Identifying novel pseudo-interfaces in protein cores.

Yotam Constantini, Eyal Perry.

Abstract

In this project we set on the goal of identifying novel interface templates by examining the interior structure of resolved proteins. According to previous research, many interfaces are comprised of 3 secondary structure elements (SSE) on each side. We denote a bilateral interaction between two triplets of SSEs inside a protein core as a *pseudo-interface*. We reduced the problem of identifying such pseudo-interfaces to a graph matching problem. Transforming protein structures into a graphs where each node corresponds to a SSE and edges represent the interaction between such two, we set to find the tightest 3 on 3 cliques, which represent the best candidates for novel interface templates. Strict conditions to the clique selection process were added in order to avoid interactions that are based on backbone torsion. Our method was validated by applying it to a selected set of resolved protein complexes, and identifying our predicted pseudo-interfaces alongside and with better scoring than the known interfaces. Our results expand the span of possible templates used to predict novel interactions between proteins.

Introduction

Biological research of proteins is aimed at unveiling three of their basic properties: function, regulation and interactions with other proteins and biological compounds. The three dimensional structure and the chemical characteristics of a protein are key determinant of both its functions and interactions. The interactions a protein has affect all of its lifespan: translation into the ER, localization to cellular compartment, partnering enzymes in complexes, regulating elements, and proteasome degradation after targeting by ubiquitin ligases. Thus, determining the interactions of a protein is a key objective of biological research.

Many methods have been designed for resolving protein-protein interactions (PPI). Amongst the classical empiric biology methods are mutation analysis, co-immunoprecipitation, yeast two-hybrid and fluorescence resonance energy transfer. X-ray crystallography resolving 3D structures of proteins allowed for a novel approach to figuring PPI, by in-silico thermodynamic analysis of their interactions. Contrary to the swift simplicity by which nature allows recognizing the favorable interaction between structures, a computational analysis traversing the interaction space is hard and time consuming. Thus, different heuristic approaches to the process of protein docking have been devised.

In 1992 former Israeli president Ephraim Katzir has developed a fast algorithm for docking that is based only on geometric matching between structures. Despite ignoring the importance of the chemical properties and the natural flexibility of proteins, the algorithm was validated on known complexes, finding the correct respective positions [Katchalski-Katzir 1992]. Structural, geometric matching results can be further filtered according to physicochemical properties. Newer approaches combine biological properties alongside the geometric (e.g. Hot Spot Filtering in [Duhovny 2002]), allow flexibility (reviewed in [Halperin 2002]) or refine the sidechain configuration [Andrusier 2007].

Analysis of interfaces between proteins have revealed some of their common features. A 1996 review article [Tsai 1996] has speculated that a substantial amount of interfaces are based on secondary structure elements (SSE) scaffolds. Specifically, it was suggested that examining three SSE on either side of the interface could be sufficient to divide most interfaces into different classes.

Two decades of computational prediction and spectral analysis of protein interactions and complexes have prompted researchers to hypothesise that representatives of most classes of different PPI interfaces have already been characterised (eg see [Gao 2010]). Basing on the template docking heuristic, the PRISM system has found novel PPI by superimposing proteins on known interfaces [Ogmen 2005]. In an article utilising the webserver, it was shown that by running the algorithm on 6000 proteins using 70 templates about 63,000 interactions were found and more than 2000 of them were verified as existing in interfaces databases [Keskin 2008].

The chemical forces that govern the PPI are not different from those guiding a single chain fold [Tsai 1996]. Electrostatic attraction and repulsion, hydrogen bonds, steric clashes and hydrophobic interactions all affect the feasible configuration in the core of protein, as much as they influence the possible interactions between proteins. This has brought H.W and N.B.T to suggest that by recognising pseudo interfaces (PI) in the cores of proteins, we will be able to recognise novel interface-class, that will improve the recognition of novel interfaces through template docking. Basing on the SSE scaffold hypothesis we have transformed the PI finding problem into a graph many-to-many, k-on-k matching problem (defined in methods), and will present in this work a method for detecting novel interface templates based on proteins' core pseudo interfaces.

Methods

Structural data and tools

Structural data for proteins was downloaded from RCSB Protein Data Bank (PDB) [Berman 2000]. To parse the PDB files, the BioPython PDB module [Hamelryck 2003] was used. The DSSP method [Kabsch 1983] was used in order to identify the secondary structure elements.

Definitions

C_α distance between two residues is the distance in Angstrom between their C_α atoms.

