Lipid bilayer analysis using 3D Voronoi diagrams

Robert Szczelina[1], Krzysztof Murzyn[2]

1) Jagiellonian University, Faculty of Math. and Comp. Science, Department of Comp. Science, robert.szczelina@uj.edu.pl,
2) Jagiellonian University, Faculty of Biochemistry, Biophysics, and Biotechnology, Department of Computational Biophysics and Bioinformatics

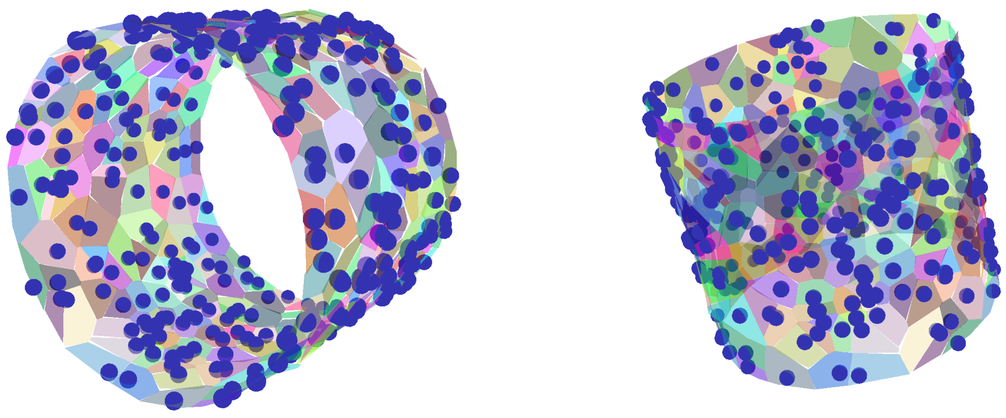
JAGIELLONIAN UNIVERSITY IN KRAKOW

Introduction

In the study of Moleculer Dynamics (MD) simulations of dipalmitoylphosphatidylcholine (DPPC) bilayer in the liquid-crystalline phase we investigated certain geometric quantities as molecular volume (V_L) and surface area per lipid molecule (SA) using 3D Voronoi Diagrams. We also developed an automatic procedure to identify solvent molecules trapped inside the lipid bilayer. In the context of lipid bilayer MD simulations the distribution of particles inside simulation box is close to uniform. This simplifies the structure of resulting Voronoi Diagrams which allows to choose simpler, light-weight and more flexible approaches to build 3D Voronoi Diagrams [1] where construction of Voronoi cells is performed by optimized local cuts by the dividing planes between neighbouring particles. On a Intel Pentium i7 processor (3.2 GHz) computation of Voronoi Diagram for 41600 particles takes about 1s.

DMG-α Library

In our analysis we used our *DMG-α* package, a C++ library and Python bindings for computation of various geometrical aspects of lipids in lamellar and non-lamellar arrangement, which is currently under development. Our library contains routines for computing Voronoi Diagrams in 3D, Alpha Shapes, surfaces, approximate Voronoi Diagrams of projections to arbitrary 2D surfaces and data structures allowing for simple access to all geometric properties of computed objects. We also created routines to effectively handle small changes in configurations of atoms and corresponding 3D voronoi data structures which occur in consecutive time frames of MD simulation trajectory. Our approach is based on Kinetic Data Structures developed for efficient modification of data structures for objects changing continuously in time [4].



DMG-α is designed to simplify most of the tasks and also contains useful components for displaying results. For example building and displaying Voronoi Diagram on the side of the cylinder for 300 random points (shown in the picture) is as simple as:

```
display = render.Render()
data = [(i, random()-0.5, random()-0.5, random(), 0.01)
            for i in range(300)]
geometry = CastCylinderGeometry(1.0, 1.0)
container = Container(geometry)
container.add(data)
diagram = Diagram(container)
for i, cell in enumerate(diagram):
  for (neighbour, polygon) in cell.sides:
    if (geometry.on_boundary(neighbour)):
      display.add_polygon(render.ConvexPolygon(polygon))
display.run()
```

DMG-α will be available to anyone interested shortly after the initial publication of developed algorithms, data structures, and sample applications. We believe that our tool will make possible to investigate qualitatively numerous biological phenomena occurring in the lipid membrane or in biologically relevant nonlamellar lipid phases and at various types of interfaces including water/lipid membrane and lipid/membrane protein. *DMG-α* project is supported by the Polish National Science Center under grant no. 2011/01/N/ST6/07173.

References

[1] Chris H. Rycroft, *Voro++: A three-dimensional Voronoi cell library in C++*, Chaos 19, 041111 (2009).

[2] *CGAL, Computational Geometry Algorithms Library*, http://www.cgal.org

[3] John F. Nagle, Stephanie Tristam-Nagle, *Structure of lipid bilayers*, Biochimica et Biophysica Acta 1469 (2000)

[4] L. Guibas. *Kinetic Data Structures. In Handbook of Data Structures and Applications*, D. Mehta and S. Sahni, Eds, Chapman and Hall/CRC, 2004.

