

Asymmetry in lipid dynamics induced by transmembrane cytolsin pore

Sayantan Dutta^a, Rajat Desikan^b, K.G.Ayappa^b, and J.K.Basu^c

^a Department of Chemical Engineering, Indian Institute of Technology Kharagpur

^b Department of Chemical Engineering, Indian Institute of Science, Bangalore,

^c Department of Physics, Indian Institute of Science, Bangalore,

ABSTRACT

Bacterial pore forming toxins (PFTs) are virulence factors that penetrate through the host cell membrane and cause cell lysis. This study focuses on the in-silico observation of the dynamics and structure of the lipid proximal to a transmembrane pore of the alpha-helical PFT, Cytolysin A (ClyA). We observe that the lipids near the protein in the upper leaflet become slow, whereas lipids become disordered and lose orientation in the lower leaflet. Lipids more than 2-3 nm away from the protein surface remains unaffected. These observations have implications on our understanding of the effect of proteins on membrane dynamics.

Keywords: Molecular Dynamics, DMPC, ClyA

1. INTRODUCTION

Pore forming toxins (PFT) act as one of the most offensive natural biological weapons, and are used by prokaryotes such as bacteria to higher order eukaryotes such as poisonous reptiles. They function by making pores across the cell membrane to destroy the permeability barrier ultimately leading to cell death. Though these pore forming toxins are expressed in a water soluble form, after membrane insertion they go through a conformational change to penetrate through the lipid bilayer [1]. So, the change in lipid dynamics due to introduction of these toxins is an area of interest. As the pores formed by these toxins are of 2-50 nm diameter [2], it is difficult to resolve the dynamics of lipids around and far from it with available experimental techniques. An alternative is molecular dynamics simulations, which can give us the trajectory of every atom with a femtosecond time resolution and Angstrom level spatial resolution.

Cytolysin A (ClyA) is one of the pore forming toxins with an alpha helical structure, which are produced by *Escherichia coli* and *Salmonella enterica* [2]. The detailed structure of the protein ClyA is available in Protein Data Bank (PDB) (ID: 2WCD). Another interesting feature of the ClyA cytotoxin is that though it has hollow cylindrical shape in upper side it becomes truncated cone like in its lower leaflet.

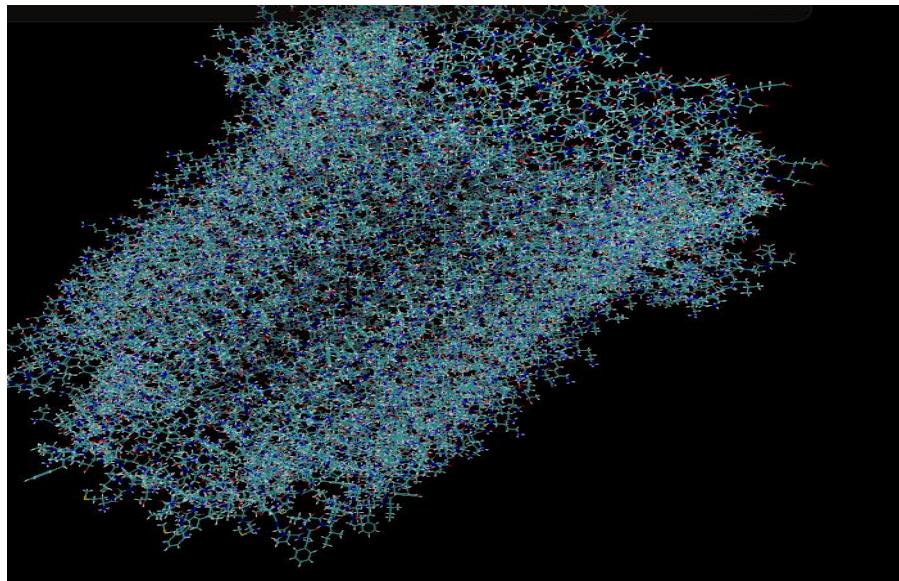


Figure 1. VMD visualisation of the ClyA pore.

Dimyristoylphosphatidylcholine (DMPC) is often used among lipids for molecular dynamics studies. The thickness of the DMPC bilayer matches closely with the length of hydrophobic patch of the ClyA pore. That is why it is chosen to study the interaction of ClyA with the bilayer. Niemela et al. [3] have studied the effect of membrane protein Lac Y on the dynamics of POPC molecules, and observed the lipids to be slow near protein. But Lac Y neither forms a pore nor has a leaflet-wise asymmetrical structure like ClyA.

Initially the study was taken for computational observation of pore forming mechanism of ClyA oligomers [4], which is the first computational evidence of growing pore mechanism as suggested earlier [5]. Now from these set of simulations some (Bare membrane, membrane with a 9-mer incomplete pore inserted, and membrane with a 12-mer complete pore inserted) were extended to longer time scales to study the change and asymmetry in the dynamics of the lipid induced by the toxin.

2. OBJECTIVE OF THE PROJECT

The study aims at:

- 1) Observing the variation in radial mobility of the lipids in the bilayer.
- 2) Observing the variation in orientation of the lipids in the bilayer.
- 3) Observing the variation of phase (i.e. ordered and disordered) of the lipids in the bilayer.
- 4) Observing the change in diffusive property of the lipids after the introduction of the protein.

Above the word “variation” refers to asymmetry in radial direction as well as across leaflet. In case of the 9-mer protein, lipids inside the pore, outside the pore, in front of the protein and at back of the protein were studied separately.

3. THEORY AND METHODS OF CALCULATION

3.1. Finding the Centre of Curvature of Pore and Pore Size:

As we are interested in radial variation of dynamics and orientation of lipid properties we need to find the geometrical centre of the pores to draw cylindrical or hemicylindrical shells. As in case of 9-mer the circle is incomplete, the centre of mass will be biased and will not be a proper representation of the protein's centre of curvature. So, drawing a circle through the centre of mass of all the individual protomers in the 9-mer arc is a better approximation. Equation of the circle is given as:

$$(x-a)^2 + (y-b)^2 = r^2 \quad (1)$$

Now this equation has 3 unknowns a, b and r (x and y coordinate of the centre and radius of the circle). So, every three equations (from coordinate of centre of mass of every 3 protomers) will give a solution. But we have 9 units for 9-mer and 12 units for 12-mer. So, to deal with this over-definition of the system, the above equation is solved for every 3 combinations of the protomers to find a, b and r. Then all the solutions were averaged. To, check the correctness of the system, standard deviation of the solutions are also calculated and radius of curvature is plotted with this standard deviation. In case of the centre of curvature coordinates the standard deviations were below 0.2 nm. As for the further calculations we divided the box into shells with thickness of 0.5 nm, it should not affect the calculations. Sample code (for 9-mer) is given in Appendix A for this calculation.

3.2. Allocating the Radial Shell for all the Lipids

As we are looking into the spatial variation of the properties of lipid molecules we need to divide all the lipids in different radial shells according to the position of the lipid as well as upper and lower leaflets.

For the 9-mer, the protein centre of curvature as calculated above are treated as the centre of all shells. As the protein itself is not a complete circle, expecting angular asymmetry we need to distinguish between lipids in front and back of the protein. For each timestep the distance of the lipid molecules were calculated from the centre and allocated a shell. Then to check it is in front of the lipid or at the back an indirect method was followed. By manual investigation, two amino acids (Glu11 of chain G and Pro189 of chain A) were identified in the edge protomers of the arc formed by the protein. Now the equation of the chord passing through these two points can be found, and a diameter parallel to the chord was found from the equation of the line passing through the centre with a same slope as that of the chord, with the centre located at the back side of the chord. Following this principle we differentiated the lipids in the front and the back of the lipid, and a schematic is given in appendix-F.

For 12-mer, the shells were distributed only on the basis of the distance from the protein centre. As it is a complete circle there is no angular asymmetry in between the lipids. For the bare membrane, there was no protein. So, the shell was distributed just on basis of the two dimensional distance from the centre of the box. There is no angular asymmetry here either. Appendices B, C, and D contain sample code used for this calculation for lipids around 9-mer.

3.3. Calculation of Survival Probability as a function of time:

Survival Probability of molecules present inside a given dimension is defined as in [7]

$$S(t) = \sum_{i=1}^n \prod_{t=t_0}^{t=t_0+t} P_i \quad (2)$$

Where n=No of lipids, t_0 =Initial timestep, t =time
 $P_i=1$ if i^{th} lipid is in its parent shell
 $=0$ (Otherwise)

In this problem the dimension of interest is the cylindrical or the hemi-cylindrical shell radially drawn from protein's centre of curvature. All the lipids were allocated with an instantaneous value of $P_i=1$ at the initial time-step, and P_i was made to 0 as soon as it comes out of the shell at which it belonged to in the initial timestep. Now, all these probabilities were averaged to get the survival probability of lipids at a particular timestep for a specific shell. These survival probabilities were averaged for all the initial time-steps till half the length of the simulation. The survival probabilities of all the shells were fitted into a double exponential function of time, as shown in the equation below, in MATLAB following Levenberg-Marquardt algorithm.

$$S(t) = A * \exp\left(\frac{-t}{\tau_f}\right) + (1 - a) * \exp\left(\frac{-t}{\tau_s}\right) \quad (3)$$

The constants τ_f and τ_s give an idea about the radial motion of the lipid. As the lipids are moving across the shells, other functions to calculate diffusivity such as mean square displacement doesn't give us the idea of diffusivity as a function of space because a particular shell cannot be allocated to randomly moving lipids throughout the time we are observing. So, this way is used to show the radial variation of the mobility of the lipids as a function of time. The time-constants got from above fitting may be compared to the time-constants experimentally got from the FCS (Fluorescence Correlation Spectroscopy). So, higher the time constants, slower the dynamics. Appendix B contains sample code used in case of calculation of SP for lipids around the 9-mer.

