0.021	-12.457
1l4dB	1l4dAB	PLMN_HUMAN	STRP_STREQ	0.51864	-12.437
4kr0B	4kr0AB	DPP4_HUMAN	K0BRG7_9BETC	0.0023	-12.408
4bvua	4bvuaAC	OSPG_SHIFL	UBC_HUMAN	0.45535	-12.357
5uwuC	5uwuCD	XPO1_YEAST	SMAD4_HUMAN	0.94833	-12.348
2goxB	2goxAB	CO3_HUMAN	FIB_STAAM	0.62126	-12.328
1r5iD	1r5iAD	DRA_HUMAN	Q48898_MYCAT	0.72593	-12.317
2h7vC	2h7vAC	RAC1_HUMAN	YPKA_YERPS	0.62425	-12.231
5wtbC	5wtbCG	SDRE_STAAM	CFAH_HUMAN	0.39376	-12.182
4wymA	4wymAM	POL_HV1N5	CPSF6_HUMAN	0.55478	-12.031
6eduD	6eduDI	B3UES2_9HIV1	CD4_HUMAN	0.07486	-12.001
5vn3G	5vn3CG	CD4_HUMAN	B3UES2_9HIV1	0.37065	-11.993
2nd0B	2nd0AB	BRD4_HUMAN	LANA1_HHV8P	0.38533	-11.974
3rbbA	3rbbAB	NEF_HV1A2	HCK_HUMAN	0.60637	-11.962
4h8wG	4h8wCG	CD4_HUMAN	Q0ED31_9HIV1	0.2274	-11.916
6b7mA	6b7mAC	Q5ZSK8_LEGPH	UBC_HUMAN	6.14E-02	-11.883
2nnuA	2nnuAB	VE2_HP16	BRD4_HUMAN	0.37608	-11.854
3ik5A	3ik5AB	NEF_SIVMA	CD3Z_HUMAN	0.7413	-11.844
5cayG	5cayBG	CD4_HUMAN	ENV_HV2ST	0.0803	-11.84
6c0bA	6c0bAB	TOXB_CLODI	FZD2_HUMAN	0.21	-11.802
5jw7A	5jw7AB	SOPA_SALTY	TRI56_HUMAN	0.76084	-11.799

5nuhA	5nuhAD	Q5QGG3_SIV	HCK_HUMAN	0.11511	-11.798
4odbA	4odbAD	SIGM1_REOVL	JAM1_HUMAN	0.58807	-11.789
4n9f2	4n9f2N	VIF_HV1N5	PEBB_HUMAN	0.98832	-11.715
2ij0A	2ij0AE	TSST_STAAU	AOA5B4_HUMAN	0.04285	-11.66
3wwtB	3wwtAB	STAT1_HUMAN	C_SENDZ	0.4169	-11.631
6ffaA	6ffaAB	POLG_FMDVO	ISG15_HUMAN	0.40131	-11.572
1l4zB	1l4zAB	PLMN_HUMAN	STRP_STREQ	0.31546	-11.409
3ve1A	3ve1AB	TBB1_NEIMI	TRFE_HUMAN	0.10997	-11.361
1g9mG	1g9mCG	CD4_HUMAN	ENV_HV1H2	0.37828	-11.349
4r4hA	4r4hAB	ENV_HV1B9	CD4_HUMAN	0.46251	-11.271
2n3uA	2n3uAB	RPN1_YEAST	RL40_HUMAN	0.17951	-11.266
1m9cC	1m9cBC	PPIA_HUMAN	Q72497_9HIV1	0.53402	-11.263
5thrD	5thrDG	Q2N0S6_9HIV1	CD4_HUMAN	0.5255	-11.263
2j1kC	2j1kCT	SPIKE_ADECT	CXAR_HUMAN	0.72591	-11.255
3mzwB	3mzwAB	ERBB2_HUMAN	SPA_STAAU	0.26863	-11.254
4l72B	4l72AB	DPP4_HUMAN	R9UQ53_9BETC	0.00329	-11.238
5dhfC	5dhfCD	XPO1_YEAST	RIOK2_HUMAN	0.94452	-11.235
1ml0A	1ml0AD	O41925_MHV68	CCL2_HUMAN	0.96751	-11.235
1hezE	1hezCE	IGKC_HUMAN	Q51918_FINMA	0.55478	-11.208
4qzvB	4qzvAB	DPP4_HUMAN	SPIKE_BCHK4	0.00473	-11.149
4kyiA	4kyiAB	Q5ZRP9_LEGPH	RAB5C_HUMAN	0.59844	-11.143
1sjeC	1sjeBC	2B11_HUMAN	GAG_HV1Z2	0.00111	-11.139
6cm3D	6cm3DG	Q2N0S5_9HIV1	CD4_HUMAN	0.75927	-11.102
1ak4C	1ak4BC	PPIA_HUMAN	POL_HV1N5	0.82895	-11.051
1d5mC	1d5mAC	DRA_HUMAN	ETXB_STAAU	0.92233	-11.003
1fc2C	1fc2CD	SPA_STAAU	IGHG1_HUMAN	0.45366	-10.972
2h59A	2h59AD	NPD_THEMA	P53_HUMAN	0	-10.801
4nuuA	4nuuAC	PVDR_PLAVS	ACKR1_HUMAN	0.16668	-10.796
5ee4A	5ee4AD	FOEX68_9NEIS	HBB_HUMAN	0.06444	-10.77
2j1kQ	2j1kAQ	CXAR_HUMAN	SPIKE_ADECT	0.86192	-10.76
4zkcA	4zkcAB	Q2F862_ORFV	CCL7_HUMAN	0.659	-10.759
2h4jA	2h4jAD	NPD_THEMA	P53_HUMAN	0.16247	-10.757
1tf0B	1tf0AB	ALBU_HUMAN	PAB_FINMA	0.91207	-10.624
6eqiC	6eqiAC	UBC_HUMAN	EOW1I1_PEDHC	0.13391	-10.6
1pywD	1pywAD	DRA_HUMAN	ENTC3_STAAM	0.22287	-10.576
1us7A	1us7AB	HSP82_YEAST	CDC37_HUMAN	0.30434	-10.555
4lqwD	4lqwAD	RBP2_HUMAN	POL_HV1N5	0.28412	-10.551
2o39A	2o39AC	SPIKE_ADE1P	MCP_HUMAN	0.06134	-10.533
4b4nA	4b4nAB	A9PKC6_9HIV1	CPSF6_HUMAN	0.62948	-10.523
5crwA	5crwAB	PDI_HUMIN	SYUA_HUMAN	0.26503	-10.452
4c56C	4c56CD	ETXB_STAAU	DRA_HUMAN	0.96788	-10.448
5nt2E	5nt2EV	NS1_I34A1	TRI25_HUMAN	0.20916	-10.404
3szkC	3szkAC	HBA_HUMAN	ISDH_STAAW	0.10252	-10.401
4ij2G	4ij2AG	HBA_HUMAN	ISDH_STAA3	0.0195	-10.387
6bvvB	6bvvAB	IMA3_HUMAN	W_NIPAV	0.99975	-10.372

