Acknowledging

I wish to thank David Ritchie and Bernard Maigret who trusted me to work on this project and followed me throughout my internship. I would also like thank Olivier Devillers for his classes on the Delaunay triangulations and his explanations regarding the CGAL library. Finally, it is necessary for me to thank all the Capsid team for having facilitated my integration in the institute as well as all of the INRIA of Nancy.

Table of Contents

Introduction

The INRIA (Institut National de Recherche en Informatique et en Automaique) is a public French institute of research in computer science and mathematics. Founded in 1967, the INRIA accounts for 2600 collaborators gathered on various sites in France, one of which being Nancy. Within this center, the team CAPSID develops algorithms and softwares allowing to study biological phenomena and systems from a structural point of view, thanks to 3D modelling. I did my internship with this team under the supervision of Dave Ritchie who is its manager.

Overseen by Dave Ritchie and Bernard Maigret, this project's main goal is to model the interface of contact between two proteins. The properties of the interface can indeed give a lot of information about the interactions between proteins. This is particularly heplful in fields such as biology and medicinal search for the development of new medicine. The project also takes place in partnership with the research team Vegas, and especially with Olivier Devillers, who participated in the implementation of CGAL (Computational Geometry Algorithms Library).

This library allows, through to the numerous features it offers, to improve and to accelerate the development of the method chosen for the project. We will give more details in this report on the theoretical method adopted to approximate the interface between proteins : the Delaunay triangulation and the Voronoï Diagram. We will also explain how the structures supplied by CGAL are used during the development of the software by insisting on some of the most crucial parts of the implementation. Finally, we will see how the results (and the difficulties encountered) allow to elaborate future prospects for this project.

1 Structure of proteic complex

Proteins are biological molecules found in all the living cells. They are formed of a chain of amino acids [Pro,]. Proteins realize many functions within the living cells. They can have an enzymatic, structural role, they allow the mobility of molecules, the regulation of the genetic expression or to pass on cellular signals. The protein chains forming the proteins are synthesized in the cell. The genetic material of the cell determines the order of the amino acids.

The proteins have a structure in three dimensions which allows them to realize their biological function. Proteins can interact together to carry out some biological functions. These interactions form protein complexes. The structures of the proteins and their interactions are particularly used in medicinal chemistry. The study of surfaces and available spaces can guide the research for new medicine. Crystallography is used to study the structure of proteins at the atomic scale. It is based on the physical phenomenon of diffraction of the electromagnetic waves (X-rays).

The data from a protein that we can us is stored in *.pdb* files (see figure**??**). These files, the reading and the interpretation of the data they contain, are essential to the observation of proteins. Indeed, every line, excepted the first one, corresponds to an atom of protein being studied. These lines contain information such as the chain to which the atom belongs, its amino acid or its address and coordinates in space (Å).

