Computational study of adhesion receptor clustering at cellular interfaces

A PROJECT REPORT

Submitted as part of BBD451 BTech Major Project (BB1)

Submitted by:

Radhika Bagmare

2019BB10045

Guided by:

Professor Amit Das

DEPARTMENT OF BIOCHEMICAL ENGINEERING AND BIOTECHNOLOGY

INDIAN INSTITUTE OF TECHNOLOGY DELHI

February 2023

DECLARATION

I certify that

a) the work contained in this report is authentic and has been done by me under the direction of my supervisor(s).

b) I adhered to the instructions given by the Department, while writing the report.

c) I have complied with the standards and directives outlined in the Institute's Honor Code of Conduct.

d) I have always provided credit where credit is due while using resources (data, theoretical analysis, figures, and text) from other sources by citing them in the report's text and providing information about them in the references. Also, whenever necessary, I obtained consent from the materials' copyright holders.

Signed by:

Radhika Bagmare

CERTIFICATE

It is certified that the work contained in this report titled "**Computational study of adhesion receptor clustering at cellular interfaces**" is the original work done by **Radhika Bagmare** and has been carried out under my supervision.

Signed by:

Name of the Faculty Mentor

Date:

ABSTRACT

Adhesion receptors are members of more varied proteins found on the cell surface. These proteins contribute to the process of cell adhesion as they conciliate the physical attractions between the surrounding cell environment (Extracellular matrix - ECM) and the cells. They are essential for a variety of biological functions, including as cell attachment, mobility, and communication using signals. A family of adhesion receptor known as E-Cadherin forms mechanical links between neighbouring cells in epithelial tissues which is the basis of many multicellular processes. These E-Cadherin molecules tend to form clusters which stabilizes the mechanical links between cells. The goal of the current work is to use computational modelling to understand the physical mechanism of E-Cadherin clustering. We employ Langevin dynamics simulations of E-Cadherin molecules on a plasma membrane. We also model the interactions between the proteins using standard pair potentials and study the effects of endocytosis of E-Cadherin on the clustering process.

TABLE OF CONTENTS

List of Abbreviations

Abbreviation	Description
ECM	Extracellular matrix
E-cad	E-cadherin
EMT	Epithelial-mesenchymal transition
PBC	Periodic boundary condition
MIC	Minimum Image Convention

Chapter 1

INTRODUCTION

In order to communicate with its surroundings, a cell uses adhesion molecules. They are structural proteins that develop as a result of non - covalent bonds among adhesion molecules in one surface of the cell and receptors of a different kind in nearby cells or mediator molecules in the ECM components. For the preservation of tissue cohesion and cell communication, cell adhesions are essential. With the transmission of forces by adhesions, interaction and sensing take place.

A transmembrane protein called E-cadherin is essential for regulating cell attachment in epithelial cells. The preservation of tissue integrity as well as the control of epithelial tissue activity depend on the aggregation of E-cadherin at intracellular boundaries.

Cell-cell adhesion occurs when E-cadherin receptors on neighbouring cells interact with one another and aggregate at the cellular membrane, strengthening the connections between cells. The preservation of tissue integrity as well as the control of epithelial tissue activity depend on the aggregation of E-cadherin at intracellular boundaries as well as the maturation of these cell-cell adhesions [6].

By creating homophilic contacts with E-cadherin molecules on nearby cells, E-cadherin mediates cell-cell adhesion. A number of cytoplasmic proteins, including beta-catenin, alpha-catenin, and p120 catenin, engage with the intracellular domain of E-cadherin. These interactions are crucial for connecting the E-cadherin adhesion complex to the cell's actin cytoskeleton. An essential stage in the development of cell-cell adhesion is the production of E-cadherin clusters at cellular interfaces. The lateral connection of E-cadherin molecules, which takes place through a number of contacts, such as calcium-dependent interactions and interactions between the cytoplasmic domains of E-cadherin and catenin, results in the formation of E-cadherin clusters [1].

Beta-catenin, alpha-catenin, as well as other F-actin-attaching proteins like Vinculin and EPLIN are necessary for the associations among E-cadherin and F-actin. The significant reworking of cell connections occurs concurrently with the development of epithelia. Firstly , this demands for control of cell-cell adhesion, such as by regulating cadherin concentrations or renewal by endocytic recycling. Secondly , junction reshaping is based on the intercellular actomyosin system, which controls regional cortical tension . Surprisingly, the two : cortical tension & adhesion rely on E-cadherin subunits, they cannot operate autonomously. E-cadherin is required to manage the stresses transferred at cell contacts because cortical actomyosin pushes on E-cadherin units through alpha-catenin. In order to assist the development and retention of cell connections as well as the transfer of tensile stresses at cell interaction during tissue remodelling, E-cadherin complexes serve these two mechanical purposes that are interwoven.

Chapter 2

MOTIVATION

The main goal for the project is to study the effects of endocytosis on E-cadherin clustering. The rate of endocytosis is known to influence the organization of E-cadherin molecules on the cell surface of many metazoan organisms.

Endocytosis is a cellular process in which endocytic vesicles carry extracellular substances, such as nutrients and signalling molecules, into the cell. E-cadherin endocytosis is a managed interaction that includes the assimilation of E-cadherin from the plasma film into endocytic vesicles. This process is thought to play a role in regulating E-cadherin expression and function at the cell surface.

The regulation of E-cadherin endocytosis is a complex process that involves a variety of molecular interactions and signalling pathways. Dysregulation of E-cadherin endocytosis has been implicated in a number of diseases, including cancer and developmental disorders.