1f5qB	1f5qAB	CDK2_HUMAN	P89883_MHV68	0.98547	-10.33
4xnzA	4xnzAB	Q0ED31_9HIV1	IGHG1_HUMAN	0.55478	-10.296
3o8eA	3o8eAB	Q772X2_9ADEN	MCP_HUMAN	0.02535	-10.269
2n3kB	2n3kAB	BRD4_HUMAN	Q8UN00_MLVMO	0.50408	-10.246
4zkbA	4zkbAB	Q2F862_ORFV	CCL3_HUMAN	0.48517	-10.202
5yijA	5yijAD	Q6RCR0_LEGPN	UBB_HUMAN	0.03143	-10.198
1lqsL	1lqsLR	IL10H_HCMVA	I10R1_HUMAN	0.22947	-10.193
3wooA	3wooAC	DAPB2_PSEMX	ANGT_HUMAN	0.96453	-10.187
2y7IA	2y7IAB	ALS9_CANAL	FIBG_HUMAN	0.59886	-10.159
3p8mC	3p8mBC	DYL2_HUMAN	GCN4_YEAST	0.28145	-10.132
2xs1B	2xs1AB	PDC6I_HUMAN	GAG_SIVMK	0.51461	-10.091
2ny1A	2ny1AD	Q993A8_9HIV1	Q6N030_HUMAN	0.00211	-10.062
3cxbA	3cxbAB	SIFA_SALTY	PKHM2_HUMAN	0.66399	-10.058
2ibfD	2ibfAD	VINC_HUMAN	IPAA_SHIFL	0.34816	-10.048
3ovuB	3ovuBC	ISDH_STAAS	HBA_HUMAN	0.08187	-10.045
5nbqG	5nbqDG	CFAH_HUMAN	Q45001_BORBG	0.66425	-9.946
5uclB	5uclAB	UBB_HUMAN	VPS27_YEAST	0.06556	-9.933
2a19A	2a19AB	IF2A_YEAST	E2AK2_HUMAN	0.24592	-9.926
2h1IA	2h1IAM	LT_SV40	P53_HUMAN	0.43546	-9.907
1g9nG	1g9nCG	CD4_HUMAN	ENV_HV1Y2	0.4369	-9.883
5fcgC	5fcgAC	BCL2_HUMAN	X_HBVF4	0.87245	-9.859
3d5rC	3d5rAC	CO3_HUMAN	FIB_STAAE	0.55242	-9.851
2winQ	2winBQ	CO3_HUMAN	SCIN_STAAR	0.0043	-9.848
3nvnA	3nvnAB	SEMA_ECTVM	PLXC1_HUMAN	0.9473	-9.846
4owrC	4owrAC	RAE1L_HUMAN	MATRX_VSIVS	0.41539	-9.778
3kbhE	3kbhAE	ACE2_HUMAN	SPIKE_CVHNL	0.06113	-9.73
6azpB	6azpAB	PERM_HUMAN	Q2G0X2_STAA8	0.76	-9.703
6cplC	6cplBC	2B1B_HUMAN	POL_HV1H2	2.6E-06	-9.7
2dohC	2dohCX	PAM_STRPY	PLMN_HUMAN	0.65462	-9.684
4dcbA	4dcbAF	COLY_YERPE	PLMN_HUMAN	0.47752	-9.624
3lqaG	3lqaCG	CD4_HUMAN	Q1PHM6_9HIV1	0.19348	-9.606
2nxyA	2nxyAD	Q49DS8_9HIV1	Q6N095_HUMAN	0.3296	-9.602
2ajfE	2ajfAE	ACE2_HUMAN	SPIKE_CVHSA	0.05698	-9.584
5uwqC	5uwqCD	XPO1_YEAST	CDC7_HUMAN	0.91079	-9.557
3besR	3besLR	IFNG_HUMAN	Q66793_9POXV	0.03596	-9.523
3l89A	3l89AM	Q2KS96_9ADEN	MCP_HUMAN	0.01113	-9.522
4xmpG	4xmpGH	Q8JDI3_9HIV1	IGHG1_HUMAN	0.26503	-9.519
5iroD	5iroAD	1A02_HUMAN	Q8BEL5_ADE04	0.12893	-9.486
3rc5A	3rc5AB	POLG_HCVH	MAVS_HUMAN	0.27741	-9.443
3eoyA	3eoyAG	SIGM1_REOVD	JAM1_HUMAN	0.64073	-9.432
2nojB	2nojAB	CO3_HUMAN	A0A0H3JUX1_STAAM	0.12559	-9.388
4ij2E	4ij2BE	HBB_HUMAN	ISDH_STAA3	0.00174	-9.302
3gytA	3gytAB	Q9XZJ5_STRER	NCOA1_HUMAN	0.56721	-9.296
1t6bX	1t6bXY	PAG_BACAN	ANTR2_HUMAN	0.81055	-9.206
4wjg3	4wjg3Z	ISDH_STAAN	HBA_HUMAN	0.00221	-9.156

4jm2E	4jm2EF	Q75760_9HIV1	CD4_HUMAN	0.40837	-9.005
2rhkA	2rhkAC	NS1_I72A2	CPSF4_HUMAN	0.47314	-8.923
1kacA	1kacAB	SPIKE_ADE12	CXAR_HUMAN	0.69917	-8.911
6cp0A	6cp0AB	Q6RCR3_LEGPN	UB2D3_HUMAN	0.05782	-8.907
4un2B	4un2AB	UBC_HUMAN	DSK2_YEAST	0.10615	-8.833
3sqvA	3sqvAC	A0A0H3JDV8_ECO57	UB2L3_HUMAN	0.46108	-8.803
5tghB	5tghAB	SNX5_HUMAN	B7SCI5_CHLTH	0.50113	-8.754
5imyA	5imyAC	B2YGA4_GARVA	CD59_HUMAN	0.3031	-8.748
1i5kC	1i5kAC	PLMN_HUMAN	PAM_STRPY	0.67693	-8.726
4q5eA	4q5eAC	OSPG_SHISS	UB2L3_HUMAN	0.40688	-8.713
5fobC	5fobBC	CO3_HUMAN	Q89859_VAR67	0.20916	-8.71
1avzB	1avzBC	NEF_HV1BR	FYN_HUMAN	0.28767	-8.694
3prxX	3prxAX	CO5_HUMAN	D3JIB2_STAAU	0.205	-8.594
1im3D	1im3AD	1A02_HUMAN	US02_HCMVA	0.05414	-8.588
4wjgD	4wjgAD	HBA_HUMAN	ISDH_STAAN	0.00222	-8.534
4wz3B	4wz3AB	UB2D2_HUMAN	LUBX_LEGPA	0.12824	-8.496
2mroB	2mroAB	UBB_HUMAN	DDI1_YEAST	0.31575	-8.474
3km9X	3km9AX	CO5_HUMAN	A0A0H3K6Z8_STAAE	0.22788	-8.457
3inbA	3inbAD	HEMA_MEASE	MCP_HUMAN	0.01171	-8.432
5moyA	5moyAB	BXA2_CLOBJ	SV2C_HUMAN	0.01483	-8.42
4kdiC	4kdiAC	TERA_HUMAN	OTU1_YEAST	0.15137	-8.393
1fxtA	1fxtAB	UBC1_YEAST	UBC_HUMAN	0.81937	-8.346
1efnB	1efnAB	FYN_HUMAN	NEF_HV1BR	0.47285	-8.253
4lcdA	4lcdAE	RSP5_YEAST	UBC_HUMAN	0.46779	-8.245
1y6mL	1y6mLR	IL10H_EBVB9	I10R1_HUMAN	0.01585	-8.225
4n9fj	4n9fCj	CUL5_HUMAN	VIF_HV1N5	0.55113	-8.213
4cc4E	4cc4EF	Q8Y6A8_LISMO	DNMBP_HUMAN	0.15542	-8.209
1lj2A	1lj2AC	NSP3_ROTST1	IF4G1_HUMAN	0	-8.164
5w1mR	5w1mBR	Q6N089_HUMAN	Q8AZ57_MACHU	0.03905	-8.114
5onpA	5onpAB	THER_GEOSE	A4_HUMAN	0.98733	-8.067
3f9kB	3f9kBC	POL_HV2RO	PSIP1_HUMAN	0.16247	-8.067
5jdoA	5jdoAE	G0UVW6_TRYCI	HBA_HUMAN	0.40131	-8.02
6cp2A	6cp2AB	Q6RCR4_LEGPN	UB2L3_HUMAN	0.17439	-8.001
1hezE	1hezAE	IGKC_HUMAN	Q51918_FINMA	0.47752	-7.988
5hytA	5hytAB	Q54901_STRPY	C4BPA_HUMAN	0	-7.97
3kmtA	3kmtAG	O41094_PBCV1	H31_HUMAN	0.9126	-7.941
4pdcE	4pdcAE	NKG2D_HUMAN	Q8QN43_COWPX	0.7907	-7.93
4jraA	4jraAD	BXA1_CLOBO	SV2C_HUMAN	0.16707	-7.928
5yikA	5yikAC	Q6RCR0_LEGPN	UBB_HUMAN	0.10733	-7.912
1i1rB	1i1rAB	IL6RB_HUMAN	Q98823_HHV8	0.43659	-7.887
2nz1A	2nz1AE	O41925_MHV68	CCL2_HUMAN	0.03565	-7.877
6acgC	6acgCD	SPIKE_CVHSA	ACE2_HUMAN	0.02165	-7.862
2b4jA	2b4jAD	POL_HV1N5	PSIP1_HUMAN	0.20916	-7.851
1buiC	1buiAC	PLMN_HUMAN	SAK_BPP42	0.10919	-7.843
5yikA	5yikAF	Q6RCR0_LEGPN	UBB_HUMAN	0.06135	-7.815

