ComplexArchitect - Report

Oscar Camacho

Introduction

Proteins are macromolecules consisting of amino acid residues that perform a vast array of biological functions within organisms, including catalyzing metabolic reactions, DNA replication, responding to stimuli, transporting molecules or providing structure to cells. The activity of the proteins is determined by its specific three-dimensional structural, which depends on its protein folding, which in turn depends on its sequence of amino acids, dictated by the nucleotide sequence of their genes.

A single chain of amino acids is called polypeptide. In many cases, in order to achieve a particular function several polypeptide chains (protein subunits) associate to form stable protein complexes. Those structures are called protein quaternary structures and consist of multiple protein molecules in a multi-subunit complex that interact in different ways, including hydrogen bonds, electrostatic forces and the hydrophobic effect. Furthermore, proteins can interact with other biomolecules like DNA and RNA, forming macrocomplexes that develop functions related to transcription and translation processes.

Nevertheless, understanding how proteins interact and determining the full structure of a macrocomplex is not an easy task. The Protein Data Bank (PDB) is a database that consists of a large set of protein structures experimentally determined by X-ray crystallography or nuclear magnetic resonance (NMR). The reality is that there are many cases in which the experimental determination of the structure of macrocomplex is very costly. Thus, the development of bioinformatics techniques that combines the experimental data with computational resources is essential to speed up the process.

ComplexArchitect is a stand-alone application that constructs protein macrocomplexes from subunit pairwise interactions, taking advantage of the already available information that PDB provides about the molecules that conform the complex. The program reads a set of PDB files containing each one a pair of subunits that are known to interact and builds different multi-subunit complexes.

Theoretical background

Macrocomplexes follow a specific spatial conformation of its subunits, taking into account the contacts or interactions between them. The main approach that ComplexArchitect follows in order to put the subunit interacting chain pairs in the correct spatial position is the structural superimposition. Protein superimposition is the procedure by which two protein structures are placed in space minimizing the distance between the backbone atoms of both structures. This implies that one of the two structures is rotated and translated in space, so that fits as good as possible the coordinated of the other structure. Furthermore, we have to take into account that in macrocomplexes there can be chains that interact with more than one chain.

As our input are pairs of interacting chains of the macrocomplex, the process of building the whole structure would start by taking one of those pairs as a first template and then superimpose the common chains of the pairs of interactions by structural superimposition. This way we can

superpose the identical chains and move the new pair interaction to the template. The process should be repeated until all the similar chains are superposed, finally obtaining the macrocomplex.

