Analysis of Random Walks on Voronoi Tessellation of Molecular Structures Data

Author: Wint Y. Hnin[1] as part of CRA-W DREU 2014,

Mentors: Amarda Shehu[2] and Estela Blaisten-Barojas[3]

[1]Department of Computer Science, Cornell College

[2]Department of Computer Science, George Mason University

[3]Computational Materials Science Center, George Mason University

31st July, 2014

1. Introduction

The research project is done based on Met-enkephalin (met-enk) peptide, which is a naturally occurring opioid found in the human body. This opioid relates to the stimulation of pain, reward and other emotional behaviors. Met-enk has a flexible structure that it could interact with multiple forms of opioid receptors.

Professor Shehu, Professor Blaisten and undergraduate student researchers provides structures of met-enk corresponding to local minima of the AMBER ff12SB force field in explicit solvent. These structures are subjected to Principal Component Analysis (PCA) to get representations of these structures in a few dimensions. My task consists of employing such representations to obtain further statistics on excursions of met-enk between local minima of its energy landscape. I proceed with first embedding the structures in a Voronoi graph, and then using random and biased random walks on the graph to collect statistics of interest. The techniques I develop are described below.

2. Methods Developed

2.1. Voronoi Tessellation

Background: Voronoi graph is a type of 3D graph with dividing borders. Each point, in our case, each model structure in its low-dimensional PCA-based representation, is centered, and a number of faces are constructed around that point by a number of edges and vertices, building a Voronoi cell for each point. Each Voronoi cell has topological neighbors, with which its faces, edges and vertices are shared. A complete Voronoi tessellation is formed by all cells together.

In this research, the already existing voro++ library (Rycroft, 2009), written in C++, is used to construct the Voronoi graph for the simulated structures. The voro++ library carries out the computations of the Voronoi tessellation, and it produces detailed information about each Voronoi cell. The number of faces and neighbors of each cell corresponding to a structure are particularly interesting for our purpose.

In the way of constructing the Voronoi cells, two choices of boundary conditions are used. One type is rectangular boundaries, in which the outermost cells are stretched to meet with the container walls, and the other one is irregular boundaries, in which all cells have its own boundaries as calculations instead of stretching out. 6000 met-enk structures in their 3D PCA-based representations are fed to the voro++ library twice, with different boundary conditions. The desired data, which are the number of faces and neighbors and the IDs of the neighboring structures of each model, are saved for later analysis.

2.2. Random and Biased Random Graphs

Among different possible directions, two types of random graphs are chosen to perform on the data from the voro++ library. Two programs are written in java for different types of graph. In each, the structures are connected randomly according to the following two approaches. First, there will be a starting point in the graph, from which one of its neighboring structures is selected at random. Then, the topological neighbors of the chosen cell are retrieved, and one is chosen randomly. This process is repeated until all 6000 cells are visited at least once. The random path is performed for 6000 times starting at different cells, and it is implemented for the both Voronoi tessellations with rectangular boundary conditions and irregular boundary conditions.

Another biased random graph, also performed on both tessellations, is Metropolis Monte Carlo (MMC) approach. In this approach, the neighbors are selected with biased randomization. One of the neighbors of the starting point is selected randomly, then the probability is calculated as $e^{-(\Delta E(\text{chosen neighbor}) - \Delta E(\text{current point}))/0.596}$, in which ΔE is calculated as the difference from the energy of its own model to the minima energy among 6000 models. Then, a random number between 0 and 1 is picked, and compared to that probability.

If the probability is greater than or equal to the random number, the point is connected to the neighbor, and the same procedure is repeated for the chosen neighbor. If the probability is less than the random number, another neighbor has to be selected, and compared its calculated probability to another random number. The Metropolis Monte Carlo walk is terminated when a point has tried all of its neighbors for 10 times, but cannot select any as they do not meet the criteria. The program is run 6000 times on both Voronoi tessellations with rectangular boundary conditions and irregular boundary conditions.

3. Results Obtained

The following graph shows the model number in the X-axis and the number of neighbors around each model in the Voronoi tessellation obtained from voro++ library in Y-axis. The left scatter plot shows the results from Voronoi graph with rectangular boundary conditions, and the right one is based on the graph with irregular boundary conditions.

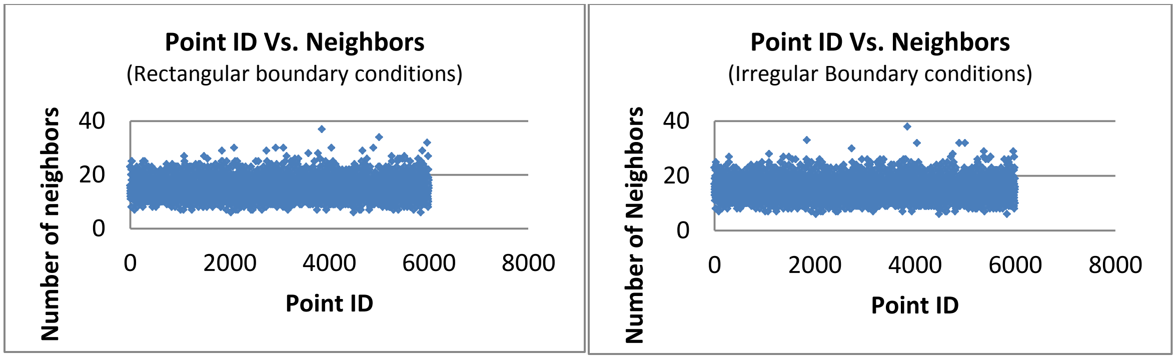

Figure 1. Point ID vs. Number of Neighbors based on data from Voronoi graphs with rectangular boundary conditions and irregular boundary conditions