2ny6A	2ny6AD	Q8Q7Z9_9HIV1	Q6N030_HUMAN	0.3296	-7.776
6atkD	6atkAD	AMPN_HUMAN	SPIKE_CVH22	0.13794	-7.769
4iw4C	4iw4CF	ECOT_ECOLI	MASP1_HUMAN	0.56414	-7.768
3onaA	3onaAB	Q7TDW8_9POXV	X3CL1_HUMAN	0.51074	-7.741
2pmsC	2pmsAC	TRFL_HUMAN	Q54972_STREE	0.03748	-7.73
3mfgA	3mfgAB	A0A0H3JMU6_STAAN	A0A5B4_HUMAN	0.0904	-7.729
4rt4E	4rt4AE	DPY30_HUMAN	BRE2_YEAST	0.20916	-7.724
5hu6D	5hu6BD	HBB_HUMAN	Q581F2_TRYB2	0.16247	-7.696
3skuA	3skuAD	Q991M3_HHV1	NECT1_HUMAN	0.54295	-7.603
3cl3A	3cl3AD	Q76RF1_HHV8	NEMO_HUMAN	0.07721	-7.584
5m3hA	5m3hAY	H6QM92_9INFA	RPB1_HUMAN	0.13107	-7.581
5hyuA	5hyuAB	M21_STRPY	C4BPA_HUMAN	0.20916	-7.581
1hqrD	1hqrBD	DRB5_HUMAN	SPEC_STRP8	0.26865	-7.58
4bvuA	4bvuAB	OSPG_SHIFL	UB2D3_HUMAN	0.44511	-7.577
4ij2H	4ij2DH	HBB_HUMAN	ISDH_STAA3	0.00252	-7.565
3it8D	3it8AD	TNFA_HUMAN	Q9DHW0_YLDV	0.00949	-7.543
3sy2A	3sy2AD	SOPA_SALTY	UB2L3_HUMAN	0.16247	-7.523
1gh6A	1gh6AB	LT_SV40	RB_HUMAN	0.47752	-7.523
4ij2F	4ij2CF	HBA_HUMAN	ISDH_STAA3	0.02644	-7.434
5hypB	5hypAB	C4BPA_HUMAN	W0T1Y4_STRPY	0.3296	-7.422
4e5xG	4e5xAG	1A02_HUMAN	E3GL_ADE06	0.26074	-7.419
3uezA	3uezAE	BARF1_EBVB9	CSF1_HUMAN	0.00661	-7.415
4wjg4	4wjg24	HPT_HUMAN	I7BA80_TRYBB	0.03483	-7.415
1buiC	1buiBC	PLMN_HUMAN	SAK_BPP42	0.10295	-7.391
4pdcE	4pdcBE	NKG2D_HUMAN	Q8QN43_COWPX	0.05061	-7.368
4bikA	4bikAB	Q9LCB8_STRIT	CD59_HUMAN	0.38024	-7.351
5m6wQ	5m6wHQ	CO3_HUMAN	SCIN_STAAR	0.16247	-7.325
2verA	2verAN	AFAE3_ECOLX	CEAM5_HUMAN	0.04625	-7.29
5nt1B	5nt1BI	NS1_I34A1	TRI25_HUMAN	3.68E-01	-7.27
1jmaA	1jmaAB	GD_HHV1P	TNR14_HUMAN	0.32496	-7.242
1xx9D	1xx9AD	FA11_HUMAN	ECOT_ECOLI	0.70759	-7.241
4e2iB	4e2i2B	DPOA2_HUMAN	LT_SV40	0.26503	-7.216
4mywC	4mywBC	NECT1_HUMAN	GD_HHV23	0.20916	-7.17
3l5nM	3l5nBM	CO3_HUMAN	SCIN_STAAM	0.02667	-7.149
4rt4E	4rt4BE	DPY30_HUMAN	BRE2_YEAST	0.16247	-7.13
4fa8A	4fa8AE	BARF1_EBVG	CSF1_HUMAN	0.01169	-7.12
3f9kB	3f9kBK	POL_HV2RO	PSIP1_HUMAN	0.14593	-7.096
5m6wQ	5m6wJQ	CFAB_HUMAN	SCIN_STAAR	0.06168	-7.062
3f9kZ	3f9kCZ	PSIP1_HUMAN	POL_HV2RO	0.17312	-7.055
5usrK	5usrBK	LYRM4_HUMAN	ACP_ECOLI	0.1941	-7.037
5wyiB	5wyiBE	B3LNW5_YEAS1	H31_HUMAN	0.39458	-7.034
1kg0C	1kg0BC	2B11_HUMAN	GP42_EBVB9	0.37717	-7
5vmmJ	5vmmDJ	HBB_HUMAN	ISDB_STAAN	0.03568	-6.968