Quick E-cad acquisition by endocytosis and a significant drop in E-cadherin concentrations are linked to a collapse of column-shaped epithelial structure all through epithelial-mesenchymal transition (EMT), a crucial stage in the cancer formation [1]. Also demonstrated to be the cause of cell-sorting process in tissue culture is variable appearance of adhesion molecules. Thus, a better understanding of the molecular mechanisms underlying E-cadherin endocytosis is critical for the development of new therapies to treat these diseases.

E-cadherins don't really aggregate evenly at adherens junctions; rather, they form clumps of widely distributed sizes in a variety of cell types which can be observed by diffraction-restricted optics. Because of the transmission of cell stress by these clumps, it is reasonable to assume that cell-cell interactions at junctions happen through discreet arrangement of E-Cadherin, not a continuous one.[1]

Moreover, it is anticipated that local forces exchanged by the cells are affected by the physical size, and in-plane movement of E-cadherin clumps. It has been recently shown that these clumps are formed by lateral specific interactions of E-Cadherin [9]. However the nature of this specific interaction is not known. Further, these E-Cadherin clusters of two neighbouring cells come together to form the adherens junctions and contribute in maturation of the junctions [10]. It is also known that the endocytosis of E-Cadherin plays an important role in stabilization of the clusters. However, it is not clear how the dynamics of endocytosis of E-Cadherin couples with the movement of E-Cadherin on cell surface. Through our project we want to investigate the nature of interaction between E-Cadherins and the above coupling using computational modelling.

Chapter 3

LITERATURE SURVEY

In the studies conducted in 2013, it was found that endocytosis reduces the amount of E-cad on interfaces by preventing the creation of large clumps by removing only those that are of interest. One hypothesis is that bigger groups may be endocytosed more often than smaller ones because they make it easier for the endocytic machinery to be built. Macroscopic clumps can disrupt the actomyosin system and result in tissue movements freezing. Endocytosis prevents this from taking place.[1]

Irrespective of the fact that E-cad may aggregate without actin, the data obtained from the experiment conducted showed that E-cad associations with actin are necessary for E-cad clump maintenance in live epithelia. Recently it has been shown through a computational model that E-cad may simply aggregate under lateral forces [7]. The disintegration of cadherin complexes is prevented by actin-based control. Even though laboratory findings demonstrate that E-cadherin transfers stress in between cytoskeleton and the cell surroundings , the problem as to the capability of E-cadherin clustering to regulate the force transfer inside the cell was not understood [1] . Recently it has been shown through a computational model that E-cad may simply aggregate under lateral force. A possible source of these force is the actin- cytoskeletons [7].

E-cadherin hinders cell movement on matrix. Its role in cell movement via cell-abundant tissues is uncertain. Researchers used in vivo mechanical stress sensors, along with cell-type-specific RNAi, photoactivatable Rac, and morpho dynamic profiling to investigate how E-cadherin affects cell movement. They observed that border cell-nurse cell adhesion stabilises forward-directed protrusion and directionally consistent mobility in the Drosophila ovary. Leading cells instruct followers via boundary cell adhesion. Motile and polar cells stick together and polarise each cell.[2]

RAB5A, a necessary endocytic protein, induces the production of separate, actin-based lump projections which produce traction forces which are transferred to long ranges via junctional

contacts. A positive feedback loop among polarity and total displacement is created by excessive mechanical coupling, which enables cells to get directional instruction signals from their neighbours. This promotes neighbouring cells to synchronize their front-rear polarity. Multicellular l ifeforms become dynamic as a result of this as well as junctional E-cadherin dynamic is improved to handle cell nearby sharing, volume, density, and stress changes. Furthermore, these elasticity changes increase the mobility of monolayers that would otherwise be jammed, allowing invasive, group movement while restrained and quick multicellular controlled movement all through embryo development.[3]

Tissue fluidity and development are impacted by intercalation. These two roles appear to be distinct depending on whether tissue changes shape. Cell-cell adhesion and cortical contractility are the mechanisms by which intercalation promotes development. We know very little about tissue intercalation. Cell-cell connections, though, might describe it. For cell differentiation and delamination from the embryonic mouse epidermis, that promotes stratification, a jammed condition must emerge. Hence, altering tissue fluidity could have broad implications as it influences growth, homeostasis, and disease.[4]

Most cancer fatalities are due to metastasis. Invasion and metastasis may start after E-cadherin depletion as there is negative associations between in vitro movement and E-cadherin concentrations. Majority of breast cancers are invasive ductal carcinomas that express E-cadherin in both primary tumours and metastases. To address this discrepancy, we examined the genetic role of E-cadherin in luminal and basal invasive ductal carcinoma metastasis in mice and human systems. E-cadherin removal increased invasion but lowered tumour growth, survival, the number of circulating tumour cells, faraway organ implantation, and metastatic spread. E-cadherin-negative cell population formation was restored when TGF-receptor signalling, reactive oxygen species accumulation, or death were inhibited. It is possible to treat metastatic breast cancer cells by preventing E-cadherin-mediated survival.[5]

Chapter 4

MATERIAL AND METHODS

When two individual cells interact with one another, the E-cadherin molecules present on the cell membrane come into play. These E-cadherin molecules present on each cell membrane interact with one another resulting in cell-cell adhesion. The stability of the cell-cell adhesion increases when E-cad molecules form clusters on one cell surface and interact or bind with E-cad clusters on another cell surface.

Figure 4.1 : E-cadherin clustering schematics

However, the mechanism for this cluster formation is not known. Also, the rules governing the clustering and the interactions between molecules are not determined.

Hence, to analyse this interactions further we make certain assumptions. We govern the motion of the protein molecules using Langevin equation of motion, given as :

$$d^2X / dt^2 = - \Gamma \, dX/dt + F_{int} + F_{random} \qquad \text{Eq.(4.1)}$$

where, left hand side represents the acceleration of a particle or object in the x direction, an it also depicts the inertial force, Γ is the friction coefficient, dX/dt is the particle velocity, F_{int} is the interaction force between two particles, and F_{random} is the force due to random motion of the fluid or environment [8].