During the project, we will consider that atoms are points of the space of identical masses. First, the address and coordinates of atoms will be extracted. The point cloud obtained constitutes the basis on which the method of determination of the interface is going to be based. It is thus crucial to obtain from the file *.pdb* a set of valid and usable data for the program to be implemented. We will also see that other information, such as the chain (A or B, column 5 of the figure **??**), plays a leading role in the running of the program. Thus the objective is to extract a coherent point cloud (modelling the complex), to apply the triangulation of Delaunay.

```
CRYST1    0.000    0.000    0.000  90.00  90.00  90.00 P 1           1
ATOM      1  CA  MET A  76      11.682 -15.962 -22.500  1.00  0.00           C
ATOM      2  C   MET A  76      10.913 -14.834 -21.818  1.00  0.00           C
ATOM      3  O   MET A  76      11.505 -13.842 -21.393  1.00  0.00           O
ATOM      4  CB  MET A  76      10.786 -16.678 -23.515  1.00  0.00           C
ATOM      5  CG  MET A  76      10.101 -17.879 -22.854  1.00  0.00           C
ATOM      6  SD  MET A  76       8.947 -18.633 -24.027  1.00  0.00           S
ATOM      7  CE  MET A  76       8.478 -20.050 -23.004  1.00  0.00           C
ATOM      8  N   LYS A  77       9.594 -14.989 -21.723  1.00  0.00           N
ATOM      9  CA  LYS A  77       8.754 -13.971 -21.096  1.00  0.00           C
ATOM     10  C   LYS A  77       8.140 -14.495 -19.799  1.00  0.00           C
ATOM     11  O   LYS A  77       7.465 -15.525 -19.790  1.00  0.00           O
ATOM     12  CB  LYS A  77       7.641 -13.555 -22.064  1.00  0.00           C
ATOM     13  CG  LYS A  77       6.806 -12.424 -21.443  1.00  0.00           C
ATOM     14  CD  LYS A  77       5.692 -11.990 -22.407  1.00  0.00           C
ATOM     15  CE  LYS A  77       6.244 -11.023 -23.461  1.00  0.00           C
ATOM     16  NZ  LYS A  77       6.791  -9.814 -22.783  1.00  0.00           N
ATOM     17  N   ASP A  78       8.370 -13.768 -18.710  1.00  0.00           N
ATOM     18  CA  ASP A  78       7.831 -14.148 -17.408  1.00  0.00           C
ATOM     19  C   ASP A  78       6.674 -13.227 -17.034  1.00  0.00           C
ATOM     20  O   ASP A  78       6.415 -12.239 -17.721  1.00  0.00           O
ATOM     21  CB  ASP A  78       8.925 -14.062 -16.342  1.00  0.00           C
ATOM     22  CG  ASP A  78       9.934 -15.188 -16.539  1.00  0.00           C
ATOM     23  OD1 ASP A  78       9.634 -16.101 -17.290  1.00  0.00           O
ATOM     24  OD2 ASP A  78      10.992 -15.121 -15.935  1.00  0.00           O
ATOM     25  N   THR A  79       5.975 -13.552 -15.951  1.00  0.00           N
ATOM     26  CA  THR A  79       4.844 -12.740 -15.514  1.00  0.00           C
ATOM     27  C   THR A  79       5.309 -11.353 -15.082  1.00  0.00           C
ATOM     28  O   THR A  79       6.434 -10.947 -15.373  1.00  0.00           O
ATOM     29  CB  THR A  79       4.127 -13.427 -14.350  1.00  0.00           C
ATOM     30  OG1 THR A  79       5.009 -13.517 -13.240  1.00  0.00           O
```

FIGURE 1.1 – Example of a .pdb file

2 Theoretical method

A protein is thus represented as a point cloud where each of these points represents an atom of the protein. Aiming at optimizing the time to go through this point cloud and modelling the interface between both proteins, we use the triangulation of Delaunay [Devillers, 2014].

This triangulation is unique and can be explained like this : every circumscribed circle of a triangle of the point cloud contains only the points of the aforementioned triangle (see figure 2.1). We shall explain at first the method to determine the interface in two dimensions(size).