4cguA	4cguAC	TAH1_YEAST	HS90A_HUMAN	0.3296	-6.959
4wjg4	4wjg14	HBB_HUMAN	I7BA80_TRYBB	0.16247	-6.899
2wbwA	2wbwAB	Q64823_9ADEN	CXAR_HUMAN	0.40495	-6.896
5nt1J	5nt1AJ	TRI25_HUMAN	NS1_I34A1	0.35385	-6.866
2h4fA	2h4fAD	NPD_THEMA	P53_HUMAN	0.26503	-6.794
2mwsB	2mwsAB	UBC_HUMAN	DDI1_YEAST	0.13409	-6.786
1ktkA	1ktkAE	SPEC_STRP8	TRBC1_HUMAN	0.47752	-6.782
3epf1	3epf1R	POLG_POL2L	PVR_HUMAN	0.11296	-6.781
2ntsA	2ntsAP	O54476_STAAU	TRBC1_HUMAN	0.40131	-6.75
3ddaA	3ddaAB	BXA1_CLOBO	SNP25_HUMAN	0.40131	-6.727
5vqmA	5vqmAH	C9YJ35_PEPDR	S6B291_HUMAN	0.16247	-6.658
1lj2A	1lj2AD	NSP3_ROT51	IF4G1_HUMAN	0.25962	-6.651
4u0cA	4u0cAB	GAG_HV1N5	NU153_HUMAN	0.16247	-6.573
4jioA	4jioAU	GOVBG6_NAUCC	RL40_HUMAN	0.16247	-6.547
3d2uE	3d2uEH	UL18_HCMVA	LIRB1_HUMAN	0.02511	-6.519
5nem1	5nem1B	Q98W00_9PICO	ITB6_HUMAN	0.68676	-6.517
5onrA	5onrAB	THER_BACTH	A4_HUMAN	0.965	-6.504
5vmmH	5vmmBH	HBB_HUMAN	ISDB_STAAN	0.06941	-6.5
1jwmD	1jwmAD	DRA_HUMAN	ENTC3_STAAU	0.6202	-6.486
4e2iJ	4e2iJU	LT_SV40	DPOA2_HUMAN	0.12553	-6.422
5net1	5net1B	Q6PMW3_9PICO	ITB6_HUMAN	0.85847	-6.39
3t4aG	3t4aBG	CO3_HUMAN	A0A0H3JZ23_STAAM	0	-6.363
2n3wA	2n3wAB	RPN1_YEAST	RL40_HUMAN	0.10476	-6.36
1hxyD	1hxyBD	2B11_HUMAN	ETXH_STAAU	NO EPPIC	-6.292
5w6gA	5w6gAH	A7UPX0_9INFA	S6B2B6_HUMAN	6.03E- 05	-6.28
2winM	2winIM	CFAB_HUMAN	SCIN_STAAR	0.08322	-6.277
4e2iC	4e2i3C	DPOA2_HUMAN	LT_SV40	0.20916	-6.268
2xpxA	2xpxAB	EAR_EBVB9	BAK_HUMAN	0.99008	-6.26
2nyzA	2nyzAD	O41925_MHV68	XCL1_HUMAN	0.47752	-6.186
3r66B	3r66BC	NS1_INBLE	ISG15_HUMAN	0.3125	-6.172
1gzsB	1gzsBC	SOPE_SALTM	CDC42_HUMAN	0.20916	-6.116
5jdoB	5jdoBF	G0UVW6_TRYCI	HBB_HUMAN	0	-6.081
4ogrD	4ogrBD	CCNT1_HUMAN	TAT_HV1H3	0.71891	-6.064
5ol0A	5ol0AC	Q8I6E4_LEIIN	P53_HUMAN	0.26503	-6.028
2i5yG	2i5yGH	ENV_HV1Y2	Q5EBM2_HUMAN	0.26503	-6.002
3hpgC	3hpgCJ	POL_VILVK	PSIP1_HUMAN	0.16247	-5.937
1lo5D	1lo5AD	DRA_HUMAN	ETXA_STAAU	0.3949	-5.898
1wa7B	1wa7AB	LYN_HUMAN	TIP_SHV2C	0.45517	-5.869
2qadA	2qadAD	ENV_HV1Y2	A4F255_HUMAN	0.33802	-5.856
5tdrA	5tdrAB	SET3_YEAST	H31_HUMAN	0.20916	-5.831
2phgB	2phgAB	TF2B_HUMAN	VP16_HHV11	0.35108	-5.815
5nt2D	5nt2DV	NS1_I34A1	TRI25_HUMAN	0	-5.811
5fobC	5fobAC	CO3_HUMAN	Q89859_VAR67	0	-5.8

1fccC	1fccAC	IGHG1_HUMAN	SPG2_STRSG	0.66049	-5.765
3rc4A	3rc4AB	POLG_HCVH	TCAM1_HUMAN	0.28996	-5.712
5imyA	5imyAD	B2YGA4_GARVA	CD59_HUMAN	0.27311	-5.662
4grgA	4grgAD	L7MTK7_ECOLX	IGHE_HUMAN	0.16247	-5.648
1ktkB	1ktkBF	SPEC_STRP8	TRBC1_HUMAN	0.40131	-5.647
4e2iL	4e2iLW	LT_SV40	DPOA2_HUMAN	0.16247	-5.643
5w1mS	5w1mBS	Q6N089_HUMAN	Q8AZ57_MACHU	0.09945	-5.612
2pkgC	2pkgAC	2AAA_HUMAN	ST_SV40	0.16247	-5.587
3v3kH	3v3kHK	NLEF_ECO57	CASP9_HUMAN	0	-5.575
3f9kA	3f9kAC	POL_HV2RO	PSIP1_HUMAN	0.16247	-5.563
4pdcF	4pdcBF	NKG2D_HUMAN	Q8QN43_COWPX	0.00055	-5.555
5w1kE	5w1kEF	GLYC_JUNIN	IGK_HUMAN	0	-5.434
3epd1	3epd1R	AOA0B6VQW0_9ENTO	PVR_HUMAN	0.01226	-5.431
4u0qC	4u0qBC	BASI_HUMAN	B2L3N7_PLAFA	0.00546	-5.397
5hu6D	5hu6CD	HPT_HUMAN	Q581F2_TRYB2	0.08341	-5.311
4zncC	4zncCE	SPA_STAAU	IGHG3_HUMAN	0.18733	-5.311
3prmC	3prmBC	UBC_HUMAN	Q6TFZ7_9VIRU	0.30901	-5.286
6gh8B	6gh8BC	C5ILC1_9VIRU	NRP2_HUMAN	0.20326	-5.233
4cgqA	4cgqAQ	TAH1_YEAST	HS90A_HUMAN	0.16247	-5.227
2n3vA	2n3vAB	RPN1_YEAST	RL40_HUMAN	0.16247	-5.225
2j12A	2j12AB	Q80S15_9ADEN	CXAR_HUMAN	0.6496	-5.219
1i4eA	1i4eAB	P35_NPVAC	CASP8_HUMAN	0.99158	-5.182
4hkjD	4hkjBD	B2MG_HUMAN	Q8QMP2_COWPX	0.02362	-5.178
5hytG	5hytFG	C4BPA_HUMAN	Q54901_STRPY	0.16247	-5.079
1i5kD	1i5kAD	PLMN_HUMAN	PAM_STRPY	0.26503	-5.068
4gjtA	4gjtAC	HEMA_MEASC	NECT4_HUMAN	0.16247	-5.02
2n3uA	2n3uAC	RPN1_YEAST	RL40_HUMAN	0.16247	-5.002
3it8F	3it8AF	TNFA_HUMAN	Q9DHW0_YLDV	0.20916	-4.98
1upnB	1upnBE	Q8JKE8_9ENTO	DAF_HUMAN	0.001	-4.872
2gs0A	2gs0AB	TFB1_YEAST	P53_HUMAN	0.16247	-4.849
4m38A	4m38AE	ANM7_TRYB2	H4_HUMAN	0.61109	-4.848
4en2B	4en2BE	Q90VU7_9HIV1	1A02_HUMAN	0.08672	-4.815
5w1kP	5w1kAP	IGK_HUMAN	GLYC_JUNIN	1.62E-01	-4.751
4z8IB	4z8IBD	Q7ZB17_SIV	DCAF1_HUMAN	0.02398	-4.705
3hphA	3hphAE	POL_VILVK	PSIP1_HUMAN	0	-4.638
3u82A	3u82AB	GD_HHV11	NECT1_HUMAN	0.80292	-4.587
4y49B	4y49BE	ARD1_YEAST	COLI_HUMAN	0.24641	-4.58
4or5C	4or5CE	TAT_HV1H2	AFF4_HUMAN	0	-4.575
5vqmA	5vqmAL	C9YJ35_PEPDR	Q8TCD0_HUMAN	0.00047	-4.562
5m3jA	5m3jAX	Q5V8Z9_9INFB	RPB1_HUMAN	0.11774	-4.549
5diyA	5diyAP	D1CDN2_THET1	TAB1_HUMAN	0	-4.548
3d2eC	3d2eCD	HSP7F_YEAST	HS71A_HUMAN	0.17853	-4.538
3iypC	3iypCF	Q6W9E5_9ENTO	DAF_HUMAN	0.00254	-4.456