Computational Langevin dynamics simulation is used to model the stochastic dynamics of complex systems like biological systems, materials, and molecular systems. It can model the

effects of thermal fluctuations and random forces on the system, which are often critical for accurately predicting the behaviour of complex systems.

A) Without interaction between molecules

Initially, we look at random movements of non-interacting protein molecules on the cell membrane. For this motion we assume the over-damped limit that is when the inertial forces become negligible. Here we implement the overdamped limit of proteins considering they move against high viscosity of the membrane. Considering these dynamics, now the equation of motion becomes :

$$\Gamma \, dX/dt = F_{random}$$

Here we make another assumption. Instead of a Brownian random force, we consider an active random force experienced by the protein. This force models the effect of F-actin in the cell cortex which transiently couple to the proteins and move them [7]. Solving this equation further we get:

$$\Gamma \, dX/dt = F_{random} = v_0 \, \hat{n} \quad \text{Eq.(4.2)}$$

v_0 is the characteristic velocity of the protein molecules arising from F-actin, and $\hat{n}$ is a unit vector that points in a random direction at any instant. dX is the displacement with time t. [8]

Reshuffling the equation (2) gives us: $dX = (v_0 \, \hat{n} / \Gamma) \, dt$

We can write this as: $X(t + \Delta t) = X(t) + v_0 \, \hat{n} / \Gamma \, \Delta t$ Eq.(4.3)

Now, since the unit vector is given by: $\hat{n} = \hat{\imath} \cos\theta + \hat{\jmath} \sin\theta$

θ is the angle made by the unit vector with a reference direction at time t. Equation (3) can be further written in terms of X and Y components as :

: $X(t + \Delta t) = X(t) + (v_0 / \Gamma) * \cos\theta \, \Delta t$ .....Eq.(4a)

: $Y(t + \Delta t) = Y(t) + + (v_0 / \Gamma) * \sin\theta \, \Delta t$ .....Eq.(4b)

We initially assume that ɵ evolves randomly in time. In future we will implement a suitable rate of changes in ɵ.

B) With interaction between molecules

Going towards a more realistic approach to study the adhesion receptor clustering on cell surface, we will be assuming the presence of harmonic spring-like bonds between two E-cadherin molecules that come in proximity of each other. Harmonic spring-like bonds refer to a type of potential energy function that is commonly used in molecular simulations to model the interactions between atoms or particles. In this type of potential energy function, the energy of the system is determined by the distance between pairs of atoms or particles, and is modelled as a spring-like bond. The potential energy of the bond is proportional to the square of the deviation of the distance from its equilibrium value, which is the distance where the energy is at its minimum. So, F_{int} which is the interaction force between two particles in Equation(1) is given by the first derivative of the potential energy U. The potential energy U of a spring that obeys Hooke's law can be given by the formula U = 1/2 k (x - x0)^2, where k is the spring constant, x is the distance between two molecules i and j and x0 is the equilibrium distance between the interacting molecules. So, the components of F_{int} in x and y direction are given as:

$$F_x = -k\,(x_{ij} - x_0) \qquad \text{where} \quad x_{ij} = x_i - x_j$$

$$F_y = -k\,(y_{ij} - y_0) \qquad \text{where} \quad y_{ij} = y_i - y_j$$

Now putting the value for F_{int} back in Equation (1), we get:

$$\Gamma\, dX/dt = F_{int} + F_{random}$$

Solving this equation further by substituting the values for F_{int} and F_{random} ,we get:

: X_i (t + Δt) - X_i(t) = (-k / Γ)∑ [($x_i(t)$ - $x_j(t)$) - x_0] Δt + (v_0 / Γ)* cos ɵ Δt ……Eq. (5a)

: Y_i (t + Δt) - Y_i(t) = (-k / Γ)∑ [($y_i(t)$ - $y_j(t)$) - y_0] Δt + (v_0 / Γ)* sin ɵ Δt ……Eq. (5b)

The above two equations are used in the code to obtain the motion of multiple interacting protein molecules.

Here k is the bond strength and it scales with the thermal energy k_BT, available in the medium, where k_B is the Boltzmann constant and T is the absolute temperature. We use values of k equal to fraction of k_BT, like 0.1-0.5k_BT, to make sure the bonds can be broken by stochastic movements.

In order to capture the bulk effect, we employ the Periodic boundary condition. In a periodic simulation cell, the positions of particles are repeated periodically in space, creating an infinite lattice of identical cells. The features of PBC ensure that we are simulating a patch of the cellular interface which is much larger than the size of individual proteins.

The implementation of PBC requires that we follow the minimum image convention to compute the interaction between the molecules. Under MIC, the molecules interact with their closest image when they go beyond the boundary of the particle.

When computing the interactions between particles, the conventional approach would be to use the positions of all particles in all cells, leading to a large number of redundant calculations.

By only taking into consideration the periodic image that is closest to each particle, the minimum image convention reduces the number of particle-particle interactions that must be calculated. By subtracting the periodic image of the particle that is the furthest away from the cell of interest, the position of each particle is specifically wrapped into the simulation cell. This reduces the simulation's computational cost while ensuring accurate calculation of distances between particles.

To incorporate the above convention into the code, np.round is used which is a function in the NumPy library of Python that is used to round a given array or scalar to a specified number of decimal places. So,

$$x'_{ij} = x_{ij} - \text{np.round}\,(x_{ij}/\,\text{L})\,\text{L}$$

Where L is the size of our simulation box.

In addition to MIC, we consider a range of interaction for the protein molecules. We assume that the proteins that come within a distance r_{cut} with each other they only interact. This r_{cut} is the cutoff distance between the centres of two protein molecules that allows them to form a harmonic bond in our simulations. We set $r_{cut} = 2x_0$ which makes the protein-protein interaction a short-range attractive potential. This means, the proteins interact only when they are close by, at distances longer than r_{cut} the interaction energy is zero.