3.4. Order Parameter

Nematic Order Parameter is defined as

$$O.P = 0.5 * (3\cos^2\theta - 1) \quad (4)$$

Where θ is the angle between a particular bond vector with respect to a particular axis. The value of $\cos\theta$ is calculated by Equation (5).

$$\cos\theta = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}| \cdot |\vec{B}|} \quad (5)$$

Here we are calculating two types of order parameters- tail to head order parameter to observe the orientation of the lipid, and S_{CD} order parameter to observe the disorder of the system.

3.4.1. Tail to head order Parameter

This parameter is calculated by using the angle between tail to head vector of the lipid and membrane normal (i.e. Z axis). Tail to head vector is defined as the average between the vectors from P atom at the head of the lipid to the two C atoms at the tail of the lipid (i.e. C14 atom at two tails). A completely straight lipid makes an angle of 0 or 180 degrees with Z axis which leads to the value of order parameter to 1 whereas a lipid perpendicular to Z axis will make the value of order parameter =-0.5. So, more the value is closer to 1 straighter the lipid is. Appendix C contains sample code used in case of calculation of lipids around 9-mer.

3.4.2. S_{CD} order parameter

This parameter is often calculated in NMR experiments to describe the order of a lipid [8]. In experiments it is calculated by using the angle between C-D bond and membrane normal. In case of simulation the same thing is mimicked by calculating over all C-H bonds attached to a particular carbon, averaged over all lipids in a particular shell, and over all time-steps as suggested [9]. In the case of a completely disordered random system, the average value of order parameter is close to zero whereas when there is some order in the system, this value goes either to 1 or -0.5 depending on the orientation of the system. So, if the order parameter value is far from zero either with positive or negative sign, the system has lost its disorder and becomes less random. Appendix D contains sample code used in case of calculation of lipids around 9-mer.

3.5. Mean Square Deviation (M.S.D)

Mean Square deviation is defined as

$$M.S.D(t) = \left| \overline{\vec{r}(t_0 + t)} - \overline{\vec{r}(t_0)} \right|^2 >. \quad (6)$$

Where $\left| \overline{\vec{r}(t)} \right|$ refers to position vector of a species at time t, and t_0 suggests the initial time-step.

In our calculations, M.S.D. for all the lipids is calculated by shifted time origin averaging till half of the total length of simulation, by taking the periodic boundary unwrapped coordinates of the phosphate atoms of the lipids. The sample code used for calculation in bare membrane is given in appendix E

Einstein's equation relates the M.S.D. equation with the diffusion coefficient as follows.c

$$M.S.D.(t) = 6Dt \quad (7)$$

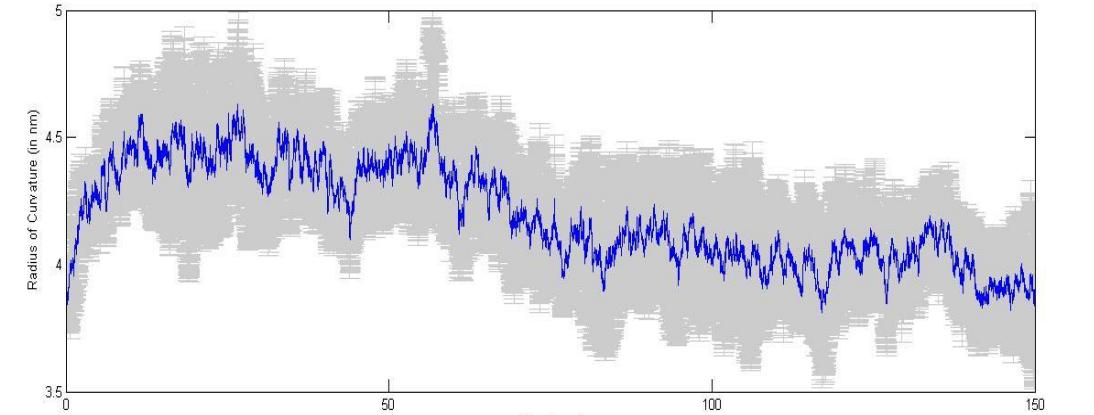
But if the M.S.D. vs. t curve is not linear (as most of the lipids in our case) and then depending on the value of α in Equation (8), the system is called subdiffusive ($\alpha < 1$) or superdiffusive ($\alpha > 1$).

M.S.D. of all the lipids is fitted in a function of time as to find the distribution of alpha before and after insertion of protein.

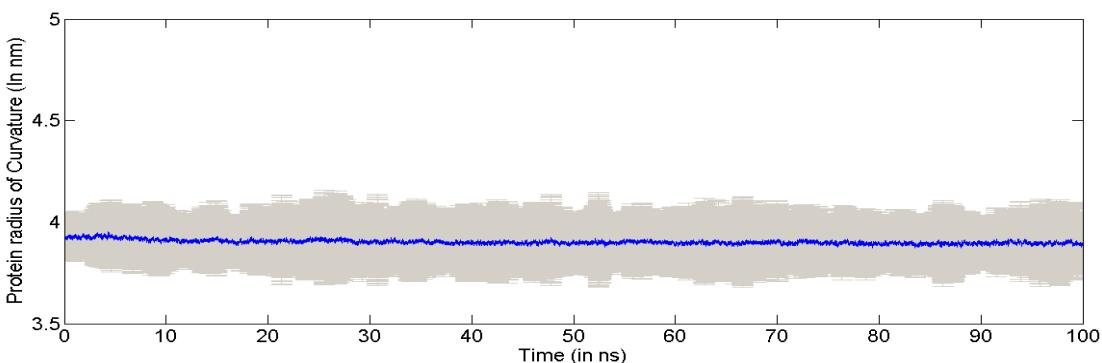
$$\text{M.S.D. (t)} \propto ct^\alpha \quad (8)$$

4. RESULTS AND DISCUSSION

4.1. Variation of Protein Radius of Curvature:



(a) 9-mer



(b) 12-mer

Figure2: Protein radius of curvature as a function of time

Figure 2 illustrates that in case of 9-mer the protein radius is fluctuating more than that of 12-mer. The error bar in this curve refers to the standard deviation due to the averaging of different combination of points. The radius thus obtained will help us to distinguish the lipids inside and outside the pore and lipids near the protein surface. These deviations through sampling of points and fluctuations over time should not affect the result as it is within the shell size taken by us.

4.2. Survival Probability Calculations:

As discussed earlier survival probabilities were calculated to describe the radial motion

of the lipids across shells. Two sample plots are shown below.

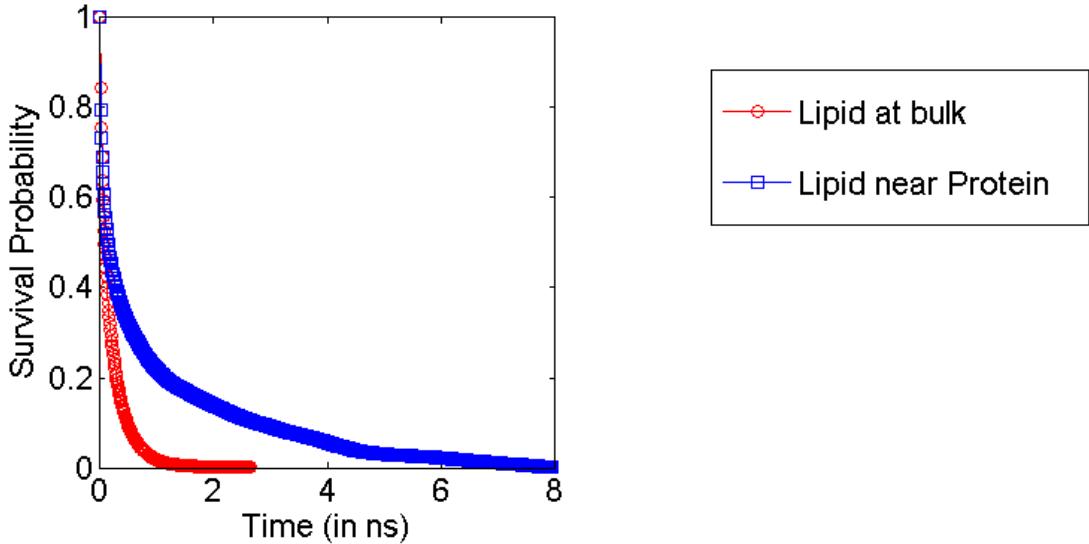
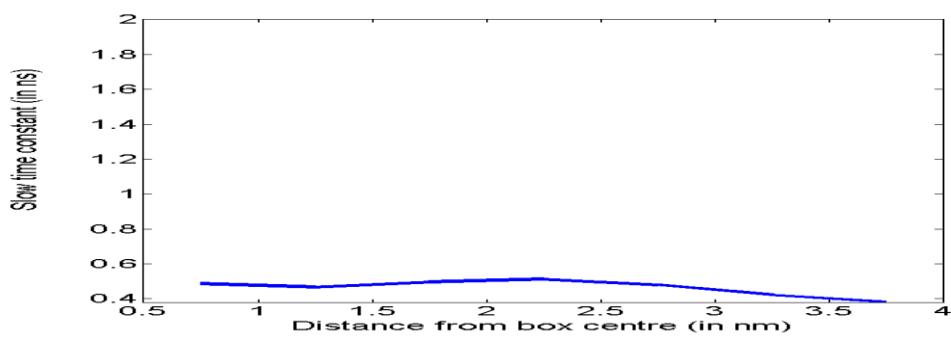
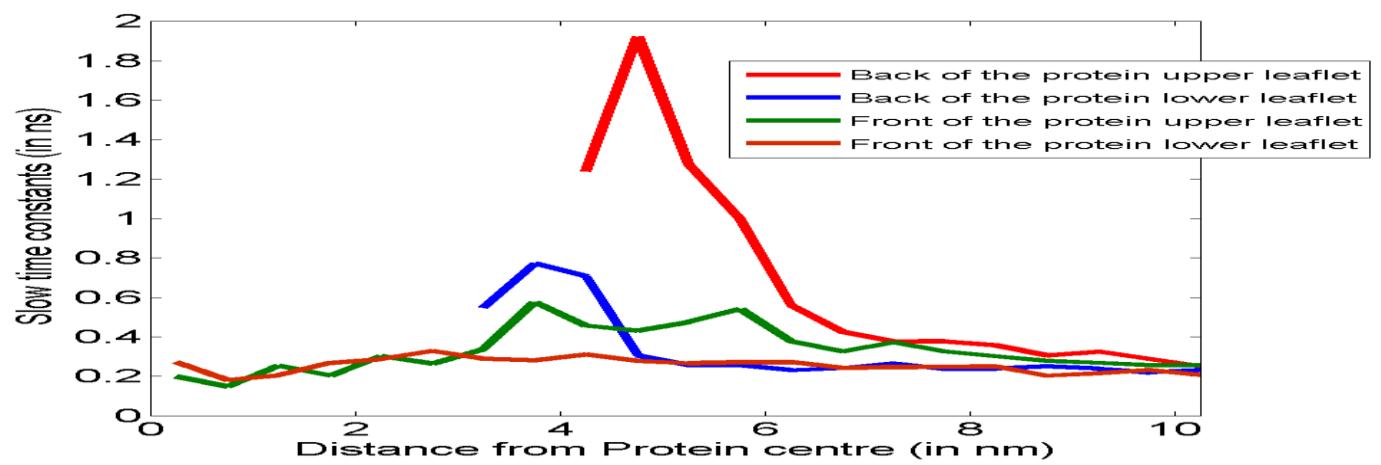


