Macro-Complex Modeling of Biomolecules

Angelo Rosace, Iain Maryanow, Zach Collester
Universitat Pompeu Fabra: MsC Bioinformatics

April 3, 2020

1 The application

This a python-based application that utilizes pairwise interaction data as input to generate macro-complex models of biomolecules. It functions for both protein-protein interactions (PPIs) as well as protein-DNA & protein-RNA interactions. The output is a PDB file of the generated model.

2 Biological Background

2.1 Protein Structure

Proteins are vital for the structure, function, and regulation of the body's tissues and organs. Further, protein structure plays an incredibly important role in determining its biological function, and this structure is organized in a hierarchical manner. Primary structure consists of amino acids bound together via peptide bonds to form chains known as polypeptides. The secondary structure arises from hydrogen bonds between amino acids in the polypeptide chain causing the chain to form a spiral (alpha helix) or bend and fold (beta pleated sheet). These intermediate folded structures can then fold together to form three-dimensional protein subunits, known as protein tertiary structure. Most functional proteins have quaternary structure, in which several tertiary domains assemble into multi-subunit complexes via non-covalent bonds (hydrogen bonding, Van der Waals forces, etc.).

2.2 Protein/DNA Interactions

Proteins can also bind to a molecule of DNA in order to perform various biological functions. For example, many of these functions include DNA

replication and repair, transcription, recombination, and packaging chromosomal DNA. These interactions occur predominantly via dipolar interactions (hydrogen bonding) and dispersion forces.

3 Building the Complex

3.1 Data Structure

Once the application receives the pairwise interaction data, it stores the unique data for each chain (chain label, residue order, atom order, atomic locations). This data is then used in subsequent modeling steps.

3.2 Chain Superimposition

The application builds macro-complexes via a series of protein chain superimpositions. This functions by fixing one chain in space, moving the other chain, obtaining the corresponding rotation matrix, and then utilizing this matrix to join the moved chain to the fixed chain and adding it the model. This structural superimposition method allows the application to construct complexes in a recursive manner, which will be explained later in more detail.

3.3 Clashing

In order to build complexes that are biologically plausible and obey basic physiochemical laws, it was necessary to ensure that chains did not clash before adding a new one to the model. The application considers the alpha carbon of each residue as the core atoms. Clashing is defined as two core atoms being within a certain distance of another. In the case of this application, clash distance is 2 Å and superimposed chains that have any clashing core atoms are not incorporated in the model.

3.4 Stoichiometry

This program allows the user to input a desired stoichiometry for the model. Stoichiometry refers to the composition of the model, specifically the number of each chain included in the final model. For example, if the inputted protein consists of four unique chains (A, B, C, and D), the user could specific a specific stoichiometry of 1A, 2B, 2C, and 1D, and the model that is built will have 1 A chain, 2 B chains, 2 C chains, and 1 D chain. If the stoichiometric constraints applied by the user do not result in a biologically plausible model, the program will inform the user that the stoichiometry is not valid.

3.5 Recursive Function

To build complexes with multiple subunits, the program utilizes a recursive function. During the recursive function, each chain is aligned with one another, and chains with greater than 95% sequence similarity are regarded as the same chain. The logic behind this step is that the number of protein folds that exist in nature is finite and there are a few folds that are extremely common. Therefore, chains with a high degree of sequence similarity are likely to share a very similar structure and can reasonably be recognized as the same chain. To ensure this, the program also checks for structural similarity to ensure that no errors are being committed when regarding two chains as the same.

After the first iteration of the recursive function, the model consists of two chains. During the next iteration, the addition of subsequent subunits is dependent upon satisfying the previously mentioned clashing and stoichiometric constraints. This function allows the creation of all plausible models. However, the application has been developed such that only the best model is saved into a PDB file. The best model is defined as the model having the lowest RMSD score. In statistical terms, RMSD (Root Mean Square Deviation) is a measure of the distance between predicted values from a model and actual values. In the case of macro-complex modeling, a low RMSD score is favorable.

4 Tutorial

The following section is an outline for how to use this application in the terminal.