C) With endocytosis and exocytosis

We have now a model of active movement of E-cadherin molecules on a membrane patch, where if two proteins are in close proximity, they can form a bond with strength of sub-k_BT and is susceptible to breakage via random movements due to actin-driven transport of the protein molecules.

The next step is to model the effects of internalization of on clustering of proteins.

Since, we are studying the motion of the molecules on a 2D-plane which represents the membrane, it will be difficult to distinguish between endocytosis of E-cad molecules and their degradation. Hence, for simplicity it is assumed that no degradation takes place. Endocytosis will be shown by the disappearance of the molecules from the membrane whereas exocytosis is shown by the appearance of new molecules on the membrane. The rates of appearing and disappearing molecules is not same, which leads to change in the total number of molecules in the system.

So, now to incorporate endocytosis into the code we say that n_d is the number of disappearing molecules and n_a is the number of appearing molecules at any instant. Here molecules are added or removed after every N_t*Δt iterations. We implement these dynamic

processes in the simulation as follows. The numbers of molecules removed (n_d) and added (n_a) are randomly generated using random.gauss, which is a function in the built-in random module of Python that generates Gaussian (normal) distributed random numbers with a specified mean and standard deviation. It has two parameters: μ (mean of the distribution) and σ (standard deviation of the distribution). This way we model both endo- and exocytosis as Gaussian random processes which is a reasonable simplification to develop fundamental understanding. This gives us :

endocytosis rate of $k_{endo} = < n_d > /(N_t \Delta t)$ and

excocytosis rate of $k_{exo} = < n_a > /(N_t \Delta t)$.

In the simulation, we supply the following sets of parameters to the random.gauss function:

$\mu_{endo} = < n_d >$ and $\sigma_{endo} = \sqrt{< n_d^2 > - < n_d >^2}$ and

$\mu_{exo} = < n_a >$ and $\sigma_{exo} = \sqrt{< n_a^2 > - < n_a >^2}$.

We keep $\mu_{endo} \approx \mu_{exo}$ to ensure the number of molecules present in the system at any given time remains nearly unchanged. We use same values for σ_{endo} and σ_{exo}. The advantage of using a Gaussian distribution for describing such random processes is that it is one of the fundamental probability distributions commonly used in scientific and engineering applications, and are often a better model for natural processes than uniformly distributed random numbers.

Chapter 5

RESULT AND DISCUSSION

Since, we have considered the first assumption that the protein molecules are non-interacting, while trying to understand the rules guiding the mechanism for cluster formation in E-cadherin. We try to simulate the motion of a single protein particle whose movement is governed by the two equations obtained in Materials and methods part for motion in a 2D plane.

$$X(t + \Delta t) = X(t) + (v_0 / \Gamma)^* \cos\theta \ \Delta t$$

$$Y(t + \Delta t) = Y(t) + (v_0 / \Gamma)^* \sin\theta \ \Delta t$$

We simulate the motion by writing a code in python. By fixing the time interval Δt to be 0.001 in simulation units, $v_0/\Gamma = 0.4$ and taking the initial coordinates of the particle to be (1,1). Furthermore, we randomly generate the angle θ using NumPy and random in python3. Running the code for 1000 iterations, we get the motion of the particle in an undefined space on 2D plane which represents the cell membrane. The graph showing the motion is:

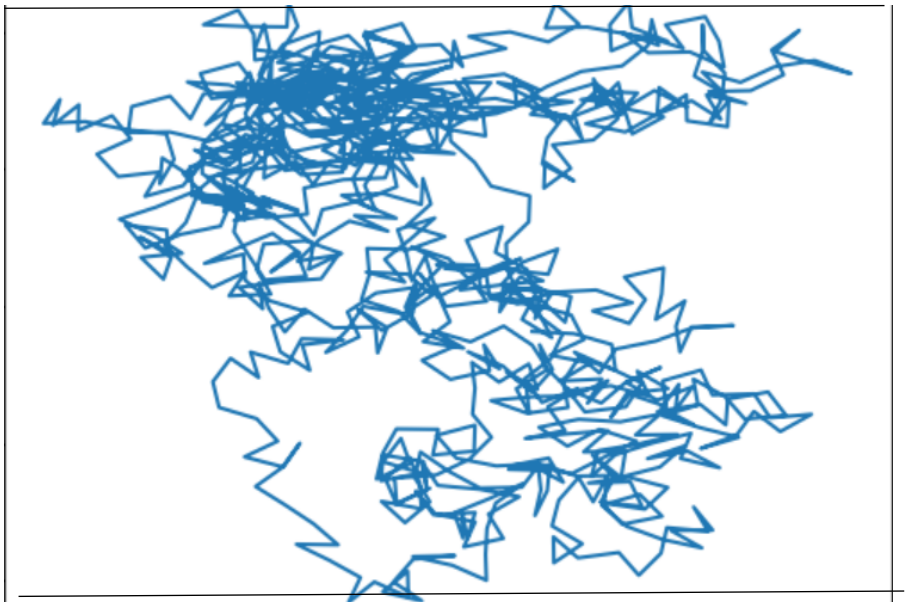


Figure 5.1 : Graph depicting random motion of a single protein molecule.

After this, we look at the motion of multiple non-interacting particles on a defined region boundary. The equations governing the motion of the particle are the same as before. We apply the periodic boundary condition [8]. Applying this condition, it is assumed that a system's physical properties repeat in space, creating an infinite lattice with identical units arranged in a repeating pattern. The technique involves imposing artificial boundaries on a simulation cell, so that particles leaving the cell on one side will re-enter it from the opposite side, as if the cell was repeated infinitely many times in all directions. This ensures we capture the properties of an infinite system. Here, along with the angle, the starting positions of the particle are also randomly generated.