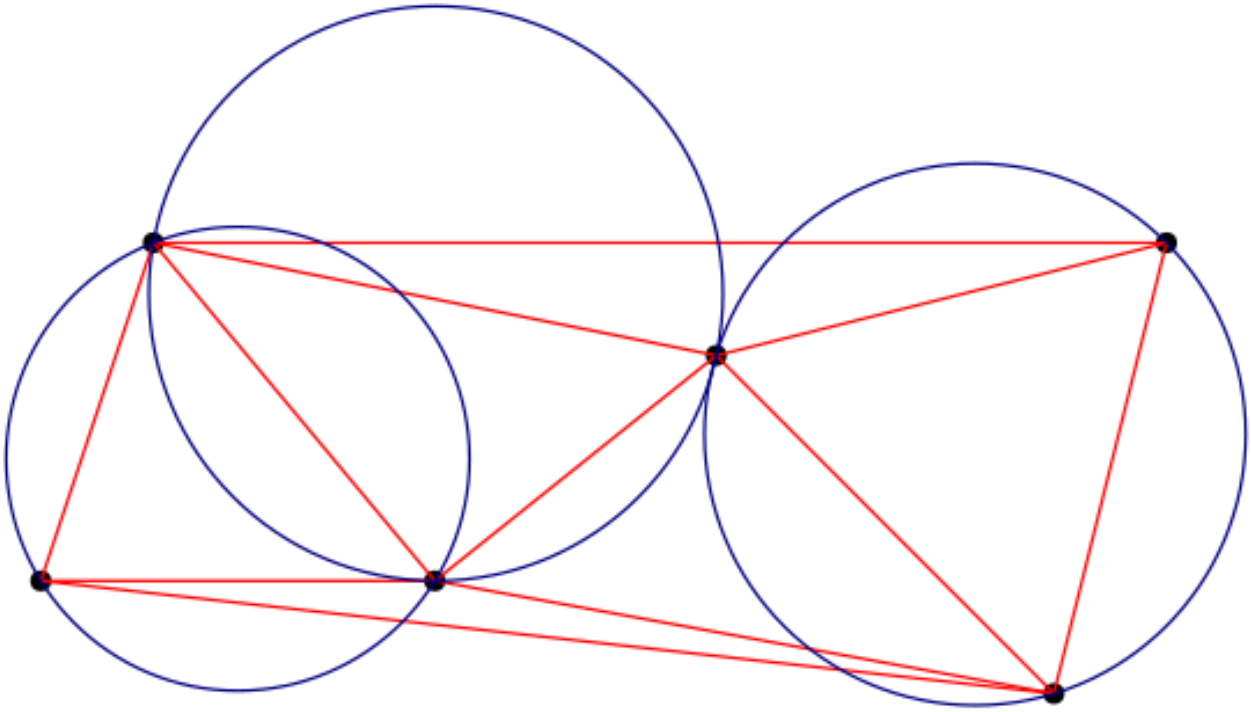


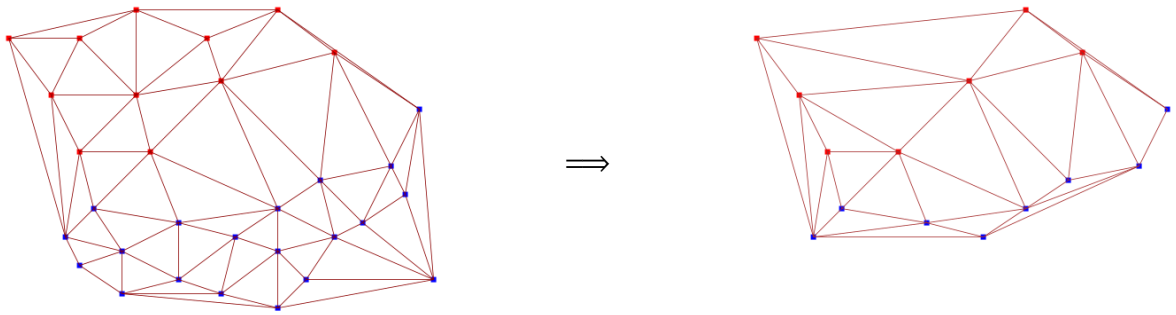
FIGURE 2.1 – Delaunay Triangulation

2.1 Dimension 2

It is necessary to apply this triangulation to the complex of which we want to determine the interface, meaning on the coordinates of the atoms that form the point cloud (see figure 2.2a). The two proteins of the complex are differentiated by their colors (red and blue) on the figure. We select then the useful part of this triangulation (see figure 2.2b) : we keep only the triangles which contain at least a point at the interface.

A point is at the interface if it is in a triangle containing at least a point of every protein.