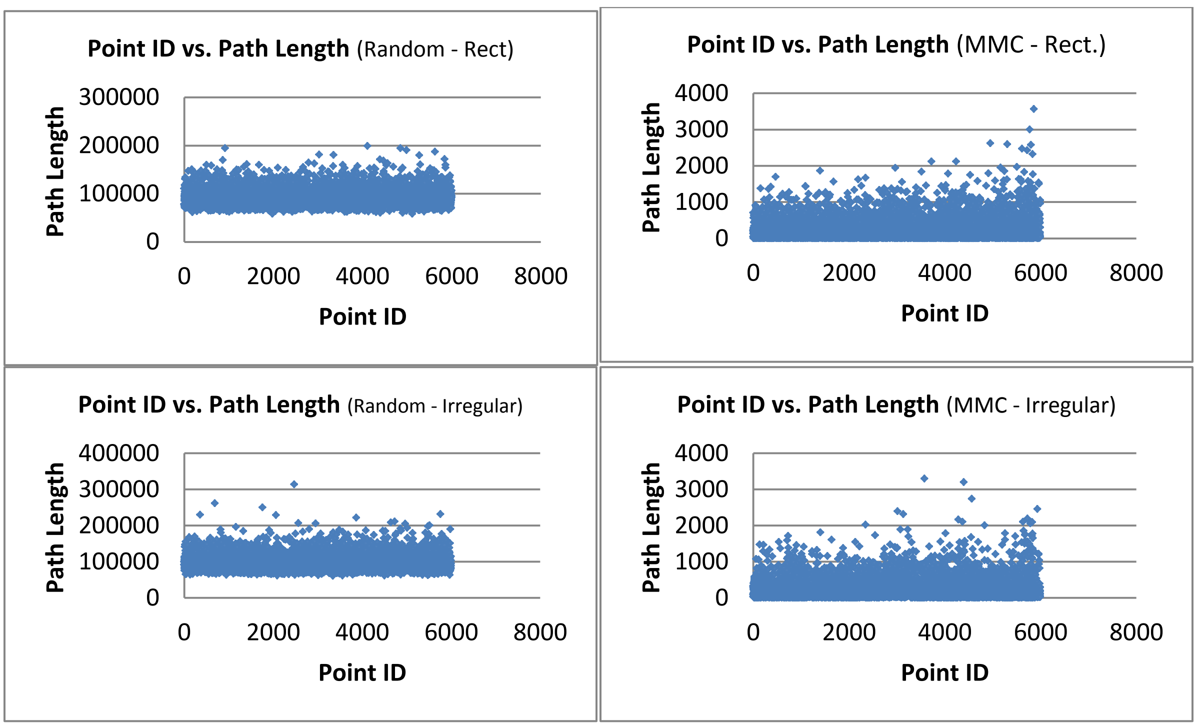

Figure 2. Point ID vs. Path Length from random walk and Metropolis Monte Carlo walk run on Voronoi cells with rectangular boundary conditions and irregular boundary conditions

The above graphs show the path length vs. the starting point ID of each path with various conditions. The graph in the top left corner shows the data run for random walk on the graph with rectangular boundary conditions, and the one next to it represents for Monte Carlo walk on the same Voronoi graph. The bottom left graph shows the statistics for random walk on the graph with irregular boundary conditions, and the bottom right graph is produced from Monte Carlo walk on the Voronoi graph with irregular boundary conditions.

4. Discussion

According to Figure 1, the Voronoi tessellation between the rectangular boundary conditions and the irregular boundary conditions do not have many differences except some of the structures have one or two more or less number of neighbors because of the boundary conditions.

According to Figure 2, it can be clearly seen that the random walks are much longer than the Monte Carlo walks as all 6000 points are visited in random walks. Both random walks have path lengths of mostly 1000. However, the maximum path length performed on the Voronoi graph with rectangular boundary conditions is approximately 200,000 while the maximum path length for the graph with irregular boundaries is over 300,000.

Most of the Metropolis Monte Carlo walks have path length of less than or around 1000. The walks on the graph with rectangular boundary conditions show that the Met-enk model IDs around 6000 tend to have longer path length while in the Monte Carlo walks on the Voronoi tessellation with irregular boundary conditions, the longest path lengths are near structure ID 4000.

Acknowledgement

The author thanks Professor Shehu and Blaisten-Barojas for guidance and provision of data. I also thank all other students in the lab for assisting me at various points during this project.

Citation

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