4SMacroBuilder

Constructing macromolecular complexes.

Pau Badia, Altaïr C.Hernández and Natàlia Segura

Index

Introduction

Proteins are the most versatil macromolecules in living systems, as are in charge on multitude of specific chemical trasnformations, which provide the cell with usable energy and the molecules needed to form its structure and maintain the intracellular *homeostasis*. Proteins also recieve signals from outside the cell, starting intracellular signal transductions and regulating the gene expression in different stress situations. Sometimes, proteins monomers interact between other monomers in order to form protein macrocomplexes, known as the quaternary structure of a protein. Some examples are hemoglobin, the ATP syntase, the RNA polymerase or the ribosome.

Nevertheless, understanding how proteins interact with others in the assembly process is not an easy task. For that reason, different research groups have developed methods that predict how this interaction may occur. In the Protein Data Bank (PDB) are stored a large set of proteins with known structure, after a process of x-ray crystallography or Nuclear Magnetic Ressonance, allowing us to study the molecular space and possible allosteric interactions.

It is important to stand out that there are some proteins that are difficult to crystallize due to its molecular conformation or dimensionallity, such as virus capsides. In other cases, it could be found a protein - DNA/RNA interaction, and it could be an interesting feature to study in process of transcriptional or translational processes (it would be the case of ribosomes, for instance).

Objective

The main scope of this project is to reconstruct protein macrocomplexes from individual protein pairwise interactions. In order to do so, we developed a stand-alone application that reads a set of protein - protein interactions in PDB format and reconstruct different multi-subunit complexes.

Background and Scientific explanation

Algorithm implementation

The aim of this project is to build a protein macrocomplex (quaternary structure) with a single previous knowledge: a set of know protein pairwise interacions. Let's suppose we have the following protein - protein interactions:

```
<div class="col-md-12">
  <div class="thumbnail">
    <img src="/approach1.png" alt="approach1_image" style="width:600px;height:500px">
    <div class="caption">
      <h4><b>Figure 1</b></h4>
      <p><i>Here A,B,C are chains/subunits of the macrocomplex and may differ in sequence and structure</i></
    </div>
  </div>
</div>
```

In protein macrocomplexes there are several chains that interact with more than one chain, allowing the rest of interactions to be done. We could start taking one of these pair interactions as a template (i.e. A-B), and then superimpose the rest of the interactions by protein superposition. We have to assume that at least one chain of the template interact with another subunit (in this example A-C). This way we could superpose those identical chains (A-A) and move the new pair interaction to the template. Therefore, we would obtain a resulting structure of three chains (in this example). If we repeat this process until all the simillar chains are superposed, then we would obtain the final macrocomplex.

In order to carry this out we should know the order in which the program would have to superpose these pair of interactions, to avoid clashes between chains or even to prevent the program to superimpose the same chain more than one time. Also, we should know how many iterations the program would have to achive in order to make the final structure.

This could be solved in different ways. For instance, starting with one pair of interactions as template it could checked all the possible interactions in each iteration an see which candidate would satisfy the problem statement(exhaustive search algorithm). Although this approach would be simple to implement, it would have a computational cost proportional to the number of candidate solutions, which would tend to grow in an exponential way.

We approached this limitation focusing on the interacting residues of each subunit. If we imagine the whole macrocomplex structure as a lego puzzle, then we could realize that each chain has some residues that are interacting with at least another chain (hydrophobic residues) and the rest of the residues that are exposed to the solvent environment (hydrophilic residues).

```
<div class="col-md-12">
  <div class="thumbnail">
    <img src="/approach2.png" alt="approach2_image" style="width:600px;height:500px">
    <div class="caption">
      <h4><b>Figure 2</b></h4>
      <p><i>As we can see, the residues indicated by the arrow are *hydrophobic*, while the rest are hydrophi
    </div>
  </div>
</div>
```

Basing on that premise, we stored the interacting residues for each chain, as well as the corresponding chain those residues are interacting with. This is possible as we have this informations in the PDB files, so that would be the first task to do. In that way we could force the program to check in each iteration/superimposition whether those residues are interacting or not. At the same time we would have to consider as feasible complexes those that has no clashes when superimposed, which means that the backbone of the model that is been superimopsed is not interacting with the rest of the complex already joined (this means a threshold distance of 2 A^o).

Then, our program would start to structurally superimpose structures with at least two identical subunits (those that share a pairwise sequence identity $>= 95\%$), but this time for each model the program knows how many interacting sites are in each protein, and even with which specific chain has to interact on those sites.

Let's explain this with an example:

Imagine that we have the same pairwise interactions as before explained:

- A-B
- A-C
- B-D

And know, when we read the PDB files, for each chain we store the interacting residues that take place by looking for those residues that are in a distance no larger that 5 A^o. Besides, we store in the same set which is the model of the corresponding interaction, so that we know which is the model to superimpose later.

For instance:

```
A -> (1,2,3) --> model A-B
  -> (15,16) --> model A-C

B -> [(5,6,9),(13,15,17)]     -> model A-B
  -> [(24,25,28),(33,34)]     -> model B-D

C -> [(5,6,9),(20,22)]        -> model A-C

D -> [(1,2,4,5),(13,15,17)]   -> model B-D
```

pandoc -f markdown+pandoc_title_block
-t latex
–variable papersize:a4paper
–variable geometry:margin=1.5cm
–variable fontsize=10pt
–highlight-style pygments
-o report.pdf
report.md