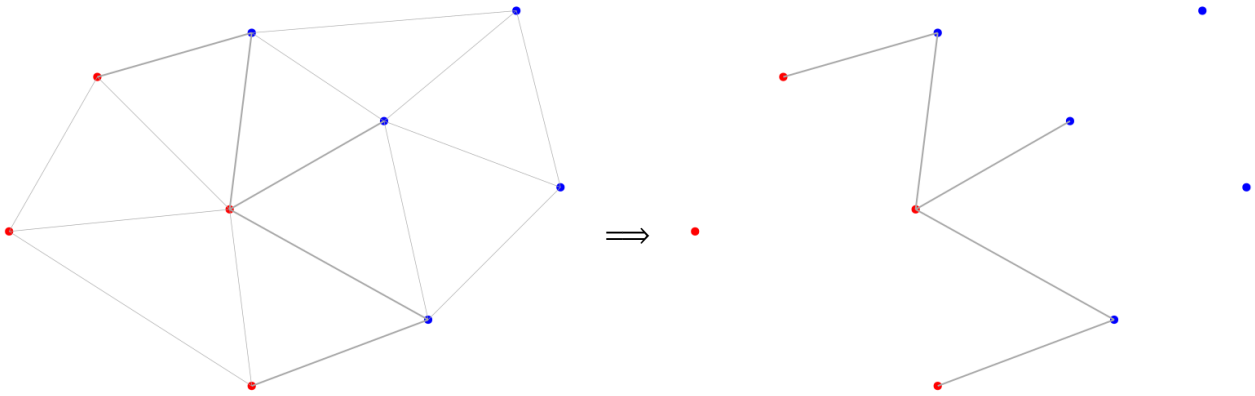
Reducing the size of the triangulation will be useful afterward to accelerate the processing time and the recovering of the data concerning the interface.



(a) Delaunay Triangulation (b) Reduced Triangulation

FIGURE 2.2 – Réducing a triangulation

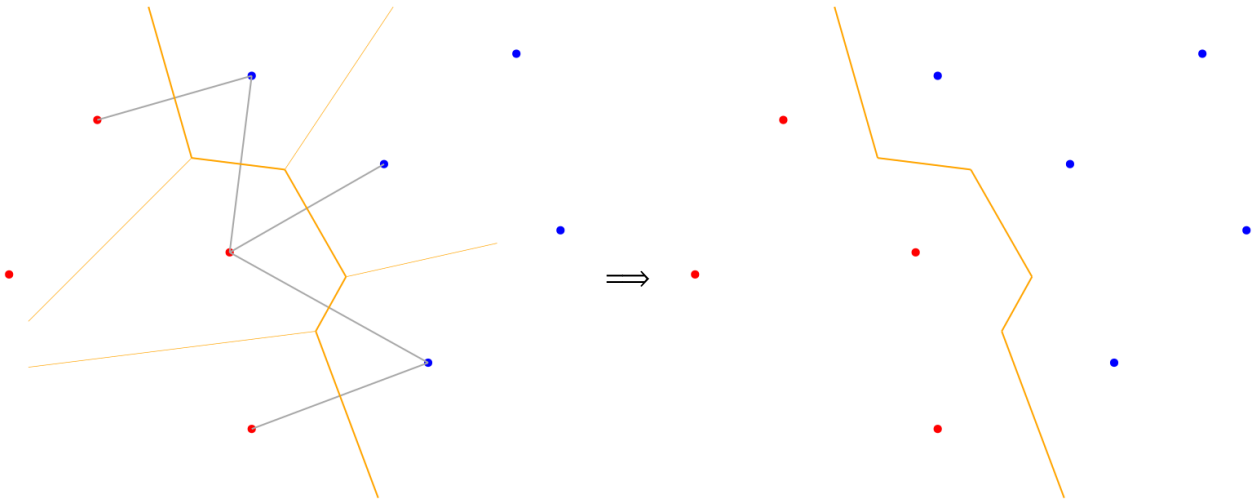
We focus now on the determination of the interface itself. By keeping only the useful edges (see figure 2.3), that is those connecting two points belonging to two different proteins, we can approximate the interface of contact thanks to the Voronoï diagram.



(a) Delaunay Triangulation (b) Edges at the interface

FIGURE 2.3 – Triangulations et useful area

This diagram is the dual of the Delaunay triangulation and represents points equally distant from the points of the triangulation (see figure 2.4a). By keeping only the parts of the Voronoï diagram which correspond to the previously selected edges (to see figure 2.4b), we obtain a line strait by pieces which approximates the interface of contact in dimension 2.