3prmA	3prmAD	Q6TFZ7_9VIRU	UBC_HUMAN	0.32437	-4.441
2gj7F	2gj7AF	IGHG1_HUMAN	GE_HHV11	0.9438	-4.417
4wymK	4wymKW	POL_HV1N5	CPSF6_HUMAN	0	-4.381
4wymE	4wymER	POL_HV1N5	CPSF6_HUMAN	0	-4.369
1k8rB	1k8rAB	RASH_HUMAN	BYR2_SCHPO	0.22315	-4.343
5w1kE	5w1kDE	S6B2B6_HUMAN	GLYC_JUNIN	7.91E-05	-4.313
5urnA	5urnAB	TFB1_YEAST	TF65_HUMAN	0.20916	-4.293
5kq1A	5kq1AF	DCP1_SCHPO	PNRC2_HUMAN	0.20916	-4.286
3u31A	3u31AB	SIR5_PLAF7	H31_HUMAN	0.85257	-4.271
6c4uC	6c4uCI	RAD53_YEAST	MYC_HUMAN	0.25329	-4.254
2ffkA	2ffkAB	O10647_9POXV	CCL4_HUMAN	0.55249	-4.241
3epc1	3epc1R	POLG_POL1M	PVR_HUMAN	0.14346	-4.196
4u5wA	4u5wAD	NEF_HV1A2	HCK_HUMAN	0	-4.189
4xnhB	4xnhBF	ARD1_YEAST	COLI_HUMAN	0.50335	-4.184
2l2iA	2l2iAB	TFB1_YEAST	KLF1_HUMAN	0.44622	-4.055
4lqwC	4lqwAC	RBP2_HUMAN	POL_HV1N5	0.08138	-4.026
5hytA	5hytAD	Q54901_STRPY	C4BPA_HUMAN	0.16247	-3.973
6dm4D	6dm4DH	Q5ZWF6_LEGPH	SHC1_HUMAN	0.16247	-3.962
1m9xG	1m9xAG	PPIA_HUMAN	Q72497_9HIV1	0	-3.954
3l89J	3l89JX	Q2KS96_9ADEN	MCP_HUMAN	0	-3.948
4zncA	4zncAD	SPA_STAAU	IGHG3_HUMAN	0.0281	-3.819
5w8uC	5w8uBC	ISG15_HUMAN	M4STU1_9BETC	0.16247	-3.807
5nj9A	5nj9AE	TLDD_ECOLI	ANGT_HUMAN	0.09927	-3.712
3epd3	3epd3R	AOA0B4MVN6_9ENTO	PVR_HUMAN	0.00197	-3.70946
5o1rA	5o1rAL	Q9JPP1_NEIME	Q6GMX0_HUMAN	0.00143	-3.699
5ln1A	5ln1AU	RPN10_YEAST	UBB_HUMAN	0.20916	-3.637
3jrvB	3jrvBE	K7_VACCW	DDX3X_HUMAN	0	-3.608
5w1kE	5w1KEG	GLYC_JUNIN	S6C4R2_HUMAN	0.3296	-3.602
2b4jA	2b4jAC	POL_HV1N5	PSIP1_HUMAN	0.16247	-3.596
2n3wA	2n3wAC	RPN1_YEAST	RL40_HUMAN	0	-3.579
5w1mR	5w1mAR	Q0KKI6_HUMAN	Q8AZ57_MACHU	0	-3.575
2hd4A	2hd4AB	PRTK_PARAQ	TRFL_HUMAN	0.55565	-3.544
3iozA	3iozAB	Q5QGG3_SIV	CD3Z_HUMAN	0.19306	-3.51
2pheC	2pheAC	TCP4_HUMAN	VP16_HHV11	0.19743	-3.486
5kq4D	5kq4CD	PNRC2_HUMAN	DCP1_SCHPO	0.20916	-3.448
1z7z1	1z7z1I	Q7T7N6_9ENTO	ICAM1_HUMAN	0.00512	-3.437
5m3hA	5m3hAX	H6QM92_9INFA	RPB1_HUMAN	0.20916	-3.396
5usrI	5usrAI	NFS1_HUMAN	ACP_ECOLI	0.20916	-3.385
5w1kP	5w1kBP	S6C4R2_HUMAN	GLYC_JUNIN	0.3296	-3.372
5o74K	5o74BK	RAB1B_HUMAN	DRRA_LEGPN	0	-3.34
1k5dC	1k5dCJ	RNA1_SCHPO	RAN_HUMAN	0	-3.339
4h1IJ	4h1IAJ	DRA_HUMAN	L7MTLO_ECOLX	0.16247	-3.332
1f5qB	1f5qBC	P89883_MHV68	CDK2_HUMAN	0.00326	-3.29
4n9f1	4n9f13	VIF_HV1N5	CUL5_HUMAN	3.48E-01	-3.267

2bovB	2bovAB	RALA_HUMAN	ARC3_CBDP	0.20916	-3.261
4y49H	4y49HK	ARD1_YEAST	COLI_HUMAN	0.30741	-3.231
6atgC	6atgAC	A0A024R962_HUMAN	O50665_BORBU	0.26503	-3.214
2pheC	2pheBC	TCP4_HUMAN	VP16_HHV11	0	-3.184
4fa8B	4fa8BG	BARF1_EBVG	CSF1_HUMAN	0	-3.166
2pkgC	2pkgBC	2AAA_HUMAN	ST_SV40	0.20916	-3.09
4u0bA	4u0bAN	GAG_HV1N5	CPSF6_HUMAN	0	-3.086
3ve1C	3ve1BC	TRFE_HUMAN	TBB1_NEIMI	0.16247	-3.045
1ee4A	1ee4AD	IMA1_YEAST	MYC_HUMAN	0.16247	-3.035
4nm8B	4nm8BH	HEMA_I68A4	S6C4S0_HUMAN	3.32E-05	-3.017
1z1dB	1z1dAB	RFA2_HUMAN	LT_SV40	0.21665	-2.997
6d03E	6d03CE	TRFE_HUMAN	A5K736_PLAVS	0	-2.981
3epf3	3epf3R	POLG_POL2L	PVR_HUMAN	0.01433	-2.942
3t4aH	3t4aDH	CO3_HUMAN	A0A0H3JZ23_STAAM	0	-2.912
3sdlA	3sdlAC	NS1_INBLE	ISG15_HUMAN	0.58139	-2.9
3uezB	3uezBH	BARF1_EBVB9	CSF1_HUMAN	0.16247	-2.876
3t4aG	3t4aAG	CO3_HUMAN	A0A0H3JZ23_STAAM	0	-2.863
4wymF	4wymFM	POL_HV1N5	CPSF6_HUMAN	0	-2.846
4wymC	4wymCP	POL_HV1N5	CPSF6_HUMAN	0.16247	-2.845
4wymB	4wymBO	POL_HV1N5	CPSF6_HUMAN	0	-2.845
4u0qA	4u0qAD	B2L3N7_PLAFA	BASI_HUMAN	0.16247	-2.832
6dm4A	6dm4AE	Q5ZWF6_LEGPH	SHC1_HUMAN	0.16247	-2.798
3phwG	3phwBG	RS27A_HUMAN	L_CCHFI	0.26503	-2.797
5kdmD	5kdmCD	DAXX_HUMAN	MTP_EBVA8	0.37544	-2.783
3phwA	3phwAH	L_CCHFI	RS27A_HUMAN	0.26503	-2.734
2n23A	2n23AB	TFB1_YEAST	KLF1_HUMAN	0	-2.645
5nuhA	5nuhAC	Q5QGG3_SIV	HCK_HUMAN	0	-2.591
5nt2D	5nt2AD	TRI25_HUMAN	NS1_I34A1	0	-2.543
3hi1J	3hi1AJ	Q6PIL8_HUMAN	ENV_HV1Y2	0	-2.514
5hytC	5hytBC	C4BPA_HUMAN	Q54901_STRPY	0.16247	-2.47
5kq1D	5kq1CD	PNRC2_HUMAN	DCP1_SCHPO	0.20916	-2.452
2wdtC	2wdtCD	Q8IKM8_PLAF7	UBC_HUMAN	0.27	-2.361
1z7z2	1z7z2I	Q7T7N6_9ENTO	ICAM1_HUMAN	0.16247	-2.307
2dp4E	2dp4EI	PRTK_PARAQ	TRFL_HUMAN	0.95401	-2.302
1v9u1	1v9u15	POLG_HRV2	VLDLR_HUMAN	0	-2.271
2pkgD	2pkgAD	2AAA_HUMAN	ST_SV40	0	-2.035
5w1mQ	5w1mAQ	Q0KKI6_HUMAN	Q8AZ57_MACHU	0	-1.955
3l3oP	3l3oBP	CO3_HUMAN	SCIN_STAAM	0.00732	-1.779
6d04F	6d04CF	TRFE_HUMAN	A5K736_PLAVS	0	-1.742
1ytvA	1ytvAM	MALE_ECOLI	V1AR_HUMAN	0.3296	-1.583
3l5nM	3l5nAM	CO3_HUMAN	SCIN_STAAM	0	-1.479
5jlvB	5jlvBC	BXA1_CLOBO	SV2C_HUMAN	0.00336	-1.405
5m5gB	5m5gBD	GOSDW4_CHATD	H31_HUMAN	0	-1.257