4.1 Software Requirements

```
Python    >= 3.0
Biopython == 1.76
argparse  == 1.4.0
```

4.2 Installation Instructions

To use this program, you may either clone this Git repo and call

```
macro_molecular_complex_builder.py
```

or by installing

```
dist/macro_molecular_complex_builder-0.0.1.tar.gz
```

using the following command:

```
pip install dist/macro_molecular_complex_builder-0.0.1.tar.gz
```

4.3 Required Command-line Arguments

–input_directory : directory of chain interactions
–output_file : name and location of output file including file extension

4.4 Optional Command-line Arguments

–verbose : printing messages during execution
default = False
–stoichiometry : name and location of TSV file for stoichiometry of complex
default = None
–chain-limit : maximum number of chains in the final complex
default = 10

4.5 Example

In the directory /example, there is a list of interaction data from PDB files. These interaction files can be run through the program to generate a protein model by running the following command:

```
python3 macro_molecular_complex_builder.py --input_directory
./4g83 --output_file ./out.pdb --verbose
```

With stoichiometry:

```
python3 macro_molecular_complex_builder.py --input_directory
./4g83 --output_file ./output.pdb --stoichiometry ./4g83/stoichiometry.tsv
```

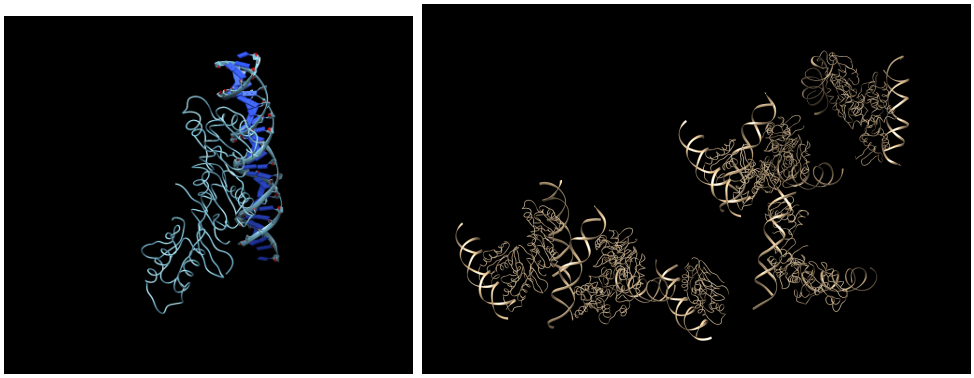
5 Analysis

For all of the following examples, the images shown on the left are the constructed models and the images on the right are they original models.

5.1 Example 1

The first example was run using 3t72, a DNA Transcription Activation Sub-Complex. The complex is a Hetero 3-mer (stoichiometry of A2B).

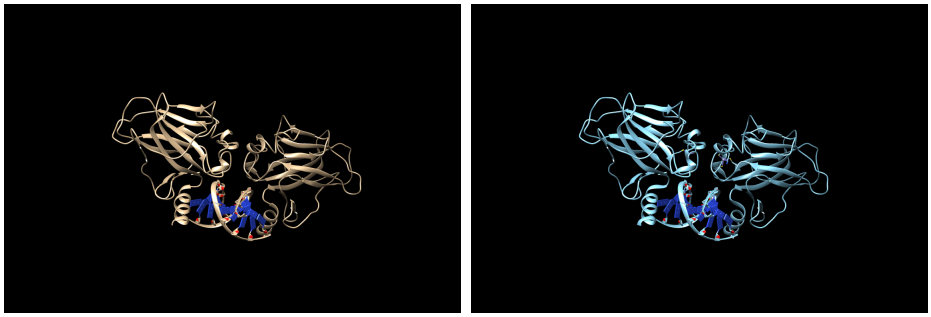
Upon running the program, it was found that the constructed model replicates just a small part of the original pdb file. This could be due to the fact that our algorithm doesn't detect a clear relationship with the other sub complexes. Although not managing to recreate the whole macro complex, the algorithm still manages to give to the user a good picture of one of its subunits.