(a) Voronoï Diagramme (b) Interface

FIGURE 2.4 – Calculation of the surface

2.2 Dimension 3

If we transpose the method seen above in dimension 3, the triangles formed by points become tetrahedrons on which we will work to calculate the interface. In the same way, a tetrahedron will be considered at the interface if it contains at least an atom of every protein. Furthermore, the dual of an edge becomes a surface surrounding this edge (see figure 2.5). By gathering these pieces, we obtain a surface in three dimensions which models the contact area between both proteins of the studied complex.

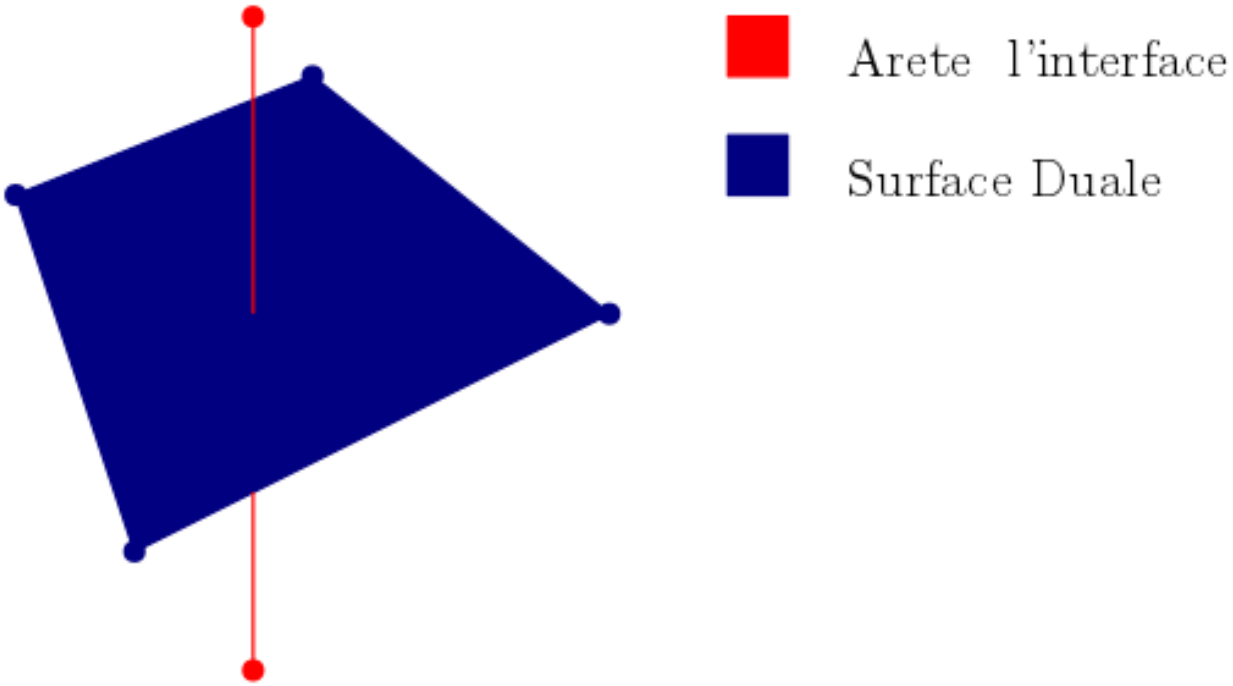
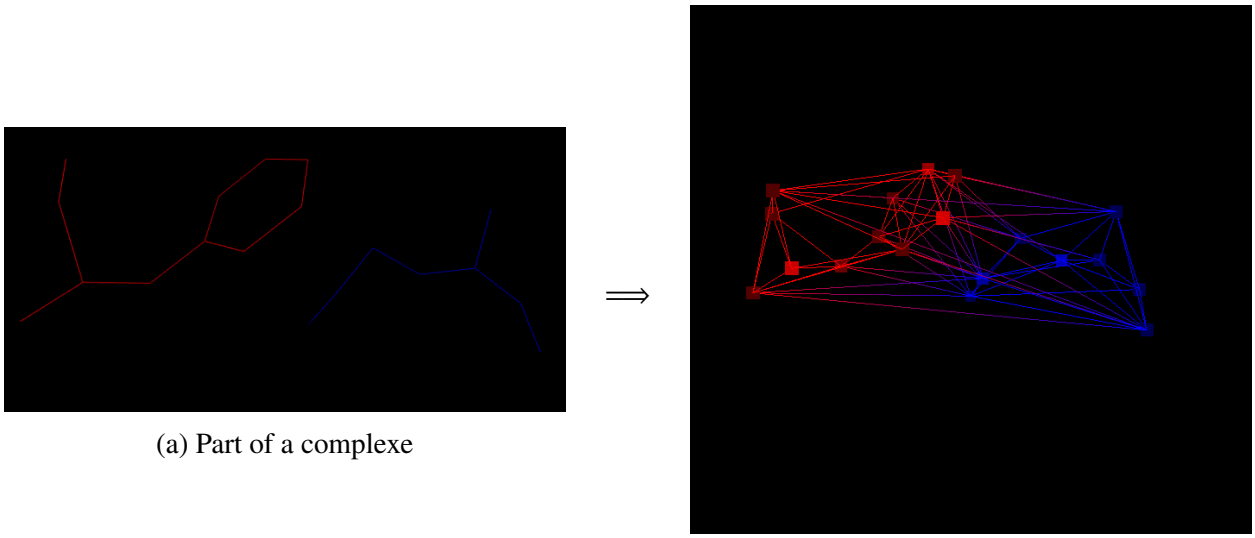