Two residues are considered to be **interacting** if the minimal distance between any non-backbone and non-hydrogen pair of atoms, is less than 6 Angstrom [Jiang 2003].

Constructing the secondary structure graph

Given a protein P, its sequence, SSE annotation, and structure we define the secondary structure graph SSG = (V, E) as following:

Every node in V represents a continuous SSE in P. We treated the two main types of SSEs - helices and strands. For helices, we define a single node as 6 or more consecutive residues identified by DSSP as any kind of helix (α, 3_{10}, π). For strands, we define a single node as 4 or more consecutive residues identified by DSSP as an extended or isolated beta strand.

We define three terms between a pair of SSE nodes:

1. A distance function D(v, u) is the average of the *k* minimal C_α distances between all residues r_1, r_2 such that $r_1 \in v \; and \; r_2 \in u$. In our implementation we set *k* to 4, which showed optimal empirical results after experimenting with various values ranging from 2 to *n*, with *n* being the minimum between the lengths of the SSE represented by v and u.
2. We define u and v as consecutive, if the number of residues on the protein sequence between the two SSEs the represent is less than S_{min}.
3. An interaction function IC(u, v) is the number of interacting residues r_1, r_2 such that $r_1 \in v \; and \; r_2 \in u$.

For every two nodes v, u we define an edge (v, u) in E iff the following conditions are met:

- D(u, v) is smaller than a defined distanced parameter D_{max}
- u and v are not consecutive.

Those two conditions guarantee that two SSEs are neighbors in the graph if they are in some proximity to each other, while not sequence-adjacent. We make sure that they are not sequence-adjacent in order to ensure that the interaction they induce, if exists, is based solely on chemical forces, and not backbone torsion.

S_{min} was chosen by examining different size loops in a set of proteins. The D_{max} value was chosen after examining structures of protein complexes interfaces, and approximating the

maximal distance between two SSEs that belong to the same interface, while not necessarily being an interacting pair. In our implementation, we set D_{max} to 24 Angstrom and S_{min} to 12 residues. It is important to note that since our method relies on the existence of edges as a prerequisite for further calculations, setting D_{max} too high would have effect only on the running performance of the method without affecting quality of the results. In contrast, setting D_{max} too low will cause the method to miss results (see MMKKM section below). Therefore our choice of D_{max} is much higher than the distances considered in literature for interactions.

Pseudo-interface scoring functions

Let A and B represent two triplets of nodes $A = \{u_1, u_2, u_3\} \subseteq V$ and $B = \{v_1, v_2, v_3\} \subseteq V$ that are disjoint $A \cap B = \varnothing$. We denote the pseudo-interface between the SSEs they represent $PI = (A, B)$. To represent the interaction strength and spatial proximity of a possible pseudo-interface, we define two types of scoring functions which will later help us capture and filter the putative pseudo-interfaces in a protein.

The first scoring function is Sum of Distances - *SD(PI)*. What makes SSEs in an interface distinct from random SSEs in a protein complex is their all-to-all spatial proximity, meaning they are expected to be closer to one another than non-interface SSEs. We define SD as follows:

$$SD(PI = (A, B)) = \sum_{u \in A} \sum_{v \in B} D(u, v)$$

While optimising our method, we have explored different variants of the SD function, amongst them weighted average, and over-weighing the closer SSE pairs. Surprisingly, the simple unweighted sum function performed best, according to our evaluation.

The second scoring function is Total Interaction Count - $TIC(PI)$. The total interaction count is meant to quantify the amount of tight interacting residues between the two sides of PI. We define TIC as follows:

$$TIC(PI) = \sum_{u \in A} \sum_{v \in B} IC(u, v)$$

Basically, the more interactions - the stronger an interface is expected to be as it is naturally occurring.

Many-to-Many-K-on-K-Matching (MMKKM)

We define many-to-many, all-on-all matching problem over graphs as follows:
Given an undirected graph $G = (V, E)$ and a function $f : P(V) \times P(V) \rightarrow \Re$, find a pair of nodes sets $\hat{A}, \hat{B}$ such that:

$$\hat{A}, \hat{B} = maxarg_{A,B}\{f(A, B) \mid A, B \subseteq V, A \cap B = \varnothing, A \times B \subseteq E\}\}$$

The classical many-to-many matching problem is usually a match between agents in two opposite sides - buyers and sellers, interns and workplaces, students and courses etc. (for example [Hatfield 2012]). Our definition alters two aspects of the problem. First, it calculates all of the interactions between agents of the different sides, and not just the matched pairs. Second, the node space is homogenic in nature, and not bipartite. The problem is exponential in the size of the graph, going over all different combinations of two node groups and computing the given function over each combination.