5.2 Example 2

The second example was run using 4g83, a DNA-Binding Domain Tetramer bound to a Full Response-Element. The complex is a Homo 4-mer (stoichiometry of 4A).

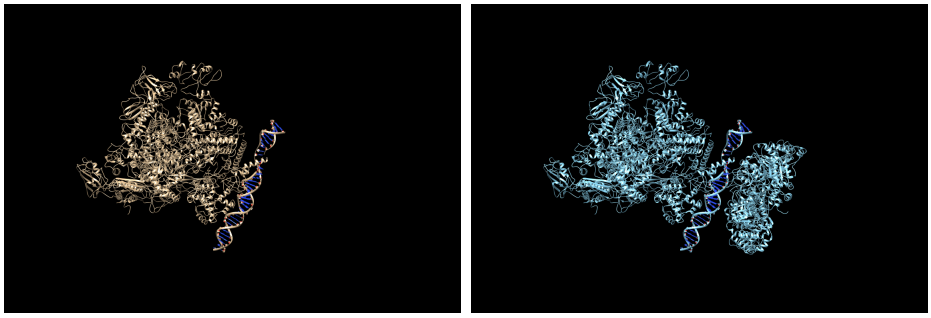
The constructed and original model are structurally the same. The only difference we see is that we are missing some small atoms. This could be due to the pdb interaction files not holding information for this atoms or due to the fact that our superposition method implies the truncation of at least one of the chains involved in the superposition.



5.3 Example 3

The third example was run using 5nss, a holoenzyme with promoter DNA and transcription activator. The complex is a Hetero 12-mer (stoichiometry of A6B2CDEF).

The constructed model and the original super-complex are really close in terms of structure. We can clearly see though that a small portion of the original model has not been modeled. This could be due to the algorithm not recognizing the interaction between the sub units of the model. This can in turn be due to the parameters set in the algorithm when checking for interactions between chains.

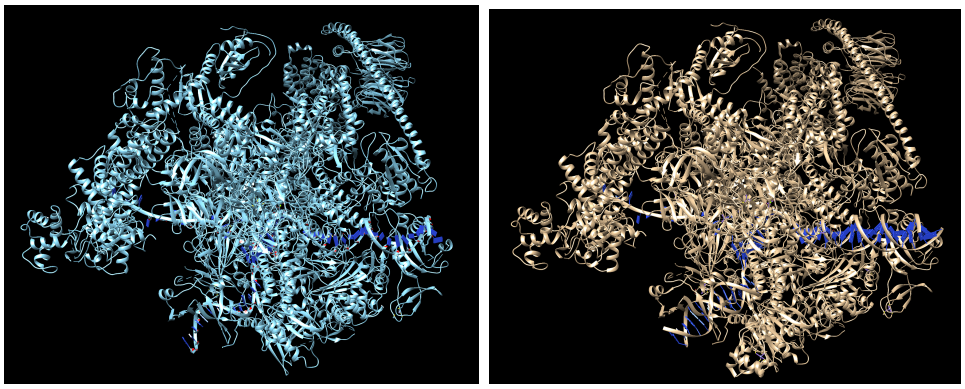


5.4 Example 4

The fourth example was run using 6gmh, an activated transcription complex. The complex is a Hetero 19-mer (stoichiometry of ABCDEFGHIJKLMNOPQRS).

The constructed and original model are structurally the same. The only difference we see is that we are missing some small atoms. This could be due to the pdb interaction files not holding information for this atoms or due to the fact that our superposition method implies the truncation of at least one

of the chains involved in the superposition.



5.5 Example 5

The fifth example was run using 6om3, the Orc1 BAH domain in complex with a nucleosome core particle. The complex is a Hetero 10-mer (stoichiometry of A2B2C2D2E2).

The constructed model replicates just a small part of the original pdb file. This could be due to the fact that our algorithm doesn't detect a clear relationship with the other sub complexes. Although not managing to recreate the whole macro complex the algorithm manages to give to the user a good picture of one of its subunits.