FIGURE 2.5 – Dual of an edge Dimension 3

The triangulation in three dimensions of a point cloud corresponding to the atoms of a complex gives the figure 2.6.



(a) Part of a complexe

(b) Triangulation

FIGURE 2.6 – 3D Triangulation of a complex

3 Implementing the method

3.0.1 CGAL

To develop the method seen previously, we chose to use CGAL (Computational Geometry Algorithms Library) [CGA,]. CGAL is a software project which supplies a free access with numerous effective and reliable geometrical algorithms in the form of a C++ library. CGAL is used in fields needing geometrical calculation, such as geographical information systems, computer-aided design, molecular biology, medical imaging, computer graphics and robotics.

We were particularly interested in a part of CGAL which allows the storage of point clouds in the form of Delaunay triangulations. The perk of this library lies in the structure and the methods accelerating the various stages of the calculation of the interface between two proteins.

CGAL indeed has the specific class (that we can see as a structure) *Delaunay_ Triangulation_ 3*, allowing to calculate and store a Delaunay triangulation from simple arrays (*C ++ Arrays*) listing points in space. Furthermore, to understand the implementation realized during this project, it is important to specify the structure of tetrahedron forming a Delaunay trianguation (see figure 3.1).

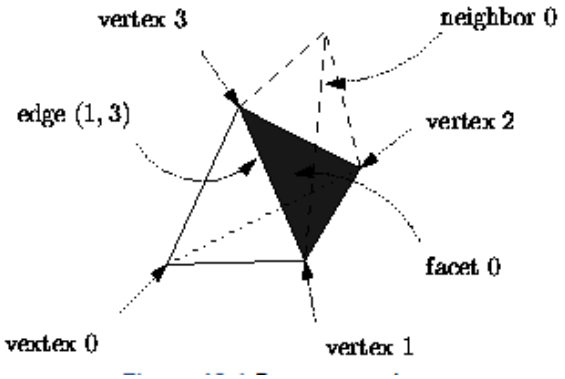


FIGURE 3.1 – Structure of a tetrahedron in CGAL

A tetrahedron is represented through four entities :

— Vertex : contains a point (3D coordinates)

— Edge : contains two vertices in specifi order and a cell

— Facet (face) : stored thanks to a cell and the vertex facing it

— Cell : a tetrahedron giving access to four vertices and four adjacent celles

It is crucial to understand the structure provided by GAL because it will be necessary to access various parts of a tetrahedron. For example, when we work on edges to look for the interface, we need to know the vertices it contains.