6a5rB	6a5rBg	C4QZQ7_KOMPG	H2A1B_HUMAN	0.16247	-1.147
5hplB	5hplBC	RSP5_YEAST	RL40_HUMAN	0	-0.417
5a5b8	5a5b89	UBP6_YEAST	RL40_HUMAN	0.94742	0
5jnxE	5jnxAE	NPC1_HUMAN	VGP_EBOZ5	0.1861	Positive Rosetta total energy score
1jew1	1jew1R	POLG_CXB3W	CXAR_HUMAN	0.02773	Clash Count> 5 (the original template interface has 7 clashes)
4i9xA	4i9xAD	UL141_HCMVM	TR10B_HUMAN	0.62873	Incorrect alignment by TM-align: TM-score: 0.98768
2omtA	2omtAB	A4GWM6_LISMN	CADH1_HUMAN	0.70304	Incorrect alignment by TM-align: TM- score: 0.78617
1ee4A	1ee4AC	IMA1_YEAST	MYC_HUMAN	0.6866	Incorrect alignment by TM-align: TM- score: 0.84467
2omxA	2omxAB	A4GWL5_LISMN	CADH1_HUMAN	0.78088	Incorrect alignment by TM-align: TM- score: 0.95334
5o74A	5o74AD	DRRA_LEGPN	RAB1B_HUMAN	3.01E- 05	Incorrect alignment by TM-align: TM- score: 0.47697
1k5dC	1k5dAC	RAN_HUMAN	RNA1_SCHPO	0.99688	Incorrect alignment by TM-align: TM- score: 0.79518
5o1rA	5o1rAH	Q9JPP1_NEIME	Q6GMX6_HUMAN	0.00256	Incorrect alignment by TM-align: TM- score: 0.40537
1o6sA	1o6sAB	INLA_LISMO	CADH1_HUMAN	0.50137	Incorrect alignment by TM-align: TM- score: 0.65577
4uf7A	4uf7AC	I0E093_9MONO	EFNB2_HUMAN	0.01372	Incorrect alignment by TM-align: TM-

					score: 0.41002
2vr3A	2vr3AC	CLFA_STAA8	FIBG_HUMAN	0.32786	Incorrect alignment by TM-align:TM- score: 0.5172
4rf0A	4rf0AB	M4STU1_9BETC	RL40_HUMAN	0.76355	Incorrect alignment by TM-align:TM- score: 0.34345
5kjhA	5kjhAD	G0S8H7_CHATD	H31_HUMAN	0.3296	Incorrect alignment by TM-align:TM- score: 0.2573
5oy9A	5oy9AD	VGLG_VSIVM	LDLR_HUMAN	0.17052	Incorrect alignment by TM-align:TM- score: 0.27087
2ydrA	2ydrAP	OGA_CLOP1	P53_HUMAN	0	Incorrect alignment by TM-align:TM- score: 0.20739
2ydsA	2ydsAT	OGA_CLOP1	TAB1_HUMAN	0	Incorrect alignment by TM-align:TM- score: 0.19707
2ydqA	2ydqAT	OGA_CLOP1	OGA_HUMAN	0	Incorrect alignment by TM-align:TM- score: 0.16871
2xn9C	2xn9CE	ETXH_STAAU	2B11_HUMAN	0.05754	Incorrect alignment by TM-align:TM- score: 0.44336
2g9hD	2g9hBD	2B11_HUMAN	Q52T95_STAAU	0.37724	Incorrect alignment by TM-align:TM- score: 0.3199
4u0dA	4u0dAO	GAG_HV1N5	NU153_HUMAN	0.20916	Incorrect alignment by TM-align:TM- score: 0.39703
1hqrD	1hqrCD	MBP_HUMAN	SPEC_STRP8	0	Incorrect alignment by TM-align:TM- score: 0.44277

3ve1A	3ve1AD	TBB1_NEIMI	TRFE_HUMAN	0.16247	Incorrect alignment by TM-align: TM-score: 0.216
2xn9C	2xn9AC	TRAC_HUMAN	ETXH_STAAU	0.26503	Incorrect alignment by TM-align: TM-score: 0.3732
6d05E	6d05DE	TRFE_HUMAN	A5K736_PLAVS	0	Incorrect alignment by TM-align: TM-score: 0.46989
5u4yC	5u4yAC	IGHG1_HUMAN	SPA_STAA8	0.69351	Incorrect alignment by TM-align: TM-score: 0.52995
3pcrA	3pcrAB	Q7DB50_ECO57	ARF6_HUMAN	0.00849	Incorrect alignment by TM-align: TM-score: 0.35225
5uwpC	5uwpCD	XPO1_YEAST	DIAP3_HUMAN	0.94539	Incorrect alignment by TM-align: TM-score: 0.51375
2qadA	2qadAB	ENV_HV1Y2	CD4_HUMAN	0.35333	Incorrect alignment by TM-align: TM-score: 0.26441
2wokA	2wokAB	Q8KRB4_STRCL	KNG1_HUMAN	0.26503	Incorrect alignment by TM-align: TM-score: 0.28072
5vqmB	5vqmBH	C9YJ35_PEPDR	S6B291_HUMAN	0	Incorrect alignment by TM-align: TM-score: 0.35374
5oylA	5oylAD	B7UCZ5_9RHAB	LDLR_HUMAN	0.31757	Incorrect alignment by TM-align: TM-score: 0.30995
5uwwC	5uwwCD	XPO1_YEAST	DEAF1_HUMAN	0.50049	Incorrect alignment by TM-align: TM-score: 0.52333
4nkgA	4nkgAB	SSPH1_SALT1	PKN1_HUMAN	0.15903	Incorrect alignment by