Figure 3 .Survival Probability as a function of time shown for lipids at the back of 9-mer

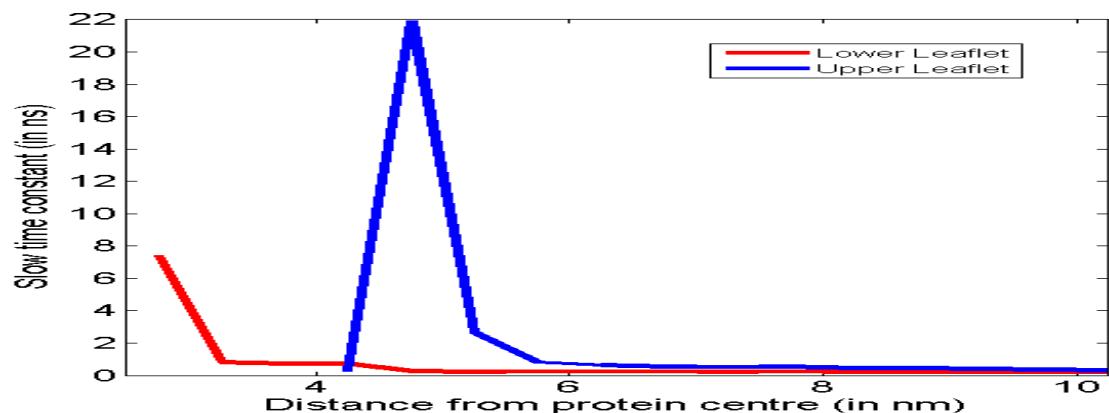
Figures 4 and 5 depict the variation of these time constants across the shell in the case of bare membrane and after insertion of 9-mer (incomplete pore) and 12-mer (complete pore). Both the time constants are almost constant, suggesting a uniform motion throughout the simulation box in case of bare membrane. Whereas in case of 9-mer we see there are peaks in the time constants near the protein surface. These peaks are much higher in case of lipid molecules at back of the protein than in front of the protein indicating a role of the protein surface in slowness of the lipids. The lipids in the upper leaflet are also slower in comparison to those in the lower leaflet. After insertion of 12-mer lipids near protein become almost 10 times more slower in terms of slow time constant near the protein with respect to the back of 9-mer. But in both the case of the 9-mer and 12-mer lipids in the bulk (far away from the protein) reach smaller values of time constants of bare membrane are faster. Studies on membrane proteins [3] also show similar slowing down of lipid near the protein, whereas lipids in the bulk membrane are faster. Almost similar trends are observed both in case of slow and fast time constants. Additional features observed in the slow time constant graphs are shifted peak in upper leaflet at back of the protein and lower time constant inside the pore in case of 9-mer. Second feature is observed due to evacuation of lipid from the pore



(a) Bare Membrane



(b) 9-mer



(c) 12-mer

Figure 4. Variation of slow time constant as a function of distance from the centre of the protein

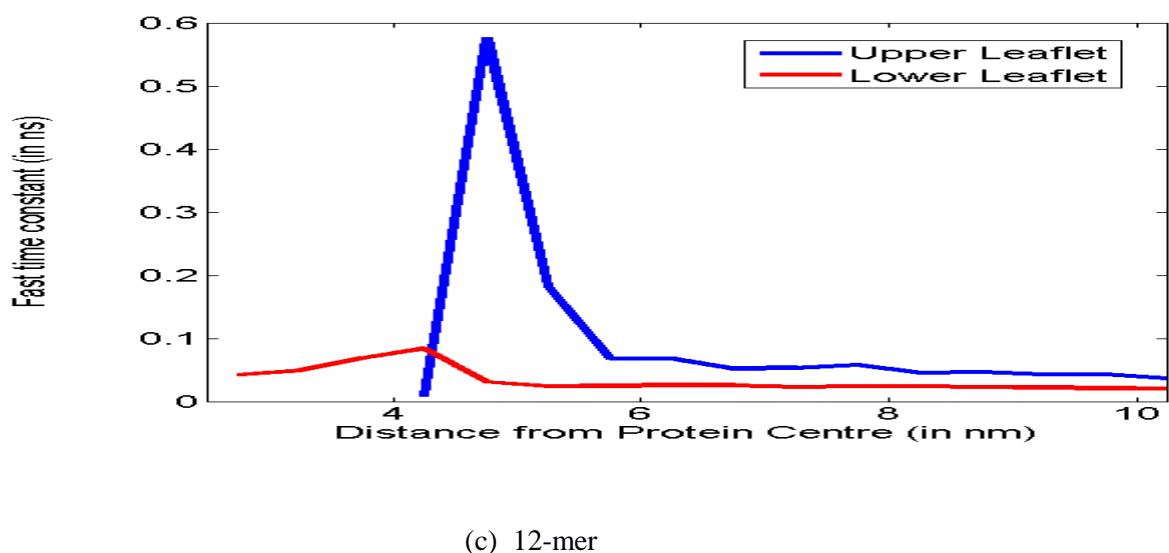
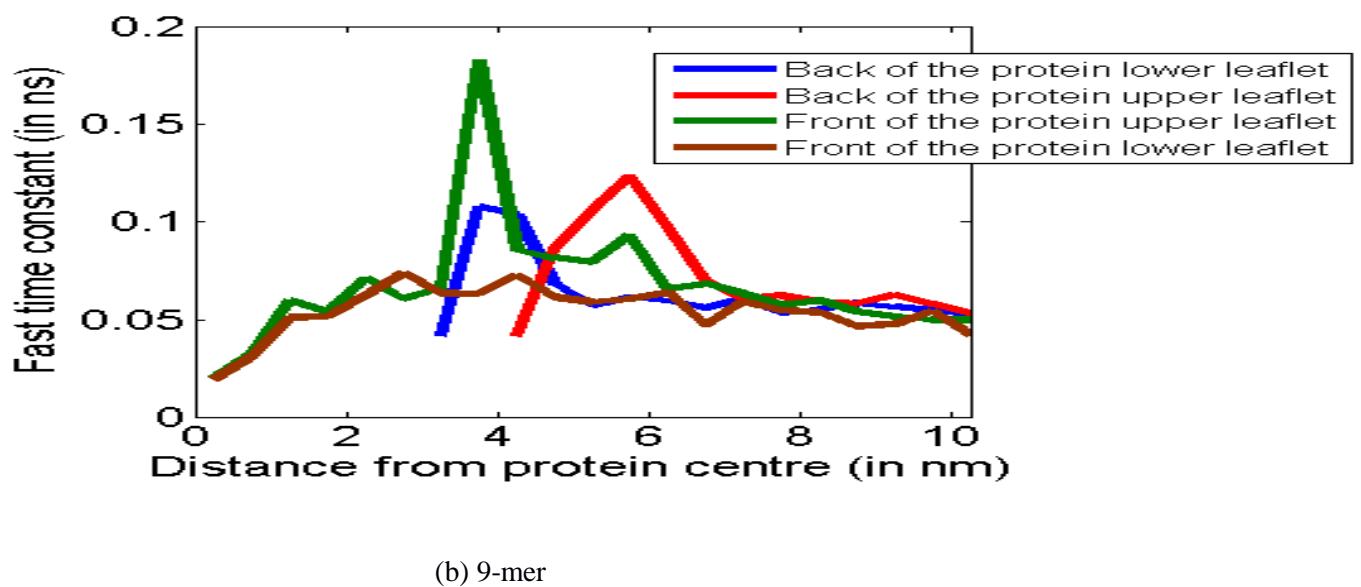
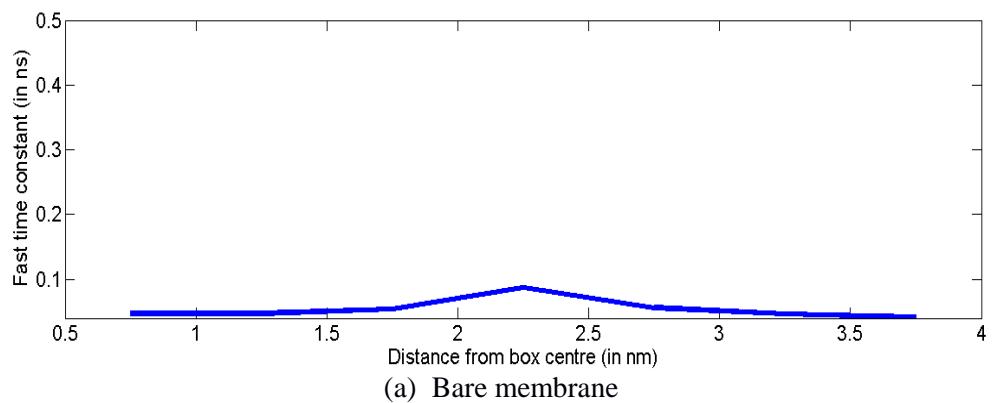
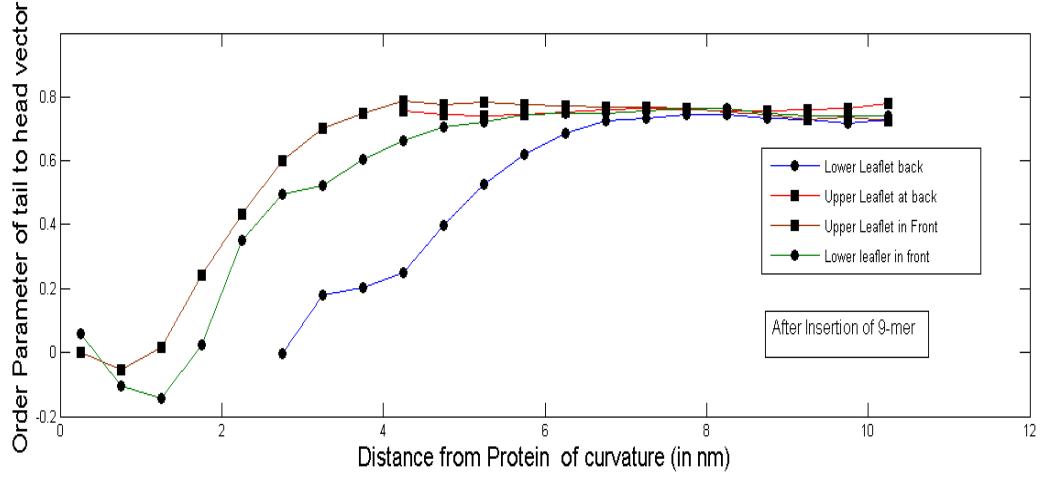