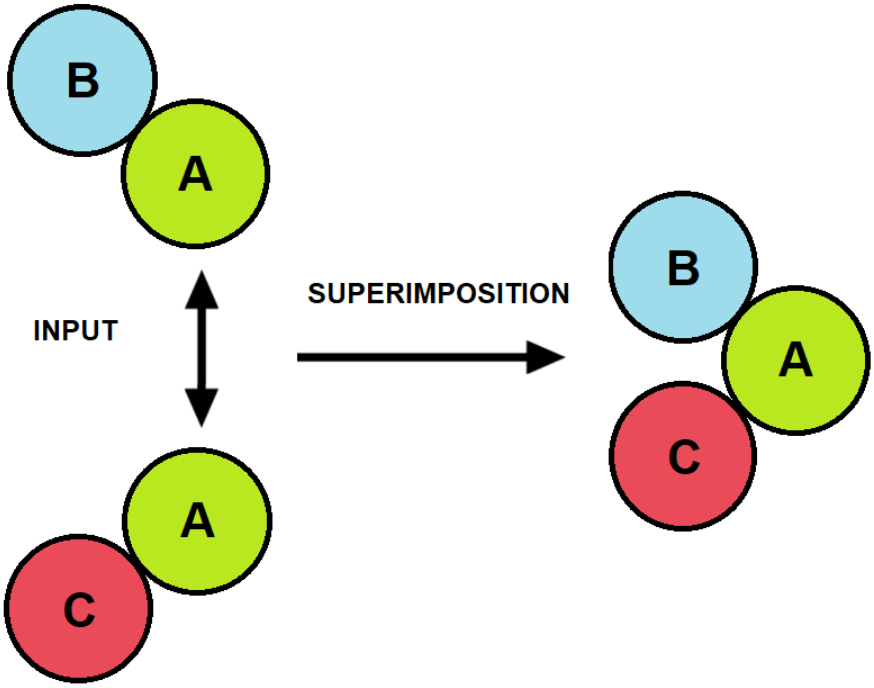

Figure 1. Example of principle of superposition. Considering two pairs of interacting chains as input (A-B) (A-C), chain A is repeated in both pairs. Actually, chain A is both interacting with B and C. Thus, chain A can be superimposed in order to build a complex of 3 different chains.

Before starting to superimpose pairs of interactions, though, we have to identify what chains are identical. This can be done by performing pairwise sequence alignments between all the chains that are provided as the input. Chains with a high sequence alignment score will be considered identical and, therefore, will be the chains that will superimpose.

Furthermore, every time a new pair of interacting chains is added to the complex it must be checked that steric clashes don't occur. Steric clashes arise due to the unnatural overlap of any two non-bonding in a protein structure. We have to take into account that the chain that interacts and accompanies the chain that superimposes could provoke clashes with other parts of the whole structure. In order to avoid the clashes, it is very important the order in which the interacting pairs are added to the complex. Soichiometry or total number of chains must be taken into consideration to build a specific model.

Finally, evaluating the resulting models is critical. Discrete Optimized Protein Energy can be used to assess the quality of the model. It is important to consider the final energy levels of the complex. A good model should have a low energy. Naturally occurring complexes are the ones with the lowest energy among the set of possible foldings. Situations such as two atoms very close to each other, amino acids with hydrophobic residues located in the external part of the macromolecule and several other cases can increase the final energy of the complex, which should be then corrected.

Below it is explained how we implemented the algorithm of our program taking into consideration the theoretical background.

Algorithm

- **Identification of common chains**

ComplexArchitect, first, takes as input all the files that are in pdb format and stores the interacting pairs of chains. One of the strong points of the application is that many recurrent actions are constructed as methods of customized objects that correspond to the different chains and the models that participate in the construction of the macrocomplex.

A FASTA file with the sequences of the unique chains of the complex is optional, but we encourage to use it. With the FASTA file, all the sequences of the PDB chains of the interacting pairs are aligned, one by one, with the different sequences of the file. If the chain gets an identity higher than 95% in the pairwise alignment, it gets the ID of the FASTA sequence. Therefore, at the end all the chains will be correctly identified and those that are equal will have the same ID. We consider that those chains can be potentially superimposed, as we assume that proteins with similar sequence will have a similar fold. As we already know how chains will be identified since the beginning of the program, stoichometry can be added as an argument before starting to run the application.

However, the program can run without providing the FASTA file. In that case, unique (non-repited) chains will be identified and will be showed to the user. Then, the user, knowing which chain corresponds to which ID, will be able to provide the stoichiometry if is desired.

- **Superimposition**

The superimposition starts with the model with the chains with more interactions, which corresponds to the chains that have been aligned overcoming the threshold score more times. From here, chains that can potentially be added are searched. An interaction dictionary with keys as the IDs of the unique chains and tuples with the information of the possible chains that can be superposed as values is used. In the process of superimposition, the program finds the common atoms among the chains that are going to superpose and the candidate chain to be added changes its orientation using a rotation matrix in order to minimize the distance between the atoms of both chains. This way, the program places the atoms of both chains in the same coordinates and orientation.

- **Steric clashes**

Before adding new chains to the complex, steric clashes are analized with the module NeighborSearch. This way we can find how many atoms of the moving chain are located closer than a threshold distance to those in the current structure of the complex. We set the threshold distance in 2 angstroms as default although the user can change it. Alpha carbons are used as the backbone atoms that are compared between the chain and the other parts of the macrocomplex in construction. If 3% of those atoms show clashes, the chain is not added. On the contrary, the program will try to fit those chains in the complex in further steps.