A polynomial bounded version of the above problem is many-to-many, *k*-on-*k* matching (MMKKM). Here, the sizes of the sets is fixated to be k_1 and k_2, thus limiting the possible amount of sets combinations to $(|V| \; choose \; (k_1+k_2)) \cdot (k_1+k_2 \; choose \; k_1) = O(|V|^{k_1+k_2})$. Therefore, the total run time, assuming f to be polynomial in the size of its input, is O($(k_1+k_2)^c \cdot |V|^{k_1+k_2}$).

Revealing pseudo-interfaces

Based on the assumption that interfaces contain at least 3 SSEs on each side, we set to solve the pseudo-interface recognition problem using the MM33M problem on SSG, with a function *f* which represents the quality of a pseudo-interface. We define *f* as follows:

$$f(A,B) = \begin{cases} SD(PI) & \forall u \in A : TIC(\{u\},B) > IC_{\min} \\ \infty & otherwise \end{cases}$$

It is worth noting minimising *f* requires that each node of A has an edge to all nodes of B. Constructing SSG so that continuous SSEs don't have an edge between them, we guarantee optimised pseudo-interface are not held by backbone torsion forces. Our implementation was set to return the list of the top ranking PIs, and their TIC and SD.

The pseudocode for MM33M is as following:

1. for each triplet $A = (u_1,u_2,u_3) \subseteq V$
2. $CloseNeighbors = \{v \in V \setminus A \mid (u_1,v),(u_2,v),(u_3,v) \in E \; and \; TIC(\{v\},A) > IC_{min}\}$
3. remove all $v \in CloseNeighbors$ s.t. $SD(\{v\},A) > SD_{threshold}$
4. for each triplet $B \in P_3(CloseNeighbors)$ s.t. $SD(B,A) < SD_{threshold}$
5. yield (A, B) as a potential pseudo-interface

$SD_{threshold}$ and IC_{min} were empirically optimised as described in the results section, and were finally set $IC_{min} = 4$ and $SD_{threshold} = 94\text{Å}$.

Regarding performance analysis, the MM33M has a worst case of O($|V|^6$) time and O($|V|+|E|$) space. Step 1 is always executed $\approx |V|^3$ times, step 2 and 3 take $O(|V|)$ each. Step 4 loop will be executed $\approx |CloseNeighbors|^3$ which in the worst case is $O(|V|^3)$, and step 5 is $O(1)$.

However, as often in computational biology, real proteins don't tend to be "worst case" input, and application of the D_{max} in construction of the SSG and $SD_{threshold}$ in step 3 will reduce the average size of $CloseNeighbors$ for each triplet to constant, therefore reducing the runtime of step 4 to constant, resulting in total runtime of $O(|V|^4)$. It should be noted that usually the number of SSEs in a single protein chain is not high, and vastly in the lower dozens.

Results

To check the ability of our method to recognise interfaces, we decided to first apply it to known structures of protein complexes, expecting to recognise the inter-protein interface alongside intra-protein PIs. Instead of searching solely for interfaces within a single-chain comprising a protein core, we scanned all chains of the structure together, and singled out results that are disjoint in chain.

From a collection of complexes predicted to be obligatory in biological conditions [Soner 2015], we visually scanned for those with protein-protein interface that is based on a backbone of three SSEs on each side, and were left with a list of 26 complexes. The complexes ranged in size from 230 to 1100 amino acids, and contained between 10 and 50 SSEs.

We started by applying our method using loose parameters: permissive $SD_{threshold}$ and D_{max}, low IC_{min} and a range of *k*'s in the distance definition. These parameters caused high runtime, but allowed us to receive more results (and more potential false positives) and inspect different sorting of the results. For each complex, we have sorted the results by SD, and singled out the sets where A and B belonged to different chains. Early analysis has showed us that between sorting by solely SD, solely TIC or a combination of both, SD-only sorting placed the inter-protein pseudo-interface in a relatively higher rank. Analysing the original complexes guided us in decreasing both $SD_{threshold}$ and D_{max}, thus improving runtime. In the next step we turned to raising the accuracy of the results by tuning the method's IC_{min} and k parameters which affect the quality. Our indication for quality of the result was the relative ranking of the known interface, which we generally tried to increase (see Discussion). We than used selected complexes to redefine $SD_{threshold}$ tighter, so to ensure lower false positive rate.