4pufA	4pufAD	SLRP_SALT1	THIO_HUMAN	0.16247	TM-align:TM-score: 0.42264 Incorrect alignment by TM-align:TM-score: 0.87364
4gjtA	4gjtAB	HEMA_MEASC	NECT4_HUMAN	0.76865	Incorrect alignment by TM-align:TM-score: 0.49431
2vsmA	2vsmAB	GLYCP_NIPAV	EFNB2_HUMAN	0.91013	Incorrect alignment by TM-align:TM-score: 0.31484
5lx2A	5lx2AB	C5DN49_LACTC	PI4KB_HUMAN	0.9358	Incorrect alignment by TM-align:TM-score: 0.44125

APPENDIX C

CASE STUDY RESULTS

Human protein	Human protein Name	Template interface	Mimicked/ Disrupted Host PPI	Rosetta l_sc for HMI	l_sc for template PPI	%Match	weight	score1	bio_prob	score2	score3
3qu8E	CP2B6_HUMAN	3qu8CE	CP2B6_HUMAN - CP2B6_HUMAN	-5.996	-5.869	72.2222	1	0.7222	0.265027	0.1914	-1.148
4rduA	DLX5_HUMAN	4rduAD	DLX5_HUMAN - DLX5_HUMAN	-6.411	-7.273	65.5172	0.5	0.3276	0.162465	0.0532	-0.341
6eoqC	DPP9_HUMAN	6eoqAC	DPP9_HUMAN - DPP9_HUMAN	-5.085	-31.859	58.8235	2	1.1765	0.996475	1.1723	-5.961
5jpoB	EF1G_HUMAN	5jpoAB	EF1G_HUMAN - EF1G_HUMAN	-5.53	-21.547	61.6667	1.5	0.925	0.983425	0.9097	-5.03
3ecsC	EI2BA_HUMAN	3ecsCE	EI2BA_HUMAN - EI2BA_HUMAN	-6.031	-6.461	66.6667	0.5	0.3333	0.761051	0.2537	-1.53
3f1nA	EPAS1_HUMAN	3f1nAB	EPAS1_HUMAN - ARNT_HUMAN	-5.999	-11.315	61.5385	1.5	0.9231	0.840764	0.7761	-4.656
6d6qF	EXOS6_HUMAN	6d6qFI	EXOS6_HUMAN - EXOS1_HUMAN	-5.201	-11.721	62.8571	1	0.6286	0.838684	0.5272	-2.742
3h9zB	IGHE_HUMAN	3h9zAB	IGHE_HUMAN - IGHE_HUMAN	-5.418	-18.336	77.3585	1.5	1.1604	0.980297	1.1375	-6.163
6mu3L	IGK_HUMAN	6mu3LM	IGK_HUMAN - IGG1_HUMAN	-5.847	-16.441	79.5455	1	0.7955	0.811533	0.6455	-3.774
3vi3A	ITA5_HUMAN	3vi3AB	ITA5_HUMAN - ITB1_HUMAN	-9.013	-22.318	53.9474	1.5	0.8092	0.999699	0.809	-7.291
5xauC	LAMC1_HUMAN	5xauBC	LAMC1_HUMAN - LAMB1_HUMAN	-6.472	-20.92	35.2113	1.5	0.5282	0.577347	0.3049	-1.974
4n6nA	LGMN_HUMAN	4n6nAB	LGMN_HUMAN - CYTM_HUMAN	-5.058	-9.421	77.1429	1	0.7714	0.977372	0.754	-3.814
4rwtD	LMOD2_HUMAN	4rwtBD	LMOD2_HUMAN - ACT1_DROME	-7.322	-10.566	62.6087	2	1.2522	0.749888	0.939	-6.875
5x6vE	LTOR1_HUMAN	5x6vEG	LTOR1_HUMAN - RRAGC_MOUSE	-6.18	-18.341	79.0698	1	0.7907	0.527367	0.417	-2.577
4mneD	MP2K1_HUMAN	4mneCD	MP2K1_HUMAN - BRAF_HUMAN	-7.315	-19.008	69.6429	1.5	1.0446	0.876669	0.9158	-6.699
2xmrC	NDRG2_HUMAN	2xmrBC	NDRG2_HUMAN - NDRG2_HUMAN	-6.064	-6.931	84.8485	1	0.8485	0.557054	0.4727	-2.866

4frwA	NECT4_HUMAN	4frwAB	NECT4_HUMAN - NECT4_HUMAN	-7.874	-9.133	75	1	0.75	0.162465	0.1218	-0.959
4ey0B	PLCG1_HUMAN	4ey0BC	PLCG1_HUMAN - PLCG1_HUMAN	-7.089	-5.457	93.3333	1	0.9333	0.209159	0.1952	-1.384
1cviB	PPAP_HUMAN	1cviAB	PPAP_HUMAN - PPAP_HUMAN	-6.78	-24.038	69.3548	1.5	1.0403	0.99069	1.0306	-6.988
2pn8H	PRDX4_HUMAN	2pn8AH	PRDX4_HUMAN - PRDX4_HUMAN	-5.299	-13.505	73.5294	1	0.7353	0.839808	0.6175	-3.272
5vhfV	PSMD3_HUMAN	5vhfVY	PSMD3_HUMAN - PSMD6_HUMAN	-7.75	-5.156	51.9231	1.5	0.7788	0.162465	0.1265	-0.981
3i2bE	PTPS_HUMAN	3i2bAE	PTPS_HUMAN - PTPS_HUMAN	-5.399	-12.939	66.6667	1	0.6667	0.556213	0.3708	-2.002
4cz2A	RAB32_HUMAN	4cz2AE	RAB32_HUMAN - ANR27_HUMAN	-6.539	-9.669	77.1429	1	0.7714	0.220432	0.17	-1.112
1h2iA	RAD52_HUMAN	1h2iAB	RAD52_HUMAN - RAD52_HUMAN	-5.718	-42.484	47.5	2	0.95	0.999654	0.9497	-5.43
1zc3C	RALA_HUMAN	1zc3BC	RALA_HUMAN - EXOC8_RAT	-6.958	-9.766	73.3333	1	0.7333	0.833195	0.611	-4.251
5iqqC	RBM7_HUMAN	5iqqBC	RBM7_HUMAN - RBM7_HUMAN	-5.042	-5.402	100	0.5	0.5	0.162465	0.0812	-0.41
3j7yQ	RM19_HUMAN	3j7yLQ	RM19_HUMAN - RM14_HUMAN	-6.022	-12.446	77.7778	1	0.7778	0.68874	0.5357	-3.226
2qkkA	RNH1_HUMAN	2qkkAR	RNH1_HUMAN - RNH1_HUMAN	-6.894	-8.117	65.5172	0.5	0.3276	0.030168	0.0099	-0.068
4pyuC	SNUT1_HUMAN	4pyuAC	SNUT1_HUMAN - UBL5_HUMAN	-5.007	-9.88	69.697	1	0.697	0.320919	0.2237	-1.12
4ol0B	TNPO3_HUMAN	4ol0AB	TNPO3_HUMAN - RAN_HUMAN	-13.392	-18.815	53.9326	2	1.0787	0.843564	0.9099	-12.19
4p22B	UBA1_HUMAN	4p22AB	UBA1_HUMAN - UBA1_HUMAN	-7.823	-9.284	48.2456	2	0.9649	0.999991	0.9649	-7.548
3prmB	UBC_HUMAN	3prmAB	UBC_HUMAN - Q6TFZ7_9VIRU	-5.399	-16.219	45.4545	1.5	0.6818	0.982925	0.6702	-3.618
1j7dB	UBE2N_HUMAN	1j7dAB	UBE2N_HUMAN - UB2V2_HUMAN	-7.43	-15.801	70.9677	1	0.7097	0.707337	0.502	-3.73