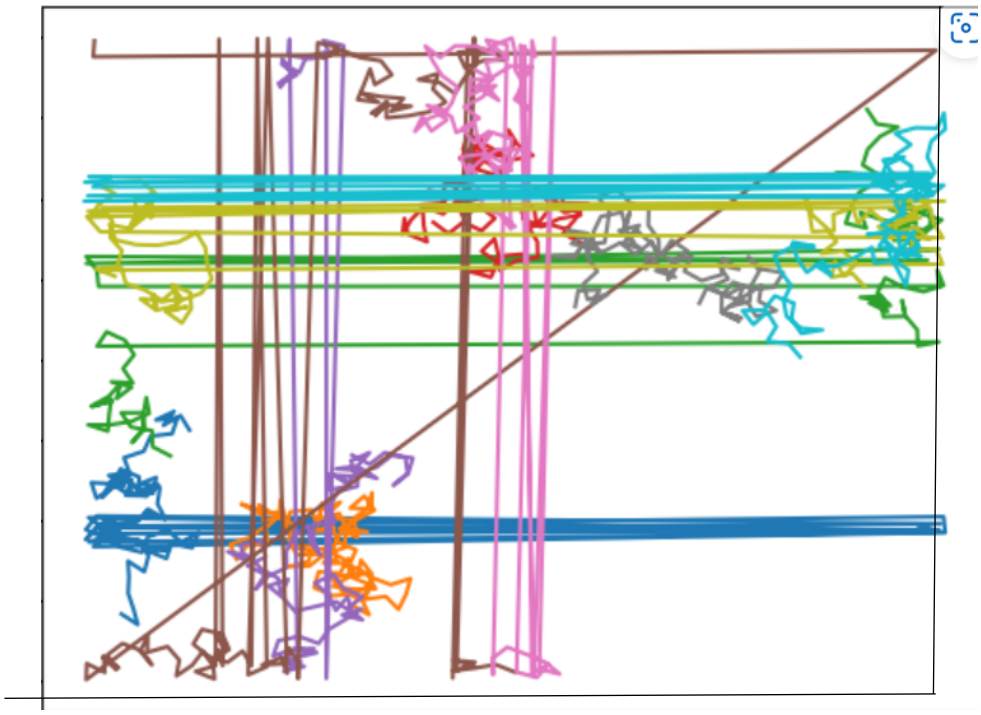


Figure 5.2: Graph depicting random motion of a multiple protein molecule under PBC.

We see the random motion of 10 particles, among which the particle movement shown by the straight lines depicts the periodic boundary condition applied in the code in effect.

5.A) Clustering with interaction

We further proceed by considering the protein molecules to be interacting with each other. We try to simulate the motion of a multiple protein particle whose movement is governed by Eq. 5a and 5b. The motion is simulated by using the python code provide in the appendix. We set values of k/Γ written simply as K and v_0 / Γ written simply as v. The code is run and the images obtained after ever 50 iterations are combined to form a video showing clustering of the molecules due to their interaction with each other. The simulation is run for different set of values of K and v from the following sets : v = (0.4,1.0,1.6) and K = (0.3,0.5,0.7,1.0). Other important parameters in the simulation are: number of molecules N = 100 and size of the square simulation box L = 20 times the average diameter, r of individual protein molecules.

It is observed that as the value of K increases for a constant v the formation of clusters is more and the clusters are more tightly packed whereas as the value of v increases for a constant and low K the cluster formation is disrupted as the molecules move at a higher speed leading to formation of less clusters.

5.B) Clustering in presence of endo- and exocytosis

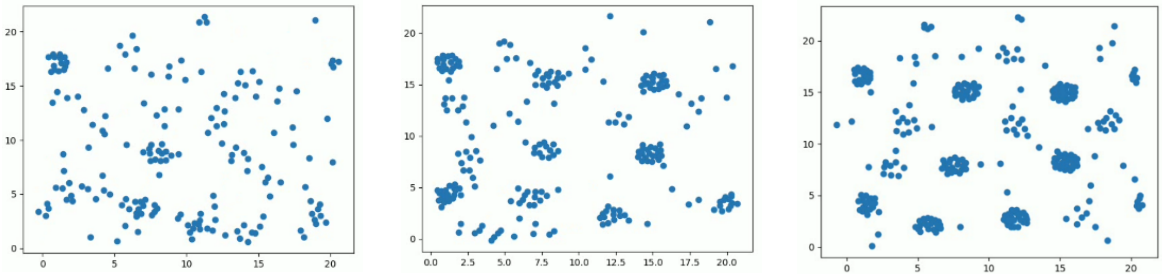
Now, we proceed by incorporating the number of appearing and disappearing molecules after a set of iterations into the code using the random.gauss function. We run the simulations for 10^5 steps and collect snapshots after 50000 steps. The endo- and exocytosis process is performed every 10 iterations.

It is observed that with endocytosis the clusters formed are dynamic and the cluster size is variable. These clusters of E-cadherin molecules formed act as adhesion patches on the cellular membrane which further interact with another set of E-cadherin clusters on the cell membrane surface of another cell. This eventually results in the adhesion of those two cells. Hence, the clusters act as a platform for cell-cell adhesion. Whereas without endocytosis, it is

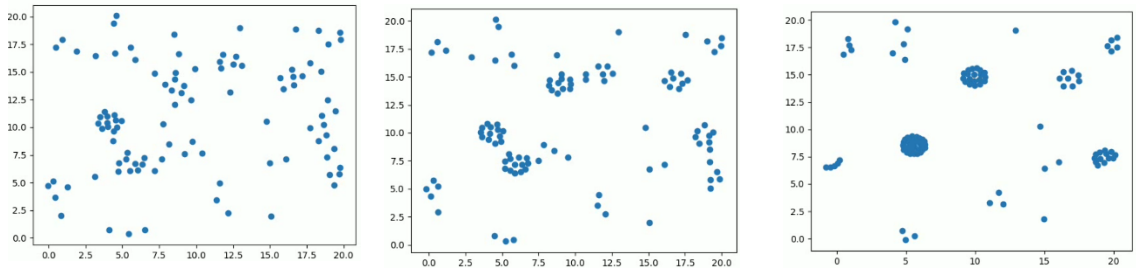
observed that the clusters formed are more localized and the cluster size is limited, not dynamic.

