PDIB

María Torralvo, Ángel Monsalve and Mireia Codina

Protein-DNA Interaction Builder (PDIB) Application

PDIB is a Python application that builds biological macrocomplexes from PDB files containing pairwise interactions between polypeptides chains or between protein and nucleotide sequences (DNA or RNA). Knowing how such structures assemble between them is crucial to understand the function of proteins and gain a deeper insight into their activity and subcellular or peripheral location.

PDIB has been developed as the final project for both Introduction to Python and Structural Bioinformatics subjects of the MSc in Bioinformatics for Health Sciences at UPF (Universitat Pompeu Fabra) by María Torralvo, Ángel Monsalve and Mireia Codina.

How does PDIB build macrocomplexes?

1. Input processing

The program starts by preprocessing all the PDB files that are stored in the selected input directory. It is possible that the same chain is coded with different IDs in the files. To identify unique chains, the program aligns the sequences and determines if they can be considered the same or not, with an initial threshold of 95% of identity in the alignment, that is updated in case a higher one is found. Above this value, chains are estimated to be equal and assigned the same random unique identifier. Both original and unique identifiers are kept, since we need the original to get the stoichiometry of the final complex.

2. Superimposition

The approach that we selected for building the macrocomplexes is based on superimposition of partial structures. If a chain is found in two pairwise interaction files, they are superimposed in such a manner that one is fixed and the other moves to obtain the rotation matrix, which is used afterwards to allocate the other chain from the non-fixed structure with respect to the fixed one. Only alpha-carbon atoms ('CA') for protein residues and backbone phosphate ('P') for nucleic acids were used to compute the rotation matrix.

RMSD calculations are essential for the performance of PDIB since it is a measure of conformational stability of a complex during its simulation. It allows the algorithm to determine whether two chains that are being superimposed are in fact homologs. This is crucial, as the program needs to superimpose homolog chains and apply the rotation matrix to the other chain of the pairwise alignment. Doing it the other way will result in a poor performance of the program. Another situation where RMSD needs to be computed is when the superimposition displaces the moving chain to a position where there is already an homolog of the moving chain. In both cases, to consider two chains as being homologs, we defined a threshold for the RMSD of 3Å. According to the results obtained in the study made by Li et al. (2010), the RMSD for backbone atoms was around 2.4 Amstrongs. Therefore, chains with backbone RMSD values above 3 Amstrongs after the superimposition are treated as different chains. The RMSD is calculated between the alpha carbons of both chains, taking into account only the backbone.

3. Recursive approach

In the first step of the algorithm, a random pairwise interaction is chosen and only those chains that interact with them are added. This could be enough for smaller complexes like a tetramer, where most likely all of the chains will interact with at least one of the two chains in the starting structure. However, a recursive approach is needed to add all the chains to larger complexes. Therefore, in deeper levels of recursion, the program will try to add all possible chains to those included in the previous level of the model, ending up with the complete structure.

4. Clash detection

Of all the possible chains that can be added to the developing structure, not all of them result in reliable models when added. It may happen that, when adding a new chain, the superimposition is successful but the chain presents clashes with other parts of the complex.

We implemented a simple approach to detect these events. Using Bio.PDB.NeighborSearch module from Biopython, we are able to determine how many atoms of the tested chain are located closer than 2.5 Angstrom, which is the minimum distance possible between two Van der Waals radii of two atoms interacting through a hydrogen bond (Tsai et al., 1999), hence identifying them as clashing atoms. Nevertheless, a small number of clashes could be due to slight errors when applying the rotation matrix to the moving chain and could be fixed with optimization. We have selected a threshold of 20 atoms to consider if a clash is real.

5. Exploring more than one model

In the previous section, we have described what we consider to be a clash between chains. On the one hand, it is clear that this could happen when we are trying to add a part of a subunit that was already included. However, it could also mean that an alternative complex could be build. Biologically, this could indicate that the macrocomplex might have a changing structure between conditions.

The algorithm implemented studies each clash between the chain that is going to be added and the structure at that point and, if it is not clashing because we are trying to add the same chain again, an alternative structure is created. This event is called “branch opening”. The clashing chain or subunit is removed from the structure and the tested chain is added in its place without losing the structure that was being assessed thanks to the recursive approach. Besides, the information of these clashing chains is stored so that, if the clash is tested again, there will not be a new branch opening event, since it would be redundant.

Since this process could end up getting the same model using different paths, when a structure is considered finished, i.e. no more chains can be added without clashes, it is compared to those previously saved performing a superimposition of complete structures. Same RMSD criteria used to determine if two chains were the same is applied for the structures.