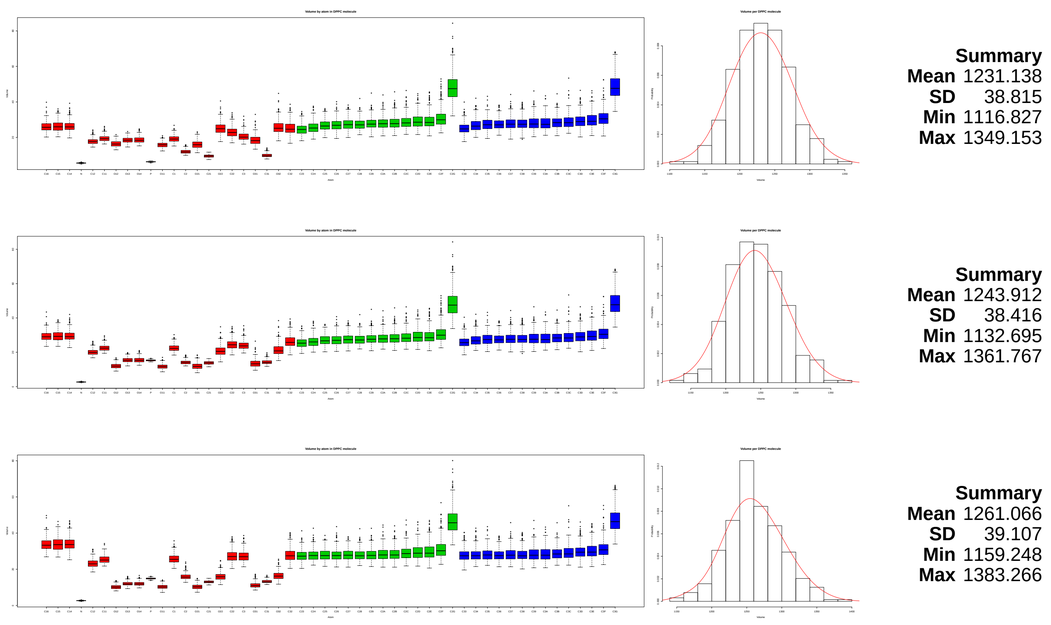
This poster was typeset using *baposter* class http://www.brian-amberg.de/uni/poster/

Volume analysis

We computed distribution of volume per heavy atom in each DPPC molecule using three methods:

1. classical Voronoi diagram for heavy atoms only - we excluded all hydrogen atoms from computation, both in DPPC and water molecules
2. weighted Voronoi diagram for heavy atoms only - we again excluded hydrogen atoms and we used weighting by Van der Waals radii.
3. weighted diagram for all atoms - hydrogen atoms are included, their volume is then included in the corresponding bonded heavy atom. Again, van der Waals radii were used for weighting.

The results are shown in the following figures:



We compared those results to the experimental values found in [3]. The most accurate results were obtained using the simplest method 1. The reason for this is probably the dense packaging of the atoms in the simulation, which itself defines the inner geometry of the space, so no additional constraints (weighting) on the voronoi cells are needed.

Area analysis

3D Voronoi diagrams allow computation of the interface area between solvent and the lipid bilayer, defined as the set of points equally distant to both sets - the set of DPPC atoms and the set of water molecules.

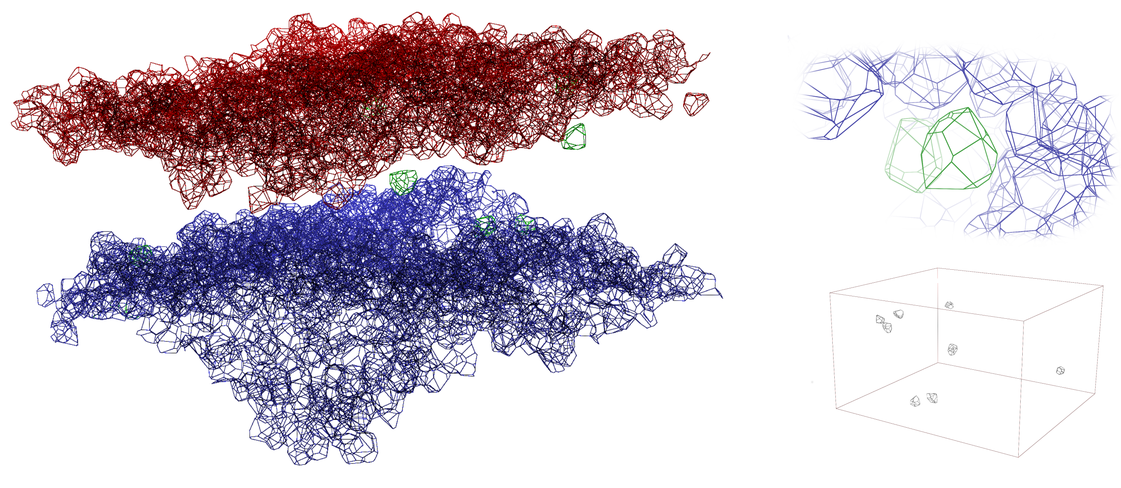


Figure 1. Interface between DPPC bilayer and solvent molecules (left). Zoom on the Voronoi cell of tthe water particle trapped inside bilayer near the surface (upper-right corner). Location of all water molecules trapped inside the bilayer, simulation bounding box drawn for reference (lower-right corner).

The construction of this set is done by the following algorithm:

```
for solvent atom in data:
  mark as visited (water molecule)
  queue = [water molecule]
  while (not empty(queue)):
    cell = compute voronoi cell(queue.pop())
    for (neighbour, polygon) in cell.sides:
      if (neihgbour is solvent atom):
        queue.append(neighbour)
      if (neighbour is DPPC atom):
        include polygon in output
```

The advantage of this approach is that we can extract information on the water molecules trapped inside the lipid bilayer. For each iteration of the outermost loop we get a collection of polygons and if we assume periodic boundary conditions then (generally) the biggest of those groups is the outer interface surface of the DPPC bilayer (shown in blue and red in the picture), the other groups corresponds to water trapped inside the bilayer. We can now use this informatione for e.g. tracking the dynamics of the trapped particles during the simulation.