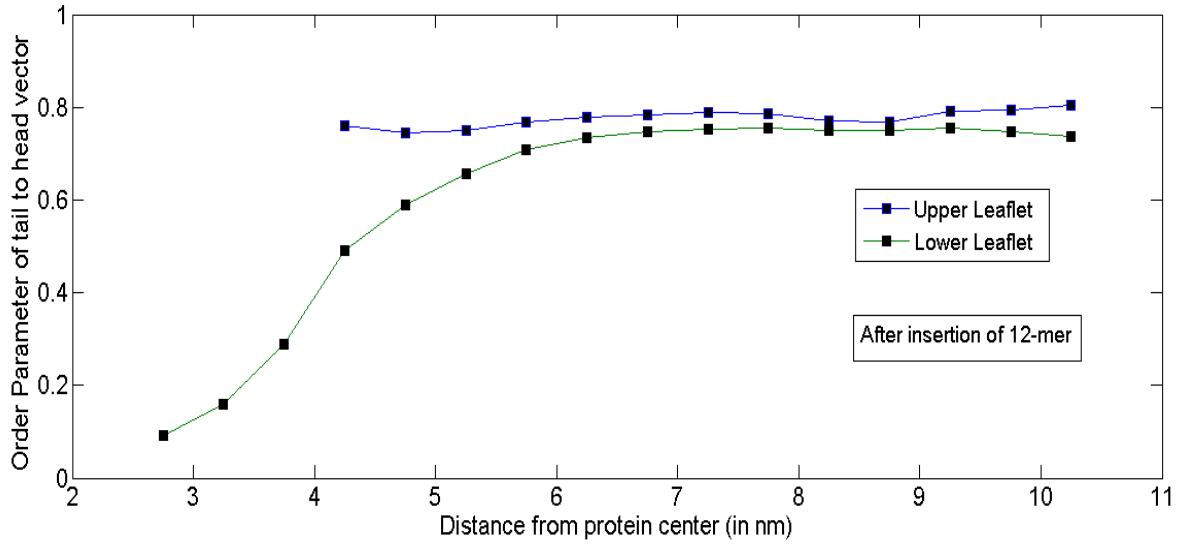
Figure 5. Variation of slow time constant as a function of distance from the centre of the Protein

4.3. Change in Orientation of the Lipid:

The orientation of the lipid with respect to membrane normal (i.e. Z axis for our case) was studied with help of order parameter as discussed in section 3.4.1.



(a) 9-mer



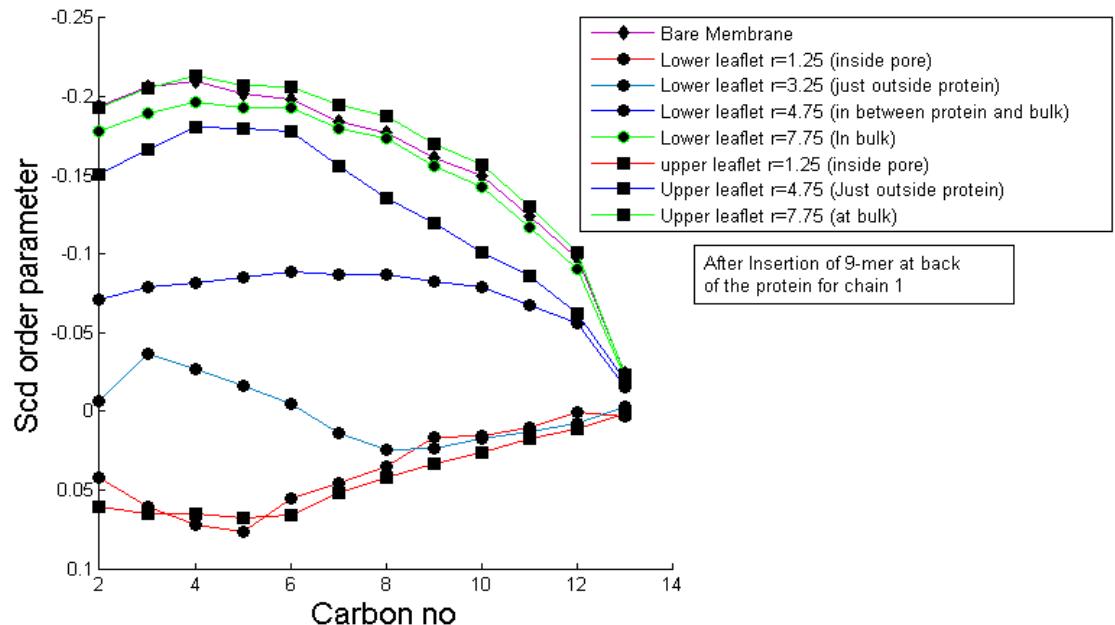
(b) 12-mer

Figure 6. Variation of Z-order parameter as a function of distance from protein centre

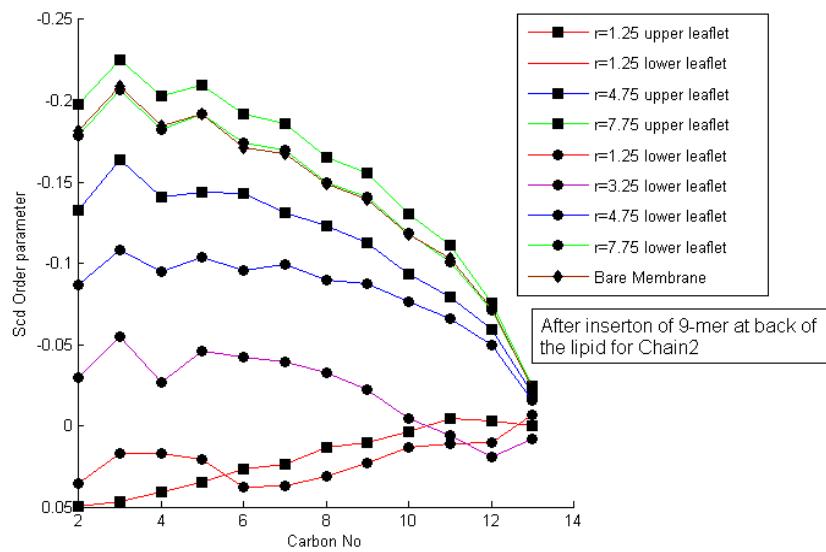
Figure 6 shows that upper leaflets at back of the 9-mer protein and near the surface of 12-mer protein are oriented straight whereas lower leaflet has become disordered (i.e. they are not parallel to membrane normal). The lipids inside the 9-mer are also not straight and in this case also lipids in the lower leaflet are more disoriented with respect to upper leaflet. The lipids in front of the pore have regained their orientation in less distance in comparison to lipids at the back.

4.4. Change in Order of the lipid

As discussed in section 3.4.2, the negative value of S_{CD} order parameter also suggests some order in system.