6. Model scoring

By using ProSa2003, the program allows the user to compute the Z-Scores of the generated models. These scores are calculated by summing the energies (both surface, paired and combined) for all the residues of a model and comparing it to the energies of several models with different folds.

This generates a file with all the models sorted by their z-scores. A low Z-Score usually indicates that the model is good and stable, while higher or even positive Z-Scores account for incorrect models with high pseudoenergies.

7. Model refinement

The program makes use of PyRosetta to refine the models, allowing for some flexibility. PyRosetta is an interactive Python-based interface to the Rosetta molecular modeling suite.

PDIB allows the user to choose between different relaxing methods, such as the classic relax protocol or the fast relax protocol. These relaxing methods allow for all-atom refinement searching the local conformational space around the starting structure. This refinement allows the program to yield better models with lower energies.

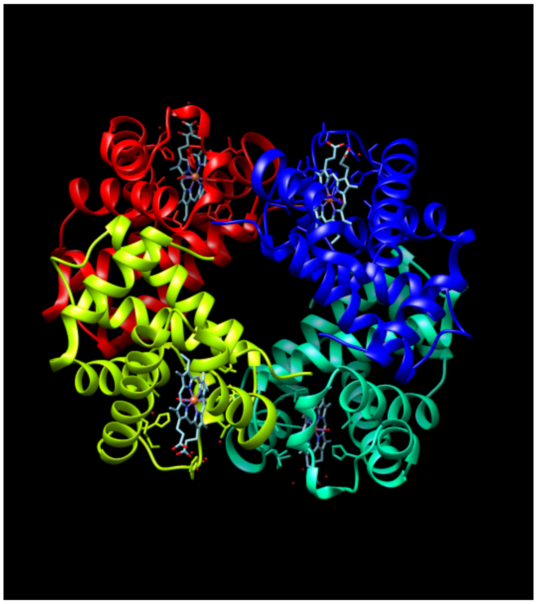
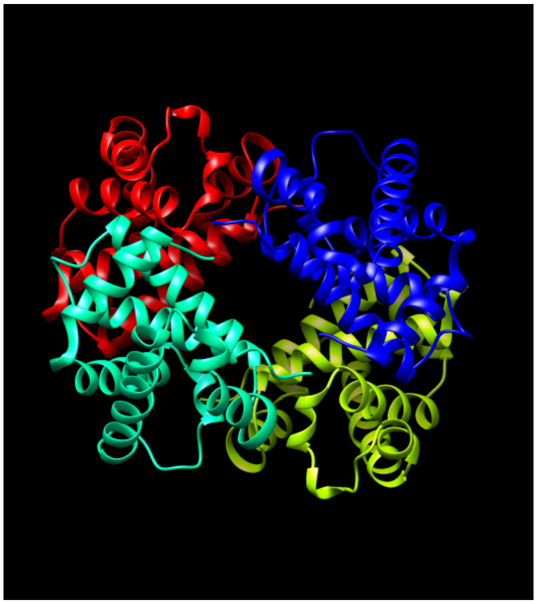
ANALYSIS OF THE PERFORMANCE OF PDIB

In order to prove the suitability of the program for modelling protein macrocomplexes, we ran the program to try to reconstruct some known proteins. The obtained models were superimposed with the original protein in order to examine the difference between them, given by the RMSD of the superimposition. We ran the program for four different PDB entries: 1GZX, 5FJ8, 2O61 and 5NSS. All the examples are provided in the folder examples/ so the user can run them. The pairwise interaction files of 2O61 and 5NSS were generated by splitting the model retrieved from PDB into pairs of chains.