Case I] For v = 0.3 and k = 0.1 :

With endocytosis:

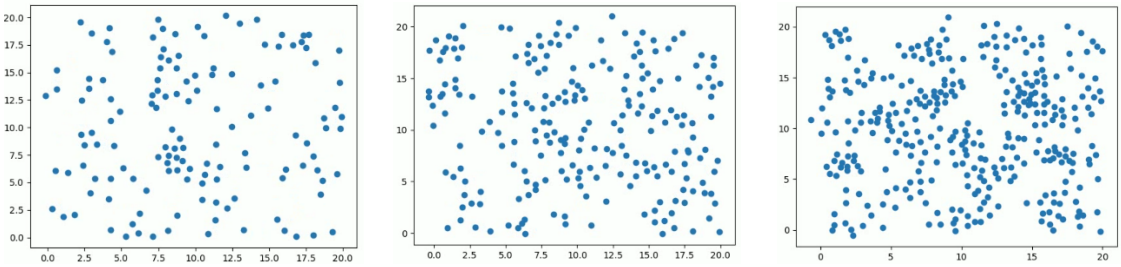


Without endocytosis:

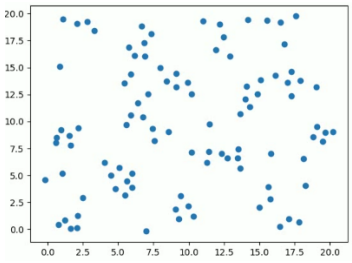
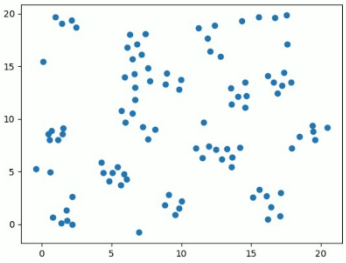
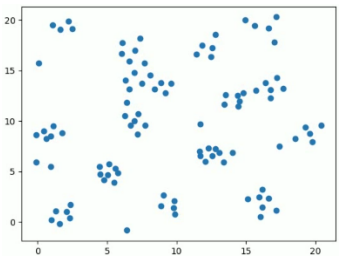


Case II] For v = 1.0 and k = 0.1 :

With endocytosis:

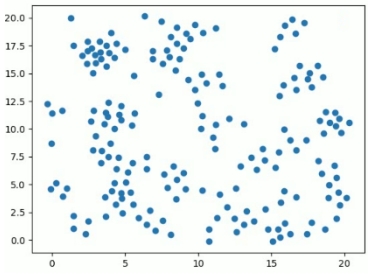
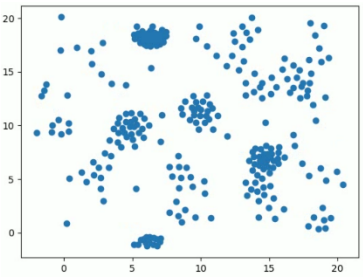
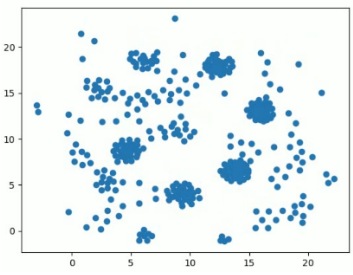


Without endocytosis:

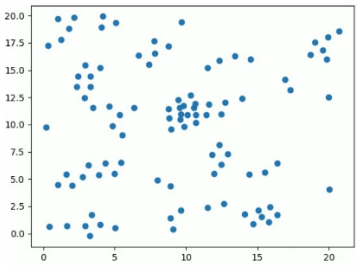
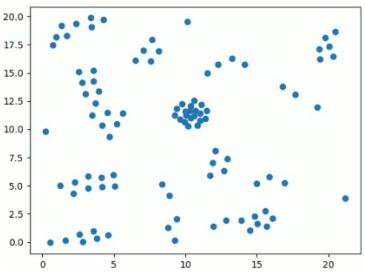
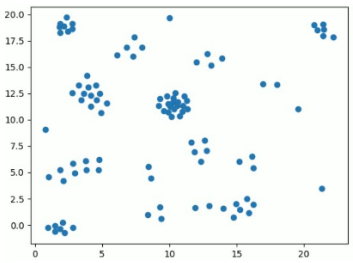




Case III] For v = 1.0 and k = 1.0 :

With endocytosis:

Without endocytosis:

Case I and II] By keeping the k constant (k=0.1) and changing the values of v (v= 0.3, 1.0), it is observed that: With and without endocytosis, the clustering is more prominent for lower value of v, whereas as v increases the molecules are not able to stick together for a longer time leading to constant breaking and forming of clusters which is evident as no clear clusters can be seen for higher v.

Case II and III] By keeping the v constant (v=1.0) and changing the value of k (k=0.1,1.0), it is observed that: With and without endocytosis, more clusters are formed for higher value of k whereas for the lower value of k, the bond strength and elasticity is reduced due to which molecules are not able to stay together as clusters over time.