Using the optimized parameters the method yielded 20 out of the 26 complexes in the results. The most impressive demonstration of our method's predictive power was PDB ID 1KFU [**Fig 1**], ranking the inter-protein interface first out of 297 proposed candidates. When the known interface is tightly based on SSE, our method marks it with good score - SD= 71.4, TIC = 51. **Table 1** Holds the details of all 26 analysed complexes. The complex where the known interface was still recognised, but scored the lowest was PDB ID 1CNZ. the inter-protein result was ranked 3204 out of 3411 candidate. Though this ranking might at first look unattractive, considering that the protein complex had (36 choose 6) * (10 triplet combinations) = 1.9E7

different options for triplets pairs. Furthermore should be noted we have decreased $SD_{threshold}$ to achieve higher confidence in the results, and yet thousands of candidate PI were detected with scores lower than those of known interfaces.

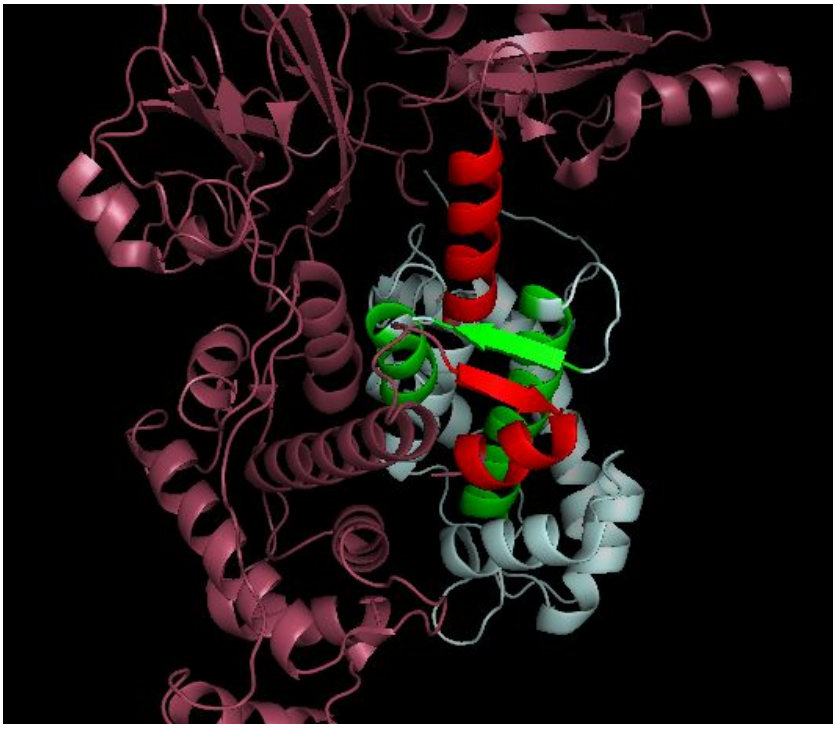


Figure 1. *The top ranking pseudo-interface in human m-Caplain heterodimer protein (PDB ID 1KFU) is the true interface between the two proteins. The large chain (L) is colored in pink, with its SSEs comprising the interface marked in red and the small chain (S) is colored in gray, with its SSEs comprising the interface marked in green.*

Table 1. *Validation using known structures of protein complexes. For each protein complex, we display the number of SSEs, the number of pseudo-interface candidates found by our method, along with total number of possible triplets combinations ((Num. of SSEs choose 6) * (10 triplet combinations)). To verify that our method manages to capture the inter-protein interface, we show the rank of the first pseudo-interface that lies between two proteins, along with its score function values.*