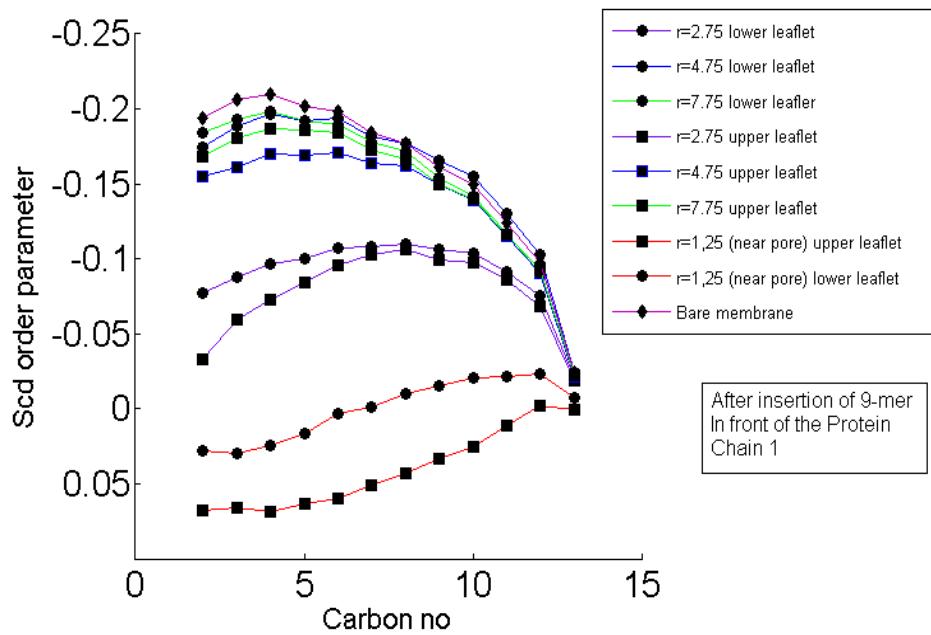


(a) Chain 1

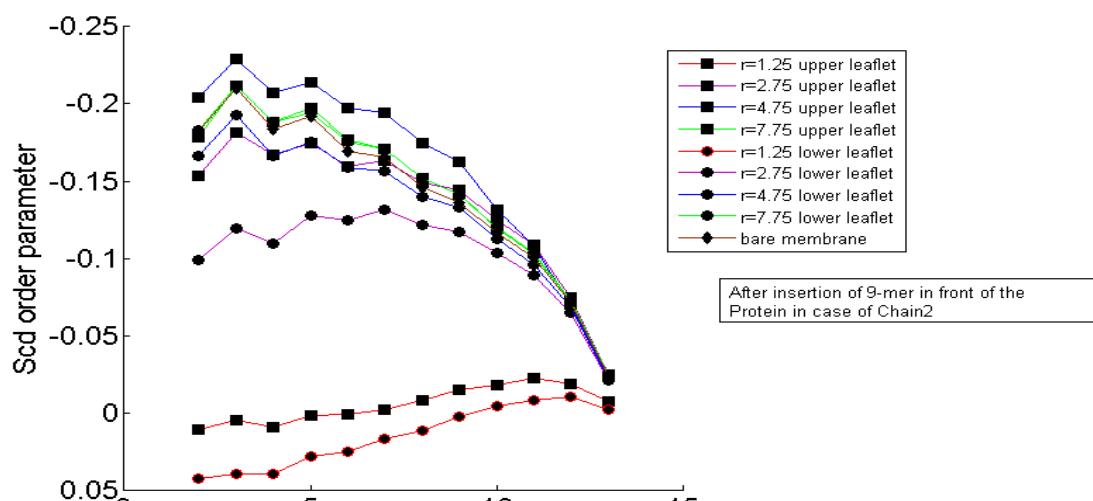


(b) Chain 2

Figure 7: S_{CD} order Parameter for different carbon atoms (for lipids at the back of the 9-mer)

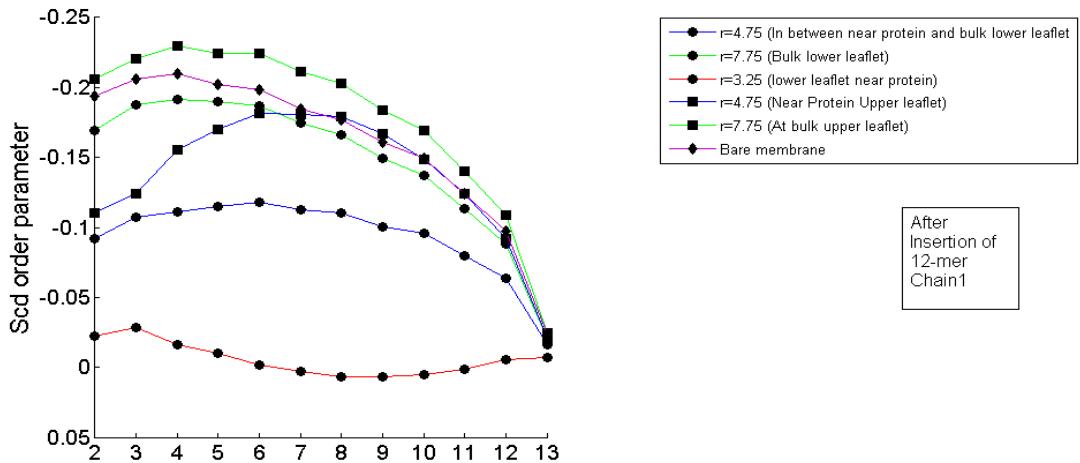


(a) Chain 1

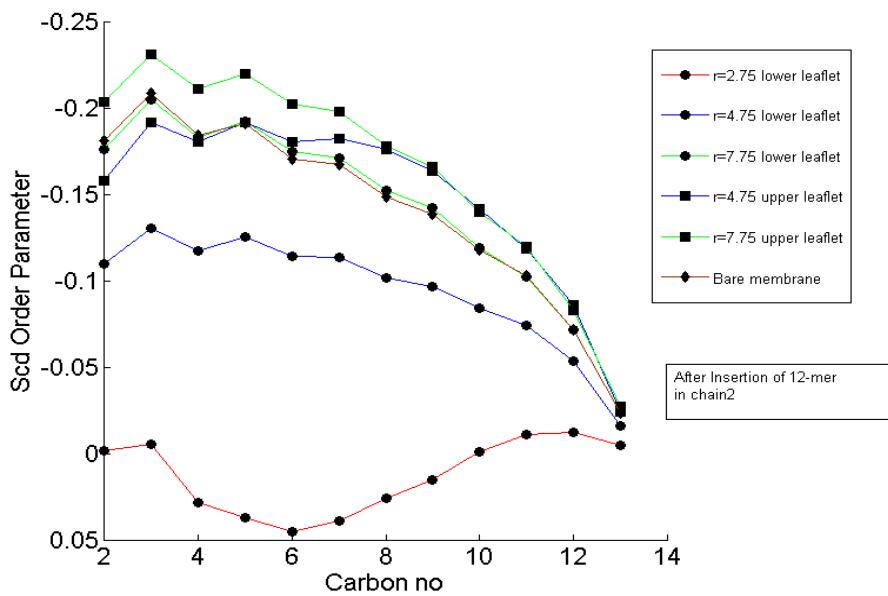


(b) Chain 2

Figure 8: S_{CD} order parameter for lipids in front of the 9-mer



(a) Chain 1



(b) Chain 2

Figure 9: S_{CD} order parameter for lipids after introduction of 12-mer

Figure 7, 8, and 9, illustrate the phase transition between liquid ordered (L_O) and liquid disordered (L_D) phase of lipid with respect to position of the inserted protein oligomers. They clearly show that lipids are completely disordered when they are inside the pore. Both in case of lipids at the back of 9-mer and around 12-mer, lower leaflet lipids are highly disordered in case of the lower leaflet but ordered in case of upper leaflet. Even in the bulk membrane, lower leaflet is slightly more disordered than the upper leaflet (could be due to long range effects). The gradual transition of disordered to ordered phase is also seen in case of lower

leaflets. Though the lipids in the front of 9-mer show similar trends the difference of disorder between upper and lower leaflets is less and the transition from disorder to order is more rapid with respect to the lipids at the back. The degree of order at the bulk is close to that of the bare membrane and the order parameter value of the bare membrane matches well with the value obtained from experiments [9] and other simulation [10] for pure DMPC.

4.5. Diffusivity of the Sample

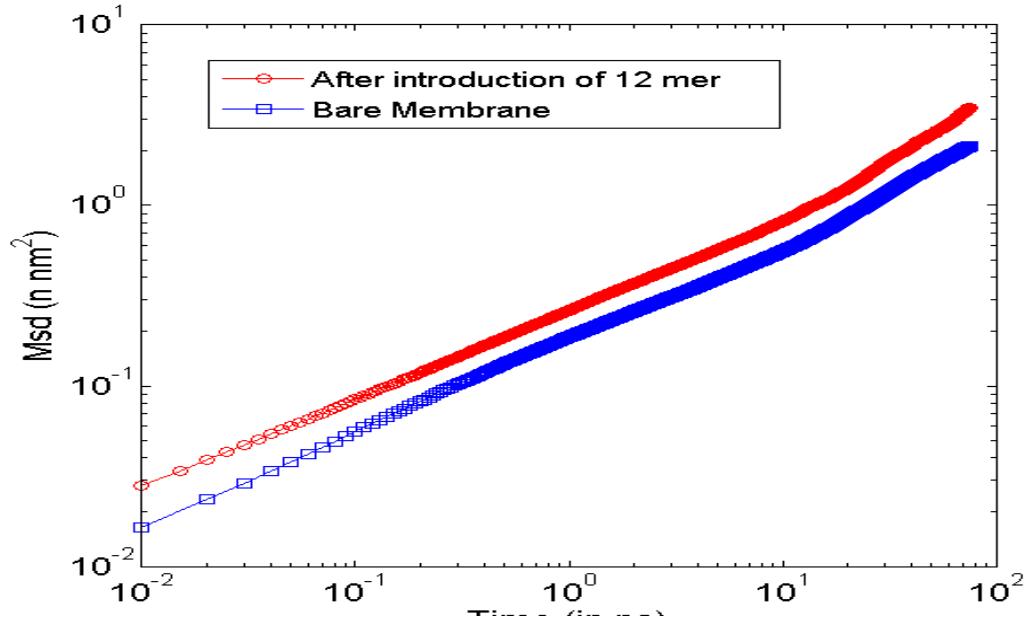


Figure 10: Comparison of average MSD of a bare membrane and the 12-mer

Figure 10 shows that average M.S.D. of the lipid has increased due to the 12-mer, whereas peak in the distribution (Figure 11) has shifted to sub diffusive region when we fit that data in equation 8 as discussed in section 3.5. This is quite anomalous.