Chapter 6

SUMMARY AND CONCLUSION

The physical laws governing the mechanism for E-cadherin cluster formation are not clearly understood. Therefore, to proceed further we assume model interaction in E-cadherin molecules. Initially we started with non-interacting protein molecule randomly moving on the cellular membrane. Here, we simulated this non-interacting protein dynamics using Langevin dynamics simulation.

Further we assume a harmonic spring-like interaction between two E-cadherin molecules with a finite range of interaction of about two molecular diameters. This, however, is an over-simplification which we will relax in future. We plan to consider more intricate interaction potentials among E-cadherin molecules focusing to capture the more realistic clustering behaviour of E-cadherin in cells.

We analyse the effective interaction arising from active movement of E-cadherin molecules. A mechanism by which E-cadherin molecules can move within cells is through active transport by motor proteins such as myosin. Myosin is a molecular motor protein that moves along actin filaments and can transport cargo molecules, including E-cadherin, to different locations within the cell. This type of active transport may be important for the formation and maintenance of E-cadherin clusters.

Finally we study the effects of endo- and exocytosis of clustering of E-cadherin which has recently been the topic of very interesting research in various contexts like development of organisms like Fruit fly [1], [9] and cancer progression [3]. We considered a very simple

Gaussian approximation of these two important and fundamental biological processes. In future, we plan to include more complex dynamical features.

Our simple study reveals that the clustering of proteins at cellular interfaces competes strongly with active transport by f-actin and endocytosis. This competition gives rise to strongly dynamical features that need more detailed and long-term studies which we plan to take up in future.

Chapter 7

REFERENCES

1) Binh-An Truong Quang, Madhav Mani, Olga Markova,Thomas Lecuit, and Pierre-Francois Lenne – "Principles of E-Cadherin Supramolecular Organization In Vivo", Current Biology 23, pg. 2197-2207, November 18, 2013
2) Danfeng Cai, Shann-Ching Chen, Mohit Prasad, Li He, Xiaobo Wang, Valerie Choesmel-Cadamuro, Jessica K. Sawyer, Gaudenz Danuser, "Mechanical Feedback through E-Cadherin Promotes Direction Sensing during Collective Cell Migration", Cell, Volume 157, issue 5, May 2014, pg.1146-1159
3) Chiara Malinverno. et.al, "Endocytic reawakening of motility in jammed epithelia", Nature materials 16, 587, January 2017
4) Robert J. Tetley and Yanlan Mao, "The same but different: cell intercalation as a driver of tissue deformation and fluidity", Philosophical transactions of the royal society, Biological Sciences 373, September 2018
5) Veena Padmanaban, Ilona Krol, Yasir Suhail, Barbara M. Szczerba, Nicola Aceto, Joel S. Bader & Andrew J. Ewald, "E-cadherin is required for metastasis in multiple models of breast cancer", Nature 573, pages: 439–444 , September 2019
6) Diego A. Vargas, Hans Van Oosterwyck, "Cell Adhesion: Basic Principles and Computational Modeling", Encyclopedia of biomedical engineering, pg: 45-58, 2019
7) Yang Chen,Julia Brasch,Oliver J. Harrison and Tamara C. Bidone, "Computational model of E-cadherin clustering under force", Biophysical Journal 120, pg: 4944–4954, November 2021

8) Understnading molecular simulations --- from algorithms to applications, by Daan Frenkel and Berend Smit.
9) Timothy E Vanderleest, Celia M Smits, Yi Xie, Cayla E Jewett, J Todd Blankenship , Dinah Loerke , University of Denver, US, "Vertex sliding drives intercalation by radial coupling of adhesion and actomyosin networks during Drosophila germband extension", Computational and systems biology, developmental biology, July 2018
10) Guillaume Charras, Alpha S Yap, "Tensile Forces and Mechano-transduction at Cell-Cell Junctions", Current Biology, 28(8):R445-R457, April 2018
11) Amit Das, Abrar Bhat, Rastko Sknepnek, Darius Köster, Satyajit Mayor, Madan Rao, "Stratification relieves constraints from steric hindrance in the generation of compact actomyosin asters at the membrane cortex", Science Advances. 6, eaay6093 (2020).
12) Collins C, Denisin AK, Pruitt BL, Nelson WJ. Changes in E-cadherin rigidity sensing regulate cell adhesion. Proceedings of the National Academy of Sciences. 2017 Jul 18;114(29):E5835-44.

APPENDICES

A) Dimensional Analysis

To implement the motion of a molecule by writing a Python code, it was essential to assess the dimensionality of each parameter being used. Equations of motion for each particle are frequently solved numerically when computer simulations are used to model the motion of molecules. We use numerical values to represent physical quantities which have dimensions like each particle's position, velocity, and acceleration in such simulations. This means that when we combine physical quantities in our code, we need to be careful because they may have different dimensions, which could result in wrong or inconsistent results.

We can use natural units to make everything dimensionless to solve this problem. Physical quantities are represented as dimensionless ratios of one quantity to another in these units.

To integrate the motion equations in molecular simulations, we must establish a time scale. To define a time scale we need the fastest motion present inside the framework. This is because the time it takes for the fastest motion to occur defines a system's time scale. For instance, the thermal motion of the particles typically represents the fastest motion in a gas of molecules. The thermal energy of the molecules, which is proportional to the system's temperature, is what causes this random motion. The molecules move more quickly and with more thermal energy as the temperature rises. Our simulations' outcomes are easier to understand and analyse as a result of this.

We make each term in the equations of motion Eq. (5a) and (5b) dimension-less using appropriate factors as units. We use the average diameter of every protein molecule, r as the unit of length, k_BT as unit of energy and $\tau = \Gamma r^2/k_BT$ as the unit of time. Eq. 5a with rescaled terms is given below:

$$X_i^*(t + \Delta t) - X_i^*(t) = -k^* \sum_j [(X_i^*(t) - X_j^*(t)) - x_0^*]\,\Delta t^* + v_0^* \cos\theta\ \Delta t^*$$

Where $X^* = X/r,\ k^* = kr^2/k_BT$, and $v_0^* = v_0 r/k_BT$.