PDB ID	Num. of SSEs	Number of PI candidates / Number of possible triplet combinations	Rank of inter-protein interface	Scores of inter-protein interface : SD / TIC
1A0F	20	259 / 387,599	78	87.1 / 47
1A4I	32	178 / 9,061,919	22	83.8 / 40
1A4U	30	887 / 5,937,749	80	81.5 / 46
1AFW	39	15,747 / 32,626,229	59	65.4 / 35
1AJ8	40	1452 / 38,383,800	17	74.0 / 55
1AQ6	32	231 / 9,061,919	22	84.3 / 58
1B5E	22	935 / 746,130	56	74.4 / 21
1B8J	44	3857 / 70,590,519	310	79.3 / 22
1BW0	43	1283 / 60,964,540	Not Found	-
1CMB	8	3 / 280	1	70.3 / 98
1CNZ	46	3,411 / 93,668,189	3204	94.5 / 67
1DXT	13	74 / 17,159	44	90.8 / 45
1E9Z	38	547 / 27,606,810	Not Found	-
1EFV	38	1012 / 27,606,810	149	81.7 / 43
1EG9	35	1528 / 16,231,599	Not Found	-
1F3U_AB	9	10 / 839	Not Found	-
1FCD	24	24 / 1,345,960	Not Found	-
1GO3	14	42 / 30,030	7	78.3 / 47
1JKM	34	2782 / 13,449,039	75	72.9 / 55
1KFU	37	297 / 23,247,839	1	71.4 / 51
1LUC	36	984 / 19,477,920	294	86.5 / 53
1QI9	30	339 / 5,937,749	Not Found	-
1QU7	11	40 / 4620	8	78.6 / 112
1VSJ	20	258 / 387,599	3	71.3 / 98
2NAC	44	619 / 70,590,519	577	93.5 / 46
4MDH	34	895 / 13,449,039	793	93.2 / 43

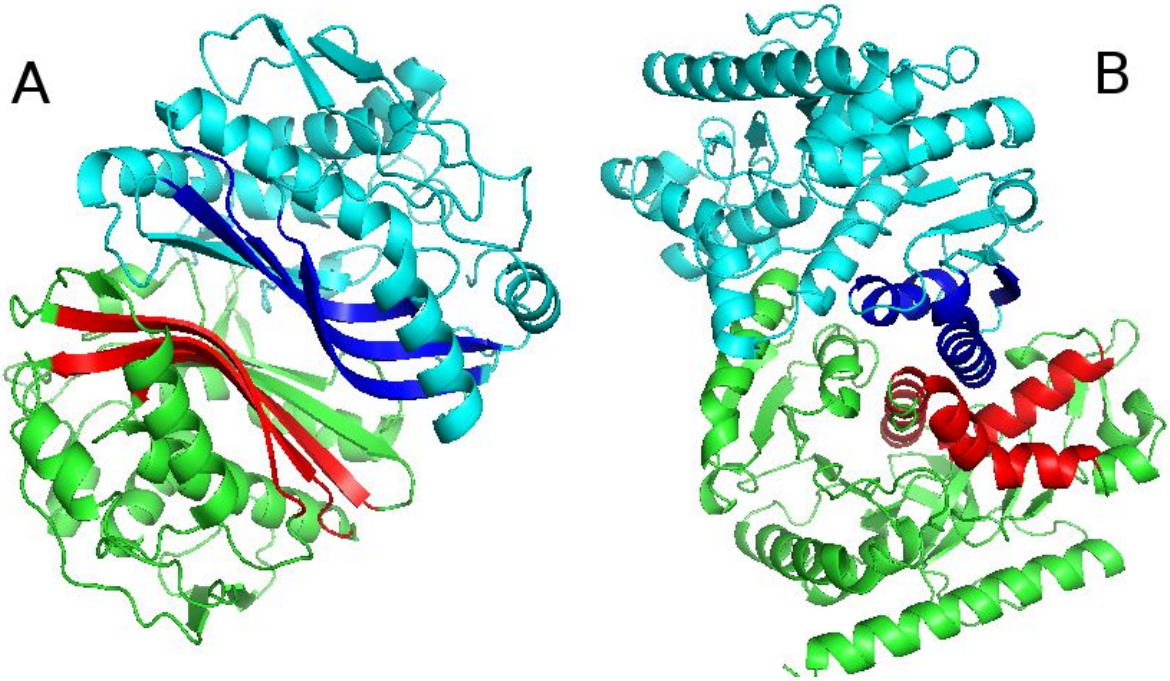


Figure 2. *Capturing interfaces of known homodimer protein complexes as pseudo-interfaces, where each chain is marked with cyan and green, and the sides of the pseudo-interfaces are highlighted in red and blue. A) Structure of thymidylate synthase (2TSC). B) Structure of malate dehydrogenase (5MDH).*