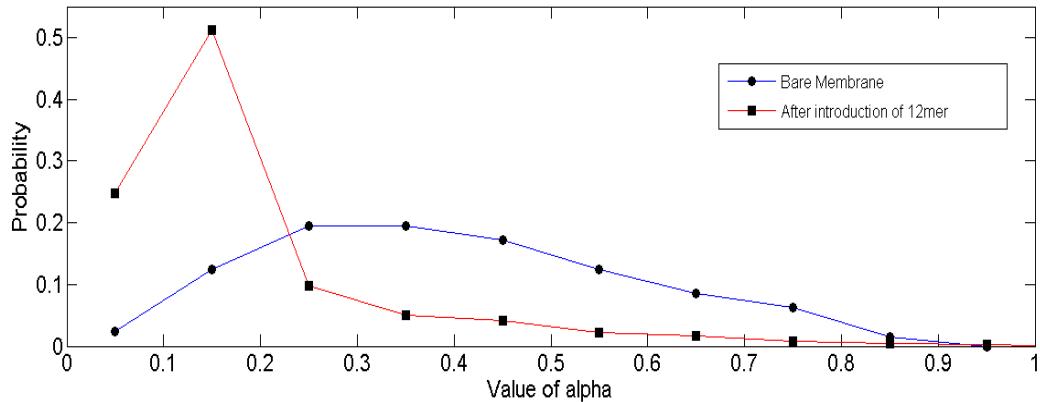


Figure 11: Comparison of value of alpha got from fitting MSD curve of a bare membrane and after introduction of proteins

But figure 12 may answer the question. We can see that in case of bare membrane the M.S.D. curve of the lower and higher alpha are not far apart. But in case of introduction of protein the M.S.D. curve of higher and lower alpha are comparatively more apart from each other. This may be because increased fluidity makes some lower leaflet lipids so diffusive (the lipid shown in figure 12 has almost 4 times more M.S.D than the average) that they suppress the slower lipids in the upper leaflet and make the average higher than bare membrane also.

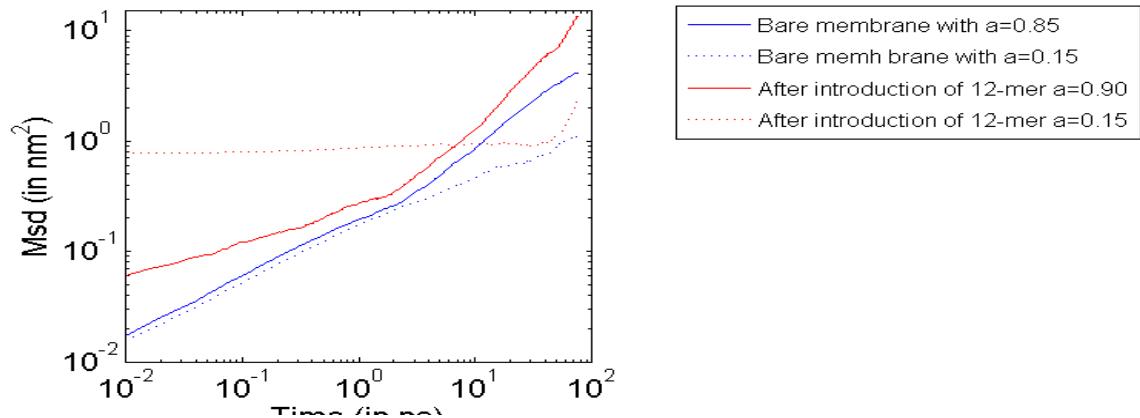


Figure 12: Comparison of M.S.D of some individual lipids both from bare membrane and membrane with 12-mer.

5. CONCLUSION

The above observations show that ClyA affects the lipid dynamics after insertion in either the 9-mer or the 12-mer form. This slowing down is more in case of upper leaflet than lower leaflet. On the other hand Lower leaflet becomes disordered and disoriented whereas order and orientation of the upper leaflet is not affected much. The insertion of the protein with truncated

cone geometry in lower side in an otherwise regular and symmetrical phospholipid bilayer is the possible reason for this. Increase of disorder in the system should increase fluidity of the sample, but we see the leaflet to be slower. We can explain this in following way: the hydrophilic phosphate group is oriented towards the water molecules outside. The protein also likes to anchor itself in the sea of lipid in such a way that the hydrophilic groups are in the lipid-water interface. The presence of basic amino acids such as Arginine and Lysine in the lipid-water interface supports this hypothesis. The interaction of these groups with phosphate group possibly makes the lipid motion slower. In case of lower leaflet the increase of fluidity counteracts this effect and makes the lipid comparatively fast.

6. APPENDICES

6.1. Appendix A: MATLAB Code for Calculation of protein centre radius of Curvature

```

(a) fid = fopen ('Protein_Chains_COM.txt' , 'r');
(b) fid2=fopen ('proteincent.txt' , 'w');
(c) timestep=15000;
(d) for (i=0:timestep)
(e)   a=fscanf (fid, '%f',28);
(f)
(g)   for (j=1:9)
(h)     x(j)=a(3*j-1);
(i)     y(j)=a(3*j);
(j)   end
(k)   count=0;
(l)   for (k=1:9)
(m)     for (l=k+1:9)
(n)       for (m=l+1:9)
(o)         A=[2*x(k),2*y(k),1 ;2*x(l),2*y(l),1;2*x(m),2*y(m),1];
(p)         B=[x(k)*x(k)+y(k)*y(k);x(l)*x(l)+y(l)*y(l);x(m)*x(m)+y(m)*y(m)];
(q)
(r)         z=linsolve(A,B);
(s)         count=count+1;
(t)         r=sqrt (z(1)*z(1)+z(2)*z(2)+z(3));
(u)         X(count)=z(1);
(v)         Y(count)=z(2);
(w)         R(count)=r;
(x)
(y)       end
(z)     end
(aa)   end
(bb)   x0=mean(X);
(cc)   y0=mean (Y);
(dd)   R0=mean (R);
(ee)   Rdev=std (R);
(ff)
(gg)   fprintf (fid2, '%f %f %f %f %f\n', a(1), x0, y0, R0, Rdev);
(hh)   fprintf (fid3, '%f %f %f\n', a(1)*0.001 , R0, Rdev);
(ii) end
(jj)

```

6. 2. Appendix B: MATLAB Code for Calculation of Survival probability

```

fin1=fopen ('proteincent.txt' , 'r');% opening the file containing time, protein center of curvature
and radius and standard deviation respectively

```

```

fin2=fopen ('lipidcoord.txt','r');% opening the file containing P coordinates
fin3=fopen ('chord.txt','r');% opening the file containing coordinates of two protein Ca atom
with which chords to be drawn
fout2=fopen ('survprobforthup.txt', 'w');% opening the output files
fout1=fopen ('survprobbackup.txt', 'w');% opening the output files
fout4=fopen ('survprobforthlow.txt', 'w');% opening the output files
fout3=fopen ('survprobbacklow.txt', 'w');% opening the output files
timestep=15000;% number of timesteps
l=7500;% number of shifted time averaging to be done
natoms=758;% number of lipids
rmax= 15;% Maximum Radius
dr=0.5;% Shell size
n=floor((rmax/dr));% Number of radial shells
domain=zeros(timestep+1,natoms);% Creating the array to store the shell no
% loop starts for allocating the coordinate
for (i=0:timestep)
protein= fscanf (fin1,'%f',5);% Reading the time,PRotein COC, Radius etc
time=fscanf (fin2,'%f',1);% Reading the time data from lipid coordinates file
chord=fscanf (fin3,'%f',6);% Reading the X,Y,Z coordinates for two points through which chord
to be drawn
s=(chord(5)-chord(2))/(chord(4)-chord(1));% Slope of the line of the chord
c=chord(2)-s*chord(1);% Intercept of the chord
if ((protein(3)-protein(2)*s-c)<0) flag=-1; %Seeing in Which side of the chord center is existing
that will be treated as back
else flag=1;
end
c2=protein(3)-protein(2)*s;
for (j=1:natoms)
coord=fscanf (fin2,'%f',3);% Reading the coordinates of P atoms
rx=coord(1)- protein(2);
ry=coord(2)- protein(3);
rl=sqrt (rx*rx+ry*ry);% Finding the distance from Protein center of curvature
if ((coord(2)-coord(1)*s-c2)<0) flag2=-1;
else flag2=1;
end
m=(rl/dr);% Allocating the radial shell
if (flag==flag2)m=-m;% Shifting the shell if it is in backside of the protein
end
if (coord(3)>12.5) m=m+2*n;% Separating the coordinates of lower leaflet Point to be
remembered highZ =lower leaflet
end
domain(i+1,j)=n+1+floor(m);% Allocating the domain by taking
end
end
avgprob=zeros (timestep,4*n);%creating the array for storing the survival probability of each
timestep in each shell
% loop starts for shifted timeaveraging
for (i=0:l)