Examples

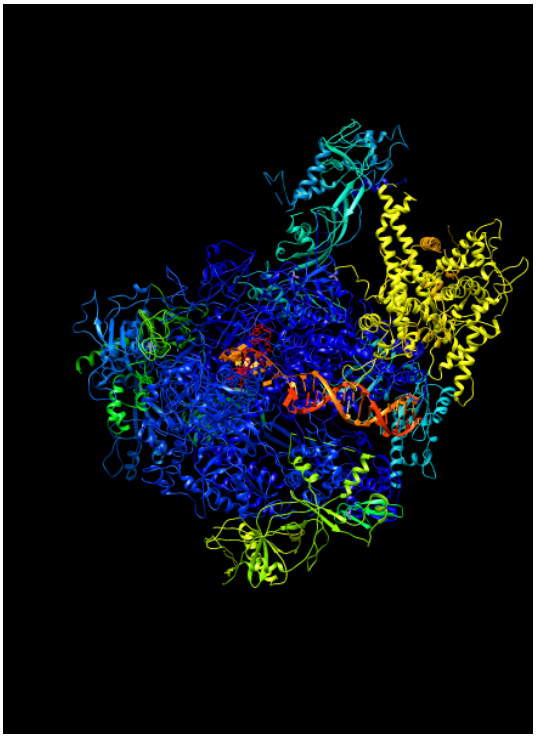
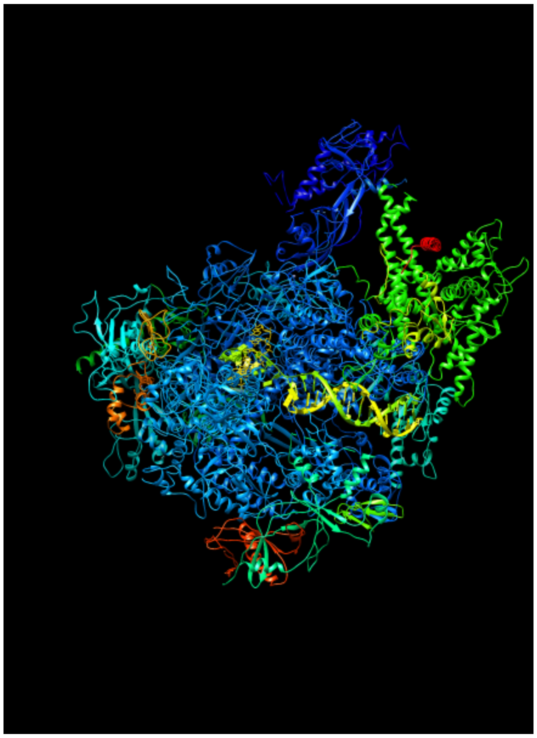
1GZX (oxy T state haemoglobin: oxygen bound at all four haems)

1GZX shows the molecular structure of human hemoglobin in the T-state, with oxigen attached to its four heme groups. The protein is made of two different subunits: α and β, and both are repeated two times, making an heterotetramer. Here PDIB achieved a perfect result, with an RMSD of 0 Angstroms after the superposition of the model with the original structura. Despite achieving a good representation of the protein chains, this model shows one of the major limitations of PDIB, as it is not able to handle the prosthetic groups, which are the protoporphyrins of the heme groups.

Model	Original
	

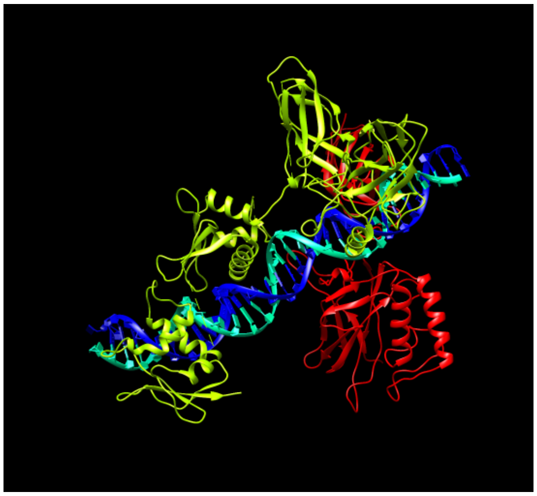
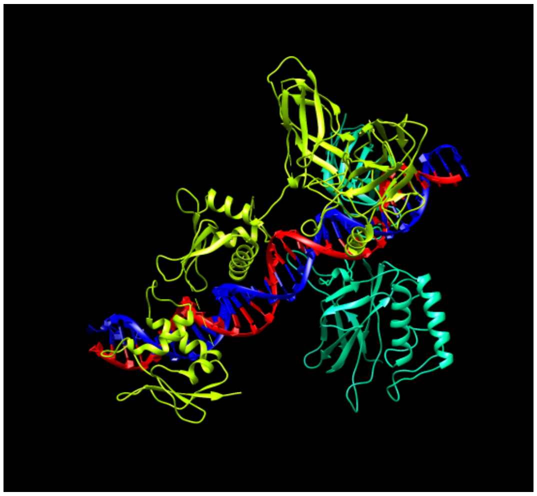
5FJ8 (Cryo-EM structure of yeast RNA polymerase III elongation complex at 3. 9 A)

This protein shows an RNA polymerase from yeast. It is made of 17 unique protein chains and 3 unique nucleic acid chains. Each of the 17 protein subunits is repeated one time. The 3 nucleic acid chains are an RNA, a non-template DNA and a template DNA. Each DNA chain represents one strand of a double helix, while the RNA is a small sequence. PDIB was very successful in reconstructing this macrocomplex, yielding a good RMSD after the superimposition. However, as no stoichiometry was provided, the model lacks the chain Q. which is a small helix shown in blue at the bottom of the image of the original protein. However, the program could model both the proteins, the DNA and the RNA in a very short time.