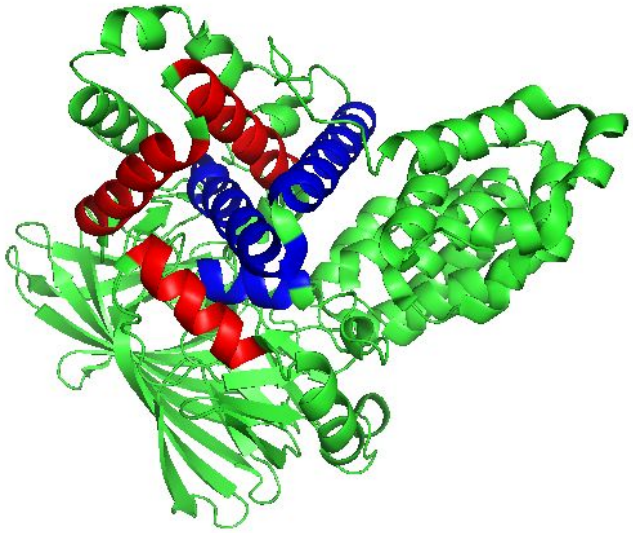


Figure 3. *Example of a pseudo-interface within a monomer protein (Leukotriene A4 Hydrolase, 3B7R). The sides of the pseudo-interfaces are marked in red and blue.*

Discussion

In this work we have presented a graph algorithm based approach for finding potential pseudo-interface in the cores of proteins. These pseudo-interfaces can be further applied to extend the breadth of search in template based computational predictions of protein-protein interfaces such as PRISM and PrePPI [Ogmen 2005, Zhang 2012]. Unlike many algorithms that rely on or exploit physicochemical properties of the elements to recognise interfaces, our strategy for the interface finding problem is based on the unsophisticated sum of distances between SSEs in two sides of the interface. The basal assumption is that in interfaces all of the elements are adjacent in space, thus our approach merely searches this feature. We reckon that for this reason many false positive results are identified as potential PIs. Nonetheless, this quality of our strategy allows swift reduction of the potential PI space into a subset enriched in true positive results [**Table 1**]. The reduced space can then be ranked by applying more timely, complex interface identification algorithms.

The $SD_{threshold}$ was chosen strictly to decrease the amount of result, but keep most of the true interfaces of the obligatory protein complexes in the results. The 6 proteins with unfound inter-protein candidate for PI might be a direct result of that decreasion, done to decrease false positive discovery. However, the term false positive we used to apprehend our buffle from the many results is deceiving - the answer to the question whether an interface truly exists or not in the wild, might only be answered in many more years of exploration. Nonetheless, it is also possible that in the visual scan we have recognised them as SSE based, but in reality the SSE interaction doesn't play a strong role in the interface.

Some clearly wanted steps for further research it would be negligent not to mention. Empirical analysis of the SSE interactions in known interfaces might stem improvements to the unsophisticated sum of distances and interaction count functions we have used. Our method relies on the assumption that each side of the pseudo-interface must have at least *k* SSEs, where in our implementation k was equal 3. We are aware many interfaces exists such that one side (or both) has four, two or even less SSE. Our algorithm is readily made for finding interfaces in which one side has 3 SSEs while the other has four, two or even one SSE, but it seems (unpublished results) that changing the value of k could introduce many false-positives, and fully rescale the scores, which will require case specific optimizations.

The ranking and ordering of the results is a full field of investigation. In this work we used the same naive sum of distances function to rank our results, but many scoring directions can be contrived which will reflect finer features of the pseudo-interfaces. For example, while the interaction count addresses residues uniformly, another option is to differentially treat them by their chemical and physical attributes. Scoring schemes taking into account spatial qualities can address the directionality one side of the pseudo-interface has in regards the the other side - we expect the majority of interfaces to be less intertwined.

In retrieving the results of potential PIs from a group of proteins, or even a single protein, we get PIs that are very similar to each other. Such cases also sometimes arise when we get different views of the same true interface. Say an interface has one side of four beta sheets, and the other has three helices. It might be that we can choose different trios out of the beta sheets, and all of them are fine candidates for PI. We believe this could be a latter stage of clustering the results, such that the final output of the method would be a representative template for each cluster.

Last, a major issue arises when optimising the results of the algorithm in respect to a specific subset of known interfaces appearing in complexes. In our experiments we tried to improve the ranking of known interfaces in complexes in respect to the PIs recognised in the complexes cores. We expected that pseudo interfaces get inferior results in respect to the resolved known interface in the complex. Doing this, we had an underlying assumption that the interface space is small, and relatively well characterised. Recent study has suggested that a clustered interface space has roughly 1000 distinct interface types [Gao 2010]. But is it so? the human genome encodes for >20,000 proteins, creating >400,000,000 potential simple protein-protein interaction. Most protein cores we have examined have given rise to thousands of potential pseudo-interfaces. Could it be that in the process we have touched the vastness of the unexplored terrain of interfaces space?