```

```

count=zeros (4*n,1);%Creating the array to store initial count of lipids in each shell
prob=ones (natoms,1);%making the probability=1 for each atom
for (j=1:natoms)
    count (domain(i+1,j),1)= count (domain(i+1,j),1)+1 ;% Counting the initial no of atom for
each shell for that t0
end
%loop starts for timestep onwards
for (p=i+1:timestep)
    totalprob=zeros (4*n,1); % Creating the array for counting the atoms whose probability=1 in
each shell in a timestep
    %loop starts for all lipids
    for (j=1:natoms)
        if (prob(j,1)~=0) %Doing the calculation only if probability is nonzero

            if (domain(p+1,j)~=domain(i+1,j)) % Making the probability=0 if the atom is out of its
mothershell
                prob(j,1)=0;
            end
            totalprob (domain(i+1,j),1) = totalprob (domain(i+1,j),1)+ prob(j,1) ;% counting the no
of atom still in that shell
        end
    end
    for (k=1:4*n)
        if(count(k,1)~=0)
            avgprob(p-i,k)= avgprob(p-i,k)+(totalprob(k,1)/count(k,1)) ;% Adding the survival
probability of all shells at that timestep
        end
    end
end

disp (i) % Keeping track of how much it is done
end

for (j=1:n)

fprintf (fout1, '%f %f %f \n',(j-0.5)*dr,0 ,1);%outputing the S.P.=1 at t=0
fprintf (fout2, '%f %f %f \n',(j-0.5)*dr,0 ,1);%outputing the S.P.=1 at t=0
fprintf (fout3, '%f %f %f \n',(j-0.5)*dr,0 ,1);%outputing the S.P.=1 at t=0
fprintf (fout4, '%f %f %f \n',(j-0.5)*dr,0 ,1);%outputing the S.P.=1 at t=0
for (i=1:timestep)
    for (k=0:3)
        m=k*n+j;
        if (i<=(timestep-1)) avgprob (i,m)= avgprob (i,m)/(l+1);% Shifted time averaging
        else avgprob(i,m)=avgprob(i,m)/(timestep-i+m);% Shifted time averaging
    end
end
% outputing for all four sections
fprintf (fout1, '%f %f %f \n',(j-0.5)*dr,i*0.01, avgprob (i,j));
fprintf (fout2, '%f %f %f \n',(j-0.5)*dr,i*0.01, avgprob (i,j+n));

```

```

    fprintf (fout3, '%f %f %f \n',(j-0.5)*dr,i*0.01, avgprob (i,j+2*n));
    fprintf (fout4, '%f %f %f \n',(j-0.5)*dr,i*0.01, avgprob (i,j+3*n));
end

```

```

disp (j);% Keeping track of how much it is done :P
end

```

6.3. Appendix C : C Code for tail to head order parameter calculation

```

#include<stdio.h>
#include<math.h>
#include<stdlib.h>
/*Defining a function for calculating order parameter w.r.t to a bond (arguement1) and an axis
(arguement2)*/
float bondorder (float veca[3],float vecb[3]) {
    int k; float dot=0;float modA=0;float modB=0;float cos,bo;
    for (k=0;k<3;k++) {
        dot+=veca[k]*vecb[k];/*calculating the dot product*/
        modA+=veca[k]*veca[k];/*Calculating modulus of 1st vector (arguement1)*/
        modB+=vecb[k]*vecb[k];
    }
    cos=(float)(dot/sqrt (modA*modB));/*Calculating the cos theta between the vectors*/
    bo=0.5*(3*cos*cos-1);/*Calculating the order parameter*/
    return bo;
}

int main () {
    FILE *fin1, *fin2 ,*fin3 , *fin4,*fin5 ,*fout, *fout2, *fout3;
    fin1=fopen ("proteincenter.txt", "r");/* opening the file containing time, protein center of
curvature and radius and standard deviation respectively*/
    fin2=fopen ("lipidcoord", "r"); /* opening the file containing P coordinates*/
    fin3=fopen ("C2coordin", "r");/*Containing all the carbon coordinates of 1st chain (i.e.
C23,c24,C25,.....C214)*/
    fin4=fopen ("C3coordin", "r");/*Containing all the carbon coordinates of 2st chain (i.e.
C33,c34,C35,.....C314)*/
    fin5=fopen ("chord.txt", "r");/* opening the file containing coordinates of two protein Ca
atom with which chords to be drawn*/
    /*output files*/
    fout=fopen ("zorderleaflet2", "w");
    int nlipids=758;
    int timestep=15000;
    float rmax=15;
    float dr=0.5;
    int n= (rmax/dr);
    int natoms=12;
    int i,j,k,l,m,t,p,count[4*n];
    float protein[5], c2pos[natoms][3],
c3pos[natoms][3],hpos[3],axisvec[3],axisvec1[3],axisvec2[3],bondvec[3],time,z[3],zorder[4*n],b
o[natoms-1][2],rx,ry,rl, border[4*n][natoms-1][2],chord[6],c,flag,c2,flag2,s;

```

```

z[0]=0;z[1]=0;z[2]=1;

/*initialising the value of count and orderparameters*/
for (i=0;i<4*n;i++){
    count[i]=0;zorder[i]=0;
    for (j=0;j<natoms;j++){
        border[i][j][0]=0;
        border[i][j][1]=0;
    }
}
/*Starting the loop for timesteps */
for (i=0;i<=timestep;i++) {
    /*Reading the time value from all files */
    fscanf (fin2, "%f",&time);
    fscanf (fin3, "%f",&time);
    fscanf (fin4, "%f",&time);

    /*Drawing the chord well explained in Survival probability code */
    for (m=0;m<5;m++) fscanf (fin1, "%f",&protein[m]);
    for (m=0;m<6;m++) fscanf (fin5, "%f",&chord[m]);
    s=(chord[4]-chord[1])/(chord[3]-chord[0]);
    c=chord[1]-s*chord[0];
    if ((protein[2]-protein[1]*s-c)<0) flag=-1;
    else flag=1;
    c2=protein[2]-protein[1]*s;

    for (j=0;j<nlipids;j++) {
        for (m=0;m<3;m++) fscanf (fin2, "%f",&hpos[m]);/* Reading P coordinates*/
        for (k=0;k<natoms;k++) {
            for (m=0;m<3;m++) fscanf (fin3, "%f",&c2pos[k][m]); /*Reading 1st chain
coordinates*/
            for (m=0;m<3;m++) fscanf (fin4, "%f",&c3pos[k][m]);/*Reading 2nd chain
coordinates*/
        }
        for (m=0;m<3;m++) {
            axisvec[m]=c2pos[natoms-1][m]+c3pos[natoms-1][m]-2*hpos[m];/*Finding
Average head to tail vector*/
        }
    }
    /*Allocating shell and dividing into upper and lower leaflet discussed in survival
probability code*/
    rx=hpos[0]- protein[1];
    ry=hpos[1]- protein[2];
    rl=sqrt (rx*rx+ry*ry);
    t=(int)(rl/dr);
    if ((hpos[1]-hpos[0]*s-c2)<0) flag2=-1;
    else flag2=1;
    if (flag==flag2 && hpos[2] < 12.5) p=t;
}

```

```

if (flag==flag2 && hpos[2] > 12.5) p=n+t;
if (flag!=flag2 && hpos[2] < 12.5) p=2*n+t;
if (flag!=flag2 && hpos[2] > 12.5) p=3*n+t;
    count[p]++;
    zorder[p]+=bondorder (axisvec,z);/*Calling the function to
calculate Z order parameter and add it*/
    for (k=0;k<natoms-1;k++) {
        for (l=0;l<2;l++) border[p][k][l]+=bo[k][l];
    }
}
printf ("%d ",i);/*Seeing how much it's done :P*/
}

/*Averanging for the bond Order*/
for(i=0;i<4*n;i++) {
    zorder[i]/=count[i];
}

/*printing the outputs*/
for (i=0;i<n;i++) {

    for (j=0;j<4;j++) {

        fprintf (fout, "%f", zorder[n*j+i] );
    }
    fprintf (fout, "\n" );
}
return 0;
}

```

6.4. Appendix D:C Code for S_{CD} order parameter calculation

```

#include<stdio.h>
#include<math.h>
#include<stdlib.h>

float bondorder (float veca[3],float vecb[3]) {
    int k; float dot=0;float modA=0;float modB=0;float cos,bo;
    for (k=0;k<3;k++) {

```

```

dot+=veca[k]*vecb[k];
modA+=veca[k]*veca[k];
modB+=vecb[k]*vecb[k];
}
cos=(float)(dot/sqrt (modA*modB));
bo=0.5*(3*cos*cos-1);
return bo;
}

int main () {
FILE *fin1, *fin2 ,*fin3 , *fin4,*fin5 ,*fin6,*fout, *fout2, *fout3;
fin1=fopen ("proteinent.txt", "r");
fin2= fopen ("lipidcoord", "r");
fin3=fopen ("C2coordn", "r");
fin4=fopen ("C3coordn", "r");
fin5=fopen ("chord.txt", "r");
fin6=fopen ("Hcoordx ","r");
fout=fopen ("scdorderpara1.txt","w");
fout2=fopen ("scdorderpara2.txt","w");
int nlipids=758;
int timestep=15000;
float rmax=15;
float dr=0.5;
int n= (rmax/dr);
int natoms=12;
int i,j,k,l,m,t,p,count[4*n];
float protein[5], c2pos[3], hpos[3], axisvec[3], axisvec1[3], axisvec2[3], bondvec[3], time,
z[3],zorder[4*n], bo1[natoms][2],bo2[natoms][2],rx,ry,rl, border[4*n][natoms][2],chord[6], c,
flag, c2, flag2, s, hydro[3];
z[0]=0;z[1]=0;z[2]=1;

for (i=0;i<4*n;i++){
    count[i]=0;
    for (j=0;j<natoms;j++){
        border[i][j][0]=0;
        border[i][j][1]=0;
    }
}

for (i=0;i<=timestep; i++) {
    fscanf (fin2, "%f",&time);
    fscanf (fin3, "%f",&time);
    fscanf (fin4, "%f",&time);

    for (m=0;m<5;m++) fscanf (fin1, "%f",&protein[m]);
    for (m=0;m<6;m++) fscanf (fin5, "%f",&chord[m]);
    s=(chord[4]-chord[1])/(chord[3]-chord[0]);
}