The process of building the complex by superimposition will be repeated as many times as necessary until all the different input chains are placed in the 3D space.

In terms of the algorithm, when a chain succesfully superposes with another in the model and, thus, the interacting chain is added, that pair of interactions is removed from the dictionary, since it is already accomplished. If clashes are found, the pair of interactions is not removed from the dictionary, waiting for future opportunities to be added on other places.

- **Stoichiometry**

If stoichiometry is provided, the program limits the number of times a given chain is added in the macrocomplex. This is done comparing in each iteration the stoichiometry of the current complex with the stoichiometry specified by the user. If a given chain reaches the maximum, that chain is not added. If the stoichiometry is fulfilled, the program stops running returning the resulting complex.

- **Resulting complexes and assessment**

ComplexArchitect generates as many complex models as the user decides. One important aspect to keep in mind is that if stoichiometry is not provided, the resulting complexes can be very diverse, since we add a random behavior regarding the order in which chains are searched and added. This is more accentuated when chains are repeated a large number of times. In those cases, some resulting models can have 20 chains A and other ones 12 chains A, for example. With lower number of repetitions, models tend to automatically be built with the stoichiometry of the native complex.

Once a model is generated, it is important to evaluate the structure. The application includes a basic MODELLER optimization with restraints. Moreover, we can obtain a DOPE profile comparing the non-optimized and the optimized models in order to check if the model improved after refinement. An energy profile give us a quick overview of the quality of our model. DOPE, or Discrete Optimized Protein Energy, is a statistical potential used to assess models in protein structure prediction. Areas with a DOPE score greater than zero indicate poor quality. With the DOPE score we can select the best structure from a collection of resulting models. The program returns the Z-score, with is the normalized version of the DOPE score.

Limitations

- A big limitation of the program is that although a given provided stoichiometry limits the maximum number of chains that are going to be added, it does not ensure that all the chains are added. This can be a problem with complexes with many chains and large numbers of repetitions. Fortunately, although not always, the model tends to follow the specified stoichiometry. The fact that the program can build more than one complex also allows the user to decide which model is the one is interested in.

- Once an interacting pair of chains is used and the corresponding chain is added to the complex, those chains are removed from the set of possible interactions. This means, the program can not build complexes with redundant interactions unless they are included more than once in the input.

- In many cases the optimization function does not change much the energy of the complex and in some examples it even increases. Since in various examples, the RMSD between the resulting complex and the native one is 0, this would mean that trying to minimize the energy of a native complex is non-viable, at least with that function. That way, some energy comparison plots show the DOPE profile of the optimized model above the unoptimized one.

Discussion and conclusion

Although computational approaches for constructing macrocomplexes are far to be perfect at the moment, a lot of effort is made to develop new methodologies. Our approach does not represent any novel strategy and shows good results when using already known structures.

In order to improve our application we would have to incorporate many types of additional data to be integrated in the models. Those types of data can include biological and experimental data that would have to be carefully analyzed, taking advantage of the computational resources, probably using machine learning. The achievement of better methodologies will definitely have an impact in many varied applied fields, such as drug discovery or biomedical research.

Bibliography

Hayes, S., Malacrida, B. , Kiely, M., Kiely, P. A. Studying protein–protein interactions: progress, pitfalls and solutions. *Biochemical Society Transactions* **44**, 994-1004. doi: 10.1042/BST20160092

Liu, S., Liu, C., Deng, L. Machine learning approaches for protein–protein interaction hot spot prediction: progress and comparative assessment. *Molecules* **23**, 2535 (2018). doi: 10.3390/molecules23102535

Keskin, O. , Tuncbag, N. , Gursoy, A. Predicting protein-protein interactions from the molecular to the proteome level. *Chemical Reviews* **116**, 4884-4909 (2016). doi: 10.1021/acs.chemrev.5b00683