4ra0D	UFO_HUMAN	4ra0AD	UFO_HUMAN - GAS6_HUMAN	-5.044	-7.253	77.4194	1	0.7742	0.714741	0.5533	-2.791
1zagC	ZA2G_HUMAN	1zagAC	ZA2G_HUMAN - ZA2G_HUMAN	-6.006	-9.596	66.6667	1	0.6667	0.267753	0.1785	-1.072

APPENDIX D

DETAILS OF KNOWN HMI'S INCORRECT ALIGNMENT

Target	Template	EPPIC score	Match count	Hotspot Analysis	TM-score / Multiprot aligned %	Alignment Method	Interface length	Full chain length	Consecutive Residues Segments lengths			
									Max	Min	Avg	Count
2vr3A	2vr3AC	0.328	59	1	0.5172	TM-align	64	308	14	1	5	11
1o6sA	1o6sAB	0.501	77	1	0.65577	TM-align	106	486	9	1	5	20
2omxA	2omxAB	0.781	95	1	0.95334	TM-align	95	520	9	2	4	20
1k5dC	1k5dAC	0.997	81	1	0.79518	TM-align	84	344	16	1	6	14
4rf0A	4rf0AB	0.764	46	1	0.34345	TM-align	58	318	14	1	6	9
2omtA	2omtAB	0.703	93	1	0.78617	TM-align	95	477	9	1	4	23
5uwwC	5uwwCD	0.5	42	0	0.52333	TM-align	46	1021	18	1	7	6
5u4yC	5u4yAC	0.694	27	0	0.52995	TM-align	31	53	16	14	10	3
1ee4A	1ee4AC	0.687	68	1	0.84467	TM-align	68	423	20	1	9	7
2xn9C	2xn9AC	0.265	26	0	0.3732	TM-align	41	216	15	4	5	7
2qadA	2qadAB	0.353	36	0	0.26441	TM-align	52	319	12	1	6	8
4gitA	4gitAB	0.769	51	0	0.49431	TM-align	64	411	11	3	5	12
4uf7A	4uf7AC	0.014	36	1	0.41002	TM-align	40	423	11	1	4	9
5uwpC	5uwpCD	0.945	34	0	0.51375	TM-align	40	1013	17	1	8	5
4nkgA	4nkgAB	0.159	43	0	0.42264	TM-align	49	233	8	5	5	9
4u0dA	4u0dAO	0.209	24	0	0.39703	TM-align	30	216	11	9	7	4
5oylA	5oylAD	0.318	41	0	0.30995	TM-align	50	409	11	1	4	11
2g9hD	2g9hBD	0.377	23	0	0.3199	TM-align	30	213	8	1	4	7
5oy9A	5oy9AD	0.171	27	1	0.27087	TM-align	44	410	9	1	4	11
2wokA	2wokAB	0.265	36	0	0.28072	TM-align	56	555	10	1	3	15
4pufA	4pufAD	0.162	48	0	0.87364	TM-align	48	599	9	3	6	8
1hqrD	1hqrCD	0	25	0	0.44277	TM-align	27	205	6	5	4	6
2vsmA	2vsmAB	0.91	57	0	0.31484	TM-align	72	430	9	1	4	15
5o74A	5o74AD	3E-05	37	1	0.47697	TM-align	38	191	14	11	9	4
6d05E	6d05DE	0	26	0	0.46989	TM-align	39	466	14	2	6	6

1i5kC	1i5kAC	0.677	10	0	0.52631579	Multiprot	19	26	19	19	9	2
4u0qA	4u0qAD	0.162	10	0	0.45454545	Multiprot	22	286	12	2	5	4
3t4aG	3t4aBG	0	11	0	0.42307692	Multiprot	26	68	19	7	8	3
5hytA	5hytAD	0.162	11	1	0.64705882	Multiprot	17	29	17	17	8	2
4u0qC	4u0qBC	0.005	12	0	0.70588235	Multiprot	17	285	14	3	5	3
1m9xG	1m9xAG	0	12	0	0.54545455	Multiprot	22	150	14	1	5	4
3ve1C	3ve1BC	0.162	12	0	0.48	Multiprot	25	536	6	2	3	7
5kq1D	5kq1CD	0.209	12	0	0.52173913	Multiprot	23	122	8	2	3	6
6d04F	6d04CF	0	12	0	0.48	Multiprot	25	466	14	4	6	4
5hytG	5hytFG	0.162	12	1	0.70588235	Multiprot	17	29	17	17	8	2
5m3hA	5m3hAX	0.209	12	1	0.5	Multiprot	24	699	13	2	4	5
4jioA	4jioAU	0.162	13	0	0.72222222	Multiprot	18	305	18	18	9	2
2qadA	2qadAD	0.338	14	1	0.56	Multiprot	25	319	6	1	3	8
5w8uC	5w8uBC	0.162	14	1	0.53846154	Multiprot	26	319	6	1	3	7
5jdoA	5jdoAE	0.401	15	0	0.51724138	Multiprot	29	250	14	4	7	4
4wjg4	4wjg24	0.035	16	0	0.66666667	Multiprot	24	261	24	24	12	2
3l5nM	3l5nBM	0.027	17	0	0.5483871	Multiprot	31	84	24	1	7	4
2n3wA	2n3wAB	0.105	17	0	0.45945946	Multiprot	37	1310	15	8	9	4
5o74K	5o74BK	0	18	0	0.72	Multiprot	25	191	14	11	8	3
2pmsC	2pmsAC	0.037	19	0	0.38	Multiprot	50	109	20	1	10	5
2wdtC	2wdtCD	0.27	9	0	0.47368421	Multiprot	20	206	13	1	4	4
3l3oP	3l3oBP	0.007	18	1	0.66666667	Multiprot	27	84	14	13	9	3
5vmmJ	5vmmDJ	0.036	21	1	0.95454545	Multiprot	22	135	8	2	4	5
4n9fj	4n9fCj	0.551	33	1	0.97058824	Multiprot	34	170	13	1	4	7
5o1rA	5o1rAH	0.003	23/30	1/1	0.58974/ 0.40537	Multiprot/ TM-align	39	115	9	2	5	7
5kjhA	5kjhAD	0.33	21/32	1/1	0.5833/ 0.2573	Multiprot/ TM-align	36	481	8	1	3	12

3pcrA	3pcrAB	0.008	39/30	1/0	0.975/ 0.35225	Multiprot/ TM-align	40	340	11	1	4	9
4i9xA	4i9xAD	0.629	69/70	1/1	0.985714/ 0.98768	Multiprot/ TM-align	70	181	12	1	5	14
5lx2A	5lx2AB	0.936	20/57	1/0	0.338983/ 0.44125	Multiprot/ TM-align	59	235	17	1	8	7
2ydsA	2ydsAT	0	11/16	0/0	0.4583/ 0.19707	Multiprot/ TM-align	24	579	5	1	2	9
2ydrA	2ydrAP	0	12/14	0/1	0.5/ 0.20739	Multiprot/ TM-align	24	579	4	1	2	10
2ydqA	2ydqAT	0	13/21	0/0	0.52/ 0.16871	Multiprot/ TM-align	25	579	5	1	2	10
2xn9C	2xn9CE	0.058	15/22	0/0	0.555556/ 0.44336	Multiprot/ TM-align	27	216	7	2	4	6
5vqmB	5vqmBH	0	19/37	0/0	0.475/ 0.35374	Multiprot/ TM-align	40	528	14	1	5	7
3ve1A	3ve1AD	0.162	22/26	0/0	0.6470588/ 0.216	Multiprot/ TM-align	34	538	6	2	3	9