```

```

c=chord[1]-s*chord[0];
if ((protein[2]-protein[1]*s-c)<0) flag=-1;
else flag=1;
c2=protein[2]-protein[1]*s;

for (j=0;j<nlipids;j++) {
    for (m=0;m<3;m++) fscanf (fin2, "%f",&hpos[m]);
    rx=hpos[0]- protein[1];
    ry=hpos[1]- protein[2];
    rl=sqrt (rx*rx+ry*ry);
    t=(int)(rl/dr);
    if ((hpos[1]-hpos[0]*s-c2)<0) flag2=-1;
    else flag2=1;
    if (flag==flag2 && hpos[2] < 12.5) p=t;
    if (flag==flag2 && hpos[2] > 12.5) p=n+t;
    if (flag!=flag2 && hpos[2] < 12.5) p=2*n+t;
    if (flag!=flag2 && hpos[2] > 12.5) p=3*n+t;
    count[p]++;
    for (k=0;k<natoms;k++) {
        for (m=0;m<3;m++) fscanf (fin3, "%f",&c2pos[m]);
        for (l=0;l<2;l++) {
            for (m=0;m<3;m++) fscanf (fin6, "%f",&hydro[m]);
            for (m=0;m<3;m++) bondvec[m]=c2pos[m]-hydro[m];
            if (l==0) bo1[k][0]=bondorder (bondvec,z);
            else bo2[k][0]=bondorder (bondvec,z);
        }
    }
}

for (k=0;k<natoms;k++) {
    for (m=0;m<3;m++) fscanf (fin4, "%f",&c2pos[m]);
    for (l=0;l<2;l++) {
        for (m=0;m<3;m++) fscanf (fin6, "%f",&hydro[m]);
        for (m=0;m<3;m++) bondvec[m]=c2pos[m]-hydro[m];
        if (l==0) bo1[k][1]=bondorder (bondvec,z);
        else bo2[k][1]=bondorder (bondvec,z);
    }
}

for (k=0;k<natoms;k++) {
    for (l=0;l<2;l++) border[p][k][l]+=(0.5*(bo1[k][l]+bo2[k][l]));
}
printf ("%d ",i);
}

for(i=0;i<4*n;i++) {
    for(k=0;k<natoms;k++) {

```

```

        border[i][k][0]=(float) (border[i][k][0]/count[i]);
        border[i][k][1]=(float) (border[i][k][1]/count[i]);
    }

}

for (i=0;i<n;i++) {
    printf ("%d ",count[i] );
    for(k=0;k<natoms;k++ ){
        fprintf (fout, "%f %d %f %f %f %f\n", (i+0.5)*dr, k+2,
border[i][k][0],border[i+n][k][0],border[i+2*n][k][0],border[i+3*n][k][0]);
        fprintf (fout2, "%f %d %f %f %f %f\n", (i+0.5)*dr, k+2,
border[i][k][1],border[i+n][k][1],border[i+2*n][k][1],border[i+3*n][k][1]);
    }
}
return 0;
}

```

6.5. Appendix E: MATLAB code for MSD calculation

```

fin=fopen ('DMPC_APL_NORMAL_200nsMD_P_pbcnojump.gro','r');%opening input
file
be shifted tmme averaged
nlipids=128;%no of lipids
coord=zeros (nlipids,timestep+1,3);% Allocating the array for coordinates
%loop starts for reading coordinates
for (i=0:timestep)
    time=fgets (fin);%reading timestep
    a=fgets (fin);%reading unnecessary line
    for (j=1:nlipids)
        b=fscanf (fin,'%s',3);% reading unnecessary words
        for (k=1:3)
            coord(j,i+1,k)=fscanf (fin,'%f',1);%reading X,y,z coordinate for each phosphorus atom
at each timestep
        end
    end
    a=fgets (fin);
    a=fgets (fin);%reading unnecessary line
    disp (i)%observing how much it's done

end
disp (coord);
for (i=1:nlipids)
    msdtot=zeros(timestep,1);%creating the array for MSD
    %loop starts for shifted time averaging
    for (j=0:1)
        for (k=j+1:timestep)

```

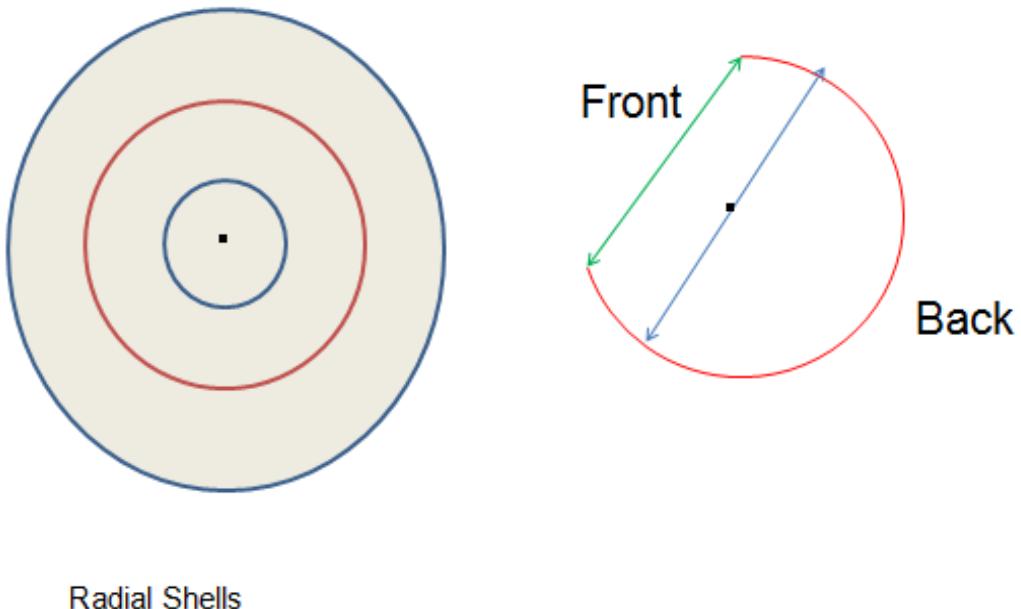
```

for (m=1:3)
    dev(m)=coord(i,k+1,m)-coord(i,j+1,m);%finding the deltarx,deltary,deltarz
end
msdx=(dev(1)*dev(1)+dev(2)*dev(2)+dev(3)*dev(3));%finding the msd
msdtot(k-j,1)= msdtot(k-j,1)+msdx;%msd is added for shifted time averaging
end
end
for (j=1:timestep)
    if (j<=(timestep-l)) msdtot (j,1)= msdtot (j,1)/(l+1);%Shifted time averaging
    else msdtot (j,1)=msdtot (j,1)/(timestep-j+1);%Shifted time averaging
end
end
fprintf (fout,'%f %f\n', i, msdtot);%printing the output

```

end

Appendix F: Schematic of Shell Allocation.



Radial Shells

ACKNOWLEDGEMENT

I would like to thank **Prof K.G. Ayappa** and **Prof. J.K.Basu** for giving this opportunity to work in this topic. The extensive discussion with them helped me to figure out the problem and its solution. I must thank **Mr. Rajat Desikan** for his invaluable support and guidance throughout the project in conceptualizing the problem. All the analysis were done on trajectory from the simulations run by him. I also want to thank BEST programme for giving us administrative supports and funds during the programme. I am also thankful to faculty members of IISc who gave lectures and research seminars for enlightenment

in the field of Bioengineering. All the members in the lab of Prof. Ayappa and Prof. Basu enriched me during my stay in IISc.

REFERENCES

1. Parker, M. W. & Feil, S. C. Pore-forming protein toxins: from structure to function. *Prog. Biophys. Mol. Biol.* 88, 91–142 (2005)
2. Mueller M, Grauschof U, Maier T, Glockshuber R, Ban N (2009) The structure of a cytolytic α -helical toxin pore reveals its assembly mechanism. *Nature* 459:726-730.
3. Aksimentiev A, Schulten K. Imaging α -Hemolysin with Molecular Dynamics: Ionic Conductance, Osmotic Permeability, and the Electrostatic Potential Map. *Biophysical Journal*. 2005;88(6):3745-3761. doi:10.1529/biophysj.104.058727.
4. Niemela Perttu S, Miettinen Markus S, Monticelli Luca, Hammaren Henri, Bjelkmar k Pa Murtola Teemu, Lindahl Erik, Vattulainen Ilpo Membrane Proteins Diffuse as Dynam Complexes with Lipid, *Journal of American Chemical Society*, 2010, 132, 7574–7575
5. Desikan R, Maity P, Ayappa K. G. ,Transmembrane intermediates reveal the pore forming mechanism of an α -helical cytolysin (sent for publication)
6. Bhakdi S, Tranum-Jensen J, Sziegoleit A (1985) Mechanism of membrane damage by streptolysin-o. *Infection and Immunity* 47: 52-60.
7. Malani, Ateeque and Ayappa K. G., Relaxation and jump dynamics of water at the mica interface, *The Journal of Chemical Physics*, 136, 194701 (2012), DOI:<http://dx.doi.org/10.1063/1.4717710>
8. Vermeer, LouicS. , de Groot, BertL., Réat, Valérie ,Milon Alain, Czaplicki Jerzy, *Acyl chain order parameter profiles in phospholipid bilayers: computation from molecular dynamics simulations and comparison with 2 H NMR experiments*, *Eur Biophys J* (2007) 36:919–931 DOI 10.1007/s00249-007-0192-9
9. Hofsäß C, Lindahl E, Edholm O. Molecular Dynamics Simulations of Phospholipid Bilayers with Cholesterol. *Biophysical Journal*. 2003;84(4):2192-2206.
10. Lu D, Vavasour I, Morrow MR. Smoothed acyl chain orientational order parameter profiles in dimyristoylphosphatidylcholine-distearoylphosphatidylcholine mixtures: a 2 H-NMR study. *Biophysical Journal*. 1995;68(2):574-583.

11. Davis JE, Rahaman O, Patel S. Molecular Dynamics Simulations of a DMPC Bilayer Using Nonadditive Interaction Models. *Biophysical Journal*. 2009;96(2):385-402. doi:10.1016/j.bpj.2008.09.048.