Model	Original
	

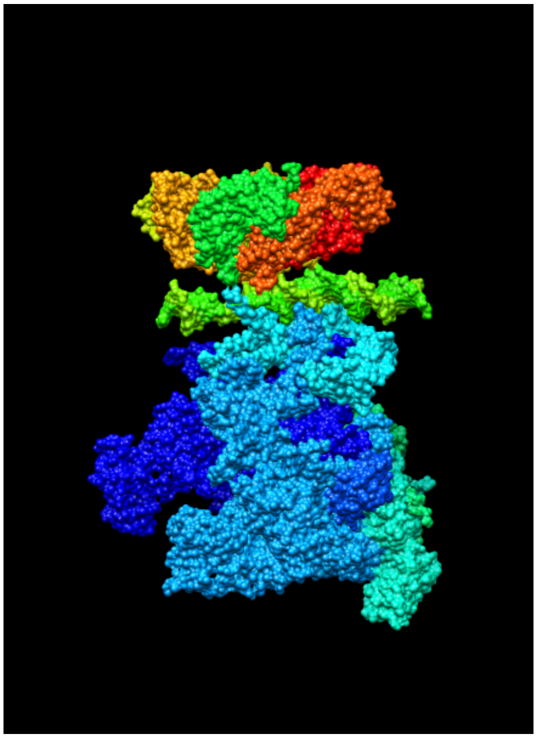
2O61 (Crystal Structure of NFkB, IRF7, IRF3 bound to the interferon-b enhancer)

This example shows a complex made of 2 unique protein chains, an NF-$\kappa\beta$ nuclear factor and a transcription factor p65. Both are interacting with a double-stranded DNA sequence. In this case, the model also performed very well, being able to perfectly reconstruct the complex, as shown in the images below.

Model	Original
	

5NSS (Cryo-EM structure of RNA polymerase-sigma54 holoenzyme with promoter DNA and transcription activator PspF intermediate complex)

5NSS is a structure obtained by cryogenic electron microscopy of a bacterial RNA polimerase that contains the alternative transcription factor σ^{54}. In a very short time, PDIB was capable of elucidating the structure of this complex. The complete structure contains 6 different protein chains and 2 nucleic acid chains. Amongst the protein chains there are two α subunits (chains A and B), a β subunit (chain C), a β' subunit (chain D), an ω subunit (chain E), six Psp operon transcriptional activator (chains F, G, J, K, L and N) and and RNA polymerase σ^{54} factor (chain M). Amongst the nucleic acids we find two promoter DNAs (chains H and I), each one representing one DNA strand. Despite the complexity of the model, as it contains 14 different chains, the program achieved a perfect result, with an RMSD of 0 Angstroms when superimposed to the original model. Also, the runtime was very short, demonstrating the convenience of the recursive algorithm.

Model	Original
	

Limitations

One of the main limitations of PDIB is that it is an heuristic solution, since it works with our examples, however there is no guarantee that it will work with other real cases. Considering that all the models we have created were done using files obtained dividing complete PDB structures into pairwise interactions, we are afraid of how it will perform with other types of data. However, since we have followed an exhaustive approach, where all possibilities are tested, we are confident that it could perform well, especially when the stoichiometry is provided.

The downside of this method is that, although a priori must be really good at finding all possible models, it implies a high computational cost. In fact, for bigger complexes if all models were requested, the computer would even kill the process before finishing. This is even more noticeable when the refine option is enabled, as PyRosetta takes a very long time itself to apply the refinement, even using a fast relax protocol.

Finally, as mentioned before, another major drawback of the program is its inability to handle prosthetic groups. The presence of these molecules would not stop the pipeline, but the final models will not contain them. Sometimes these groups are an important part of the model, as the user needs to know how they interact with the rest of the chains and they are key when assessing the function of a structure, and thus not being able to include them is one of the major disadvantages of PDIB.

References

Li, M.-H., Luo, Q., Xue, X.-G., & Li, Z.-S. (2010). Molecular dynamics studies of the 3D structure and planar ligand binding of a quadruplex dimer. Journal of Molecular Modeling, 17(3), 515–526. https://doi.org/10.1007/s00894-010-0746-0

Tsai J, Taylor R, Chothia C, Gerstein M. (1999). The packing density in proteins: standard radii and volumes. JMol Biol., 290(1), 253-66. https://doi.org/10.1006/jmbi.1999.2829