References

1. Andrusier, N., Nussinov, R., & Wolfson, H. J. (2007). FireDock: fast interaction refinement in molecular docking. *Proteins: Structure, Function, and Bioinformatics*, *69*(1), 139-159.
2. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., ... & Bourne, P. E. (2000). The protein data bank. *Nucleic acids research*, *28*(1), 235-242.
3. Duhovny, D., Nussinov, R., & Wolfson, H. J. (2002). Efficient unbound docking of rigid molecules. In *Algorithms in bioinformatics* (pp. 185-200). Springer Berlin Heidelberg.
4. Gao, M., & Skolnick, J. (2010). Structural space of protein–protein interfaces is degenerate, close to complete, and highly connected. *Proceedings of the National Academy of Sciences*, *107*(52), 22517-22522.
5. Halperin, I., Ma, B., Wolfson, H., & Nussinov, R. (2002). Principles of docking: An overview of search algorithms and a guide to scoring functions. *Proteins: Structure, Function, and Bioinformatics*, *47*(4), 409-443.
6. Hamelryck, T., & Manderick, B. (2003). PDB file parser and structure class implemented in Python. *Bioinformatics*, *19*(17), 2308-2310.
7. Hatfield, J. W., & Kominers, S. D. (2012). Contract design and stability in many-to-many matching. *Harvard Business School, Mineo*.
8. Jiang, S., Tovchigrechko, A., & Vakser, I. A. (2003). The role of geometric complementarity in secondary structure packing: a systematic docking study. *Protein science*, *12*(8), 1646-1651.
9. Kabsch, W., & Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*,*22*(12), 2577-2637.
10. Katchalski-Katzir, E., Shariv, I., Eisenstein, M., Friesem, A. A., Aflalo, C., & Vakser, I. A. (1992). Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. *Proceedings of the National Academy of Sciences*, *89*(6), 2195-2199.
11. Keskin, O., Nussinov, R., & Gursoy, A. (2008). PRISM: protein-protein interaction prediction by structural matching. In *Functional Proteomics* (pp. 505-521). Humana Press.
12. Ogmen, U., Keskin, O., Aytuna, A. S., Nussinov, R., & Gursoy, A. (2005). PRISM: protein interactions by structural matching. *Nucleic acids research*, *33*(suppl 2), W331-W336.
13. Soner, S., Ozbek, P., Garzon, J. I., Ben-Tal, N., & Haliloglu, T. (2015). DynaFace: Discrimination between Obligatory and Non-obligatory Protein-Protein Interactions Based on the Complex's Dynamics. *PLOS Comput Biol*,*11*(10), e1004461.
14. Tsai, C. J., Lin, S. L., Wolfson, H. J., & Nussinov, R. (1996). 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*, *31*(2), 127-152.
15. Zhang, Q. C., Petrey, D., Deng, L., Qiang, L., Shi, Y., Thu, C. A., ... & Honig, B. (2012). Structure-based prediction of protein-protein interactions on a genome-wide scale. *Nature*, *490*(7421), 556-560.