Using real values of $r \sim 6\ nm$ [1], and viscosity of the membrane, η_{mem} as 100 times more than the viscosity of water [11]. we get the friction coefficient $\Gamma = 4\pi\eta_{mem} r \approx 0.01\ pN.s.\mu m^{-1}$. Our simulations are done with temperature of a live cell T = 310K. This makes the time unit of our simulation, $\tau \approx 90\ \mu s$. Therefore, $v_0^* = 1$, corresponds to a speed of a free protein molecule of about 0.6 nm per time interval τ. In this units, the membrane patch we simulate is L = 120 nm. The density of E-cadherin in our simulations is = N/L^2 = 7000 molecules per sq micron which is few times greater density of E-cadherin than actual cell membrane. E-cadherin density in late stage fruit fly embryo ~ 1000 molecules per sq micron (Ref. 1) whereas in MDCK cells with mature adhesions this number can be up to 2500-5000 molecules per sq micron (Ref. 12). We use such higher density for collecting better statistics on the clustering in a smaller system. In future, we will use more realistic densities when we consider a bigger system.

B) Python program

Below written is the python program including endocytosis of E-cadherin molecules :

```
import numpy as np
import random
import matplotlib.pyplot as plt
import cv2

print("Generating random value for V between 0 and 1 ")

del_t = 0.01
L=20.0
v = 1.0
print("V = "+str(v)+" generated")
#r = 0.00005
k = 1.0
arr = []

mu_d = 10
mu_a = 12
sigma = 5
n_part = 100



n_iter = int(input("Enter the number of iterations :"))
vecx = [[] for i in range(n_part)]
vecy = [[] for i in range(n_part)]
for i in range(n_part):
    x = random.random()*L
    y = random.random()*L
    vecx[i].append(x)
    vecy[i].append(y)
    arr.append((x,y));

endo_rate = 10
exo_rate = 12
sigma_rate = 5
freq = 100
l0=1.1  # touching distance of molecules
img_names = []
for t in range(n_iter):
    # print("vecx  is "+str(vecx))
    # print("vecy  is "+str(vecy))
    if t%freq==0:
        # print("vecx  is "+str(vecx))
        # print("vecy  is "+str(vecy))
        fig, ax = plt.subplots()
        to_printx =[]
        to_printy =[]
        for i in range(n_part):
            to_printx.append(vecx[i][-1])
            to_printy.append(vecy[i][-1])
        ax.plot(to_printx, to_printy,'o')
        fig.savefig(f"graph_{t}.png")
        plt.close(fig)
```

```
        img_names.append(f"graph_{t}.png")



        # removing molecules
        nd = int(np.round(random.gauss(endo_rate,sigma_rate)))
        while(nd>=n_part):
            nd = int(np.round(random.gauss(endo_rate,sigma_rate)))

        removed_ind = set()
        while(len(removed_ind)<nd):
            x = random.randint(0, n_part-1)
            removed_ind.add(x)

        new_indices = []
        for i in range(n_part):
            if i not in removed_ind:
                new_indices.append(i)
        tvecx=[]
        tvecy = []
        tarr=[]
        for ind in new_indices:
            tvecx.append(vecx[ind])
            tvecy.append(vecy[ind])
            tarr.append(arr[ind])
        vecx = tvecx
        vecy = tvecy
        arr=tarr
        # adding molecules
        na = int(np.round(random.gauss(exo_rate,sigma_rate)))
        for i in range(na):
            x = random.random()*L
            y = random.random()*L
            vecx.append([x])
            vecy.append([y])
            arr.append((x,y))

        n_part = len(vecx)


    fxij = np.zeros(n_part)
    fyij = np.zeros(n_part)
    # print(n_part)
    # print("size of vecx is "+str(len(vecx)))
    # print("size of arr is "+str(len(arr)))
    # print()
    for i in range(n_part-1):
        for j in range(i+1,n_part):
            (xi,yi)=arr[i]
            (xj,yj)=arr[j]
            delxij = (xi-xj)
            delyij = (yi-yj)
            delxij = delxij - np.round(delxij/L)*L
            delyij = delyij - np.round(delyij/L)*L
            lij = np.sqrt(delxij**2.0 + delyij**2.0)
            if lij <= 2.0*l0:
                fij = -k*(lij-l0)
                fxij[i] += fij*delxij/abs(lij)
```

```
                    fxij[j] -= fij*delxij/abs(lij)
                    fyij[i] += fij*delyij/abs(lij)
                    fyij[j] -= fij*delyij/abs(lij)

        for i in range(n_part):
            (x,y)=arr[i]
            valid = False
            while(not valid):
                temp = random.random()
                theta = 2*np.pi*temp
                x = x + (fxij[i] + v * np.cos(theta)) * del_t
                y = y + (fyij[i] + v * np.sin(theta)) * del_t
            #    if(x<0):
            #        x+=L
            #    if(x>L):
            #        x-=L
            #    if(y<0):
            #        y+=L
            #    if(y>L):
            #        y-=L
                temp=True
                valid=True
            vecx[i].append(x)
            vecy[i].append(y)
            arr[i]=(x,y)

print("pre-processing done!")
# Set up video codec and output file name
codec = cv2.VideoWriter_fourcc(*"mp4v")
out = cv2.VideoWriter("output.mp4", codec, 18, (640, 480))



for i in range(len(img_names)):
    # Read the saved image file and add it to the video
    img = cv2.imread(img_names[i])
    out.write(img)

plt.show()


# Release the video writer and cleanup
out.release()
cv2.destroyAllWindows()

print("done!")
```