3.0.2 Displaying Method

To display the proteins and the interface, we chose the *.off* files which allow to store a list of points (colored or not) and to indicate the links between each of these points (to see figure 3.2).

```
1  [C]OFF
2  19 99 158
3
4  0.239 4.621 1.992 0 0 255
5  0.39 6.049 0.891 0 0 255
6  -0.066 7.299 2.11 0 0 255
7  -1.585 7.305 2.288 0 0 255
8  -3.73 8.162 1.335 0 0 255
9  -1.813 9.651 1.714 0 0 255
10 -2.207 8.305 1.312 0 0 255
11 -4.314 7.545 0.447 0 0 255
12 2.283 7.777 -0.568 255 0 0
13 2.794 6.49 -0.772 255 0 0
14 5.982 6.709 -1.786 255 0 0
15 2.589 8.474 0.606 255 0 0
16 3.915 6.595 1.378 255 0 0
17 3.405 7.883 1.578 255 0 0
18 4.174 4.513 -0.034 255 0 0
19 3.608 5.896 0.201 255 0 0
20 6.103 5.676 -1.129 255 0 0
21 5.091 4.543 -1.262 255 0 0
22 5.769 3.262 -1.424 255 0 0
23
24 3 0 2 3
25 3 1 2 3
26 3 1 0 3
27 3 1 0 2
28 3 7 5 4
29 3 6 5 4
30 3 7 6 4
31 3 7 6 5
32 3 9 10 7
33 3 18 10 7
34 3 18 9 7
35 3 18 9 10
36 3 15 1 8
37 3 13 1 8
38 3 13 15 8
39 3 13 15 1
40 3 1 8 7
41 3 9 8 7
42 3 9 1 7
43 3 9 1 8
44 3 6 2 3
45 3 6 1 3
```

FIGURE 3.2 – Example of a *.off* file

The first line indicates the type of the file (OFF) and the presence of coloring : [C] The second line gives, in this order, the number of points, the number of cells and the number of edges, the latter not being necessary to reading the file. In our example, the lines 4 to 22 list the coordinates of the points and the color associated with each. The color is stored in RGB (Red, Green, Blue) with integers between 0 and 255 or floats between 0.0 and 1.0.

Beyond the line 23, the cells are listed, with coming first an integer on every line giving the number of points of the cell. Since our example represents Delaunay triangulation, this number is worth 3 because every face of the triangulation corresponds to a triangle (it will not be valid any more for the surface faces which can contain any number of points). Following this integer come the indexes of the points forming the cell (or the face). This index corresponds to the rank of the point in the coordinates listed above.

Finally, thanks to the CGAL library and this visualisation method, we were able to implement the theoretical method described previously.

Conclusion

I thus participated in a project to help the research in biology and in medicine. The interactions between proteins can indedd allow the synthetisation of new medicine. As it matters, these interactions can be understood by analyzing the interface of contact between both proteins of a complex. The goal of the project was to model this interface in the form of a surface in three dimensions.

Using a method based on the Delaunay triangulation and the Voronoï diagram, we saw that it is possible to approximate this surface. To implement this method, we used a C ++ librarywhich helps developping geometrical algorithms named CGAL. The structures and functions were of a paramount importance for the good progress of the project. Using the data for the modelling of the surface is indeed facilitated by the classes provided by CGAL and the iterators which their are bound to. Going through Delaunay structures constitutes one of the most important stages of the development. After having modelled the interface of contact between two proteins, we were capable of visualizing our results thanks to *.off* files and MeshLab.

Finally, the method was a success, in spite of some difficulties in term of compilation. On the other hand, the CGAL library proved its utility and its efficiency despite its sometimes heavy and laborious use. The main objective was fulfilled : the visualization of the interface is functional and the structure of the project allows to implement new features.

Liste des illustrations

Bibliography

[CGA,] Documentation cgal. https://doc.cgal.org/latest/Manual/packages.html. Accessed : 2017-09-15.

[Pro,] Structure d'une protéine. http://www2.cegep-ste-foy.qc.ca/profs/gbourbonnais/pascal/fya/chimcell/notesmolecules/proteines_2.htm. Accessed : 2017-09-30.

[Devillers, 2014] DEVILLERS, O. (2014). Delaunay triangulation and randomized constructions. encyclopedia of al- gorithms, springer.

Annexes