Appendix A - Description of the pseudo-interface finder

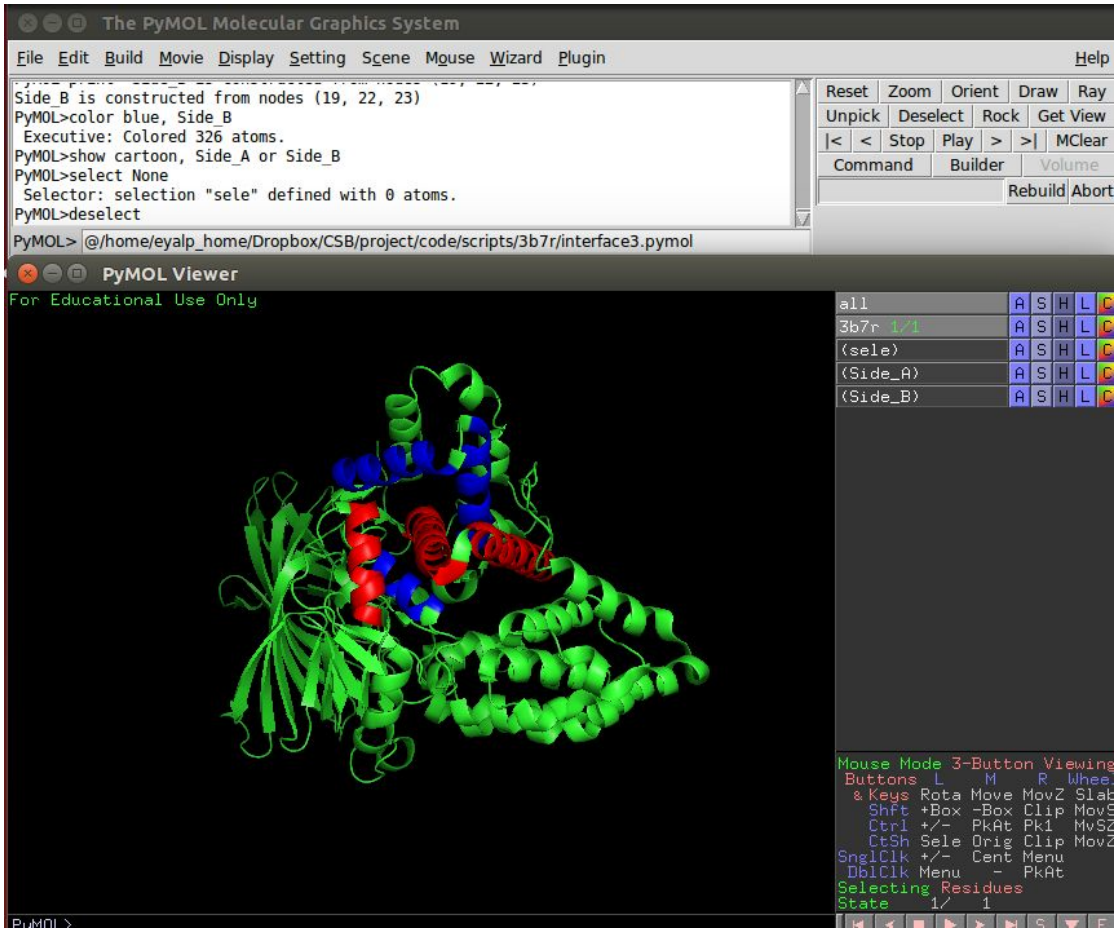
Attached to this document is the full code of the pseudo-interface finder tool which implements the method described in this work. The code requires the system to have Python 2.7, with Biopython library and the DSSP executable installed in the shell execution path.

To use the tool, you must have a the PDB file of the query protein stored in the local file system. The command line is:

```
python pinterface_finder.py [--sd 94.0] [--core] path_to_pdb num_of_results
```

- *path_to_pdb* should point to the relative path of the PDB file
- *num_of_results* determines the amount of templates the tool should output
- *sd* argument allows to manually configure the $SD_{threshold}$ parameter described above.
- For PDB files containing more than one chain, the optional argument *core* causes the tool to output only pseudo-interfaces appearing in a single chain - hence ignoring pseudo-interfaces containing SSEs of different chains.

The tool outputs the top ranking pseudo-interfaces recognized in the PDB file. The results are saved into two directories in the execution path. The first directory, **templates**, contains PDB files of the SSEs comprising each pseudo-interface, such that one side of the interface is under chain X and the other under chain Y. The second directory, **scripts**, contains PyMOL console scripts that outmarks the given pseudo-interface of the query protein. To execute a PyMOL console script, first load the query protein, then write in the console: "@/**[full_path_to_scripts]**/interface**[number]**.pymol". The pseudo-interface's SSEs will be labeled Side_A and Side_B, and colored in red and blue accordingly.


The PyMOL Molecular Graphics System
File Edit Build Movie Display Setting Scene Mouse Wizard Plugin Help
Side_B is constructed from nodes (19, 22, 23)
PyMOL>color blue, Side_B
Executive: Colored 326 atoms.
PyMOL>show cartoon, Side_A or Side_B
PyMOL>select None
Selector: selection "sele" defined with 0 atoms.
PyMOL>deselect
Reset Zoom Orient Draw Ray
Unpick Deselect Rock Get View
|< < Stop Play > >| MClear
Command Builder Volume
Rebuild Abort
PyMOL> @/home/eyalp_home/Dropbox/CSB/project/code/scripts/3b7r/interface3.pymol
PyMOL Viewer
For Educational Use Only
all
3b7r 1/1
(sele)
(Side_A)
(Side_B)
Mouse Mode 3-Button Viewing
Selecting Residues
State 1/ 1