1 Principle of permeation counting

A permeation is the event that starts with a water membrane entering the membrane and ends with the membrane exiting into the opposite water compartment where it was previously located, see Fig.1.

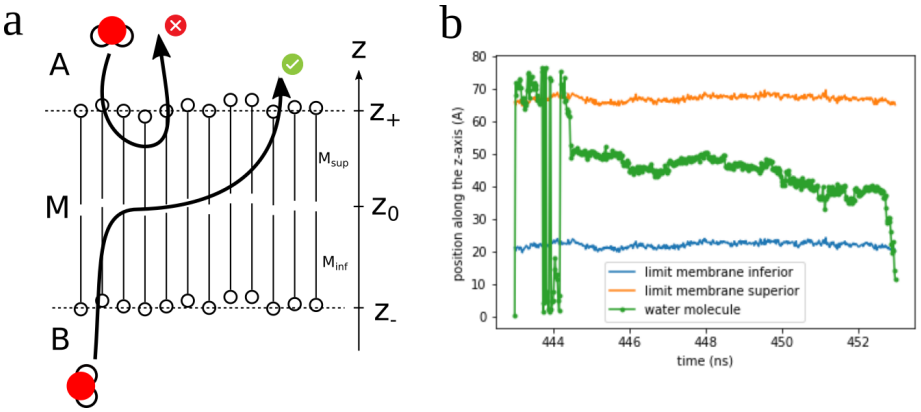

Figure 1: a) Diagram of a permeation event and definition of notations. b) Trajectory of a water molecule along the z direction normal to the membrane as a function of time covering the duration of a water permeation.

2 Algorithm

The algorithm takes place in two stages: 1) The calculation of a simplified 3-states trajectory 2) The detection of the permeation event from the simplified trajectory.

These stages are repeated for all water molecule chunks, preventing the creation of large NumPy integer arrays. If an array becomes too large due to an excessive number of timeframes, the algorithm will prompt the user to increase the number of chunks.

In the first stage, we define the state of a molecule depending on its positon along z axis. For this purpose we compute the trajectory of the lower and upper leaflets defined by the average z-coordinates over all phosphorus atoms belonging to the same leaflet ($z_+(t)$ and $z_-(t)$, see Fig.1). The state of the ith water molecule is defined as follows :

$$S^i(t) = \begin{cases} +1 & \text{if } z^i(t) > z_+(t) \\ -1 & \text{if } z^i(t) < z_-(t) \\ 0 & \text{if } z_-(t) < z^i(t) < z_+(t) \end{cases} \quad (1)$$

Another correction is applied on the water molecules in the membrane which depends on the pre-computed displacement $\Delta z^i(t) = z^i(t) - z^i(t - \Delta t)$ of

each particule. If a water molecule undergoes a large jump from or to the central membrane region ($\Delta z^i(t) > MIN\_DISP\_MEMB$ or $\Delta z^i(t+\Delta t) > MIN\_DISP\_MEMB$), then we assume this large displacement to be due to the diffusion of the water through the closest water bulk following by the periodic boundary jump. We therefore change the state $S(t)$ of the molecule according to the following criteria :

$$S^i(t) \leftarrow \begin{cases} +1 & \text{if } z_0(t) < z^i(t) < z_+(t) \\ -1 & \text{if } z^i(t) < z_-(t) < z_0(t) \end{cases} \tag{2}$$

In the original implementation of the code, this criterion was introduced to prevent counting events where a water molecule crosses the periodic boundary immediately after leaving or entering the membrane. These particular cases were observed when the saving time step in the algorithm was taken large, i.e. on the order of magnitude of the average displacement of a water molecule in the bulk. However, in order to to avoid any artifacts due to the periodic boundary crossings, it is highly recommended to work with small saving time step (e.g. a good tradeoff is $\Delta t = 50ps$). Then this correction should not be necessary. Moreover, it has been shown that for $\Delta t > 100ps$, this correction procedure introduces another bias that overestimates the real number of permeations. An example is a molecule being just above $z_0(t)$ and jumping to the lower water compartment through the membrane/channel; in this case the event after analysis of the simplified trajectory would be mistaken for permeation.

The operations performed in the first step involves NumPy arrays of size $N \times T$ where N is the number of water molecules in a chunk and T the number of frames.

In the second stage of the algorithm, we performed on each single water simplified trajectory $S^i(t)$ to identify the possible permeation events. First we compute an T-NumPy array $history^i(t)$ that stores at time t the time duration that a water molecule stay in the membrane. The sign of $history^i(t)$ provides information on the last bulk region occupied by the water molecule before entering the membrane.

In order to identify a permeation event, we look for the frame t for which these two conditions are met :
a) the water molecule has just moved from the membrane to the bulk
b) the sign of the jump $\Delta S^i(t) = S^i(t) - S(t - \Delta t)$ is opposite to the last bulk state visited

In this part of the algorithm, the operations are performed on T-NumPy arrays carrying unsigned integers.

3 Best practices

Although it is tempting to use a time step that is not too small to save disk memory, it is important to remember that biases can appear in the algorithm's results if the time step is too large. The user is therefore advised to always test

at least once the influence of the time step on the number of events detected (e.g. 100 ps, 50 ps, 20 ps). The smaller the height of the water layer above the membrane, the greater the error, so 50 ps is not always a reliable value, depending on the geometry of your system.

Algorithm 1 Compute permeations events through a lipid membrane in MD simulations

Set MIN_DISP_MEMB ▷ the minimum displacement to account for a PBC jump
From the phosphorus atom coordinates, compute T-arrays with:
$z^+[T]$: position of the upper membrane boundary
$z^-[T]$: position of the lower membrane boundary
$z_0[T] = \frac{z^+[T]+z^-[T]}{2}$: position of the center of the membrane

for all groups of water molecules **do**
▷ option to avoid large memory in RAM
  Compute a simplified water trajectory
  $S_b^+[T] \leftarrow (z[T] > z^+[T])$ ▷ 1 if in upper bulk region, else 0
  $S_b^-[T] \leftarrow -(z[T] < z^-[T])$ ▷ -1 if in lower bulk region,else 0
  $S_m^+[T] \leftarrow 1$ if in upper membrane region, else 0
  $S_m^-[T] \leftarrow$ -1 if in lower membrane region, else 0
  $S_{\text{jump}}[T] \leftarrow$ array with 1 if jump with $|dz| >$ MIN_DISP_MEMB
  $S[T] \leftarrow S_b^-[T] + S_b^+[T] + (S_m^-[T] + S_m^+[T]) \cdot S_{\text{jump}}[0:T-1] + (S_m^-[T] + S_m^+[T]) \cdot S_{\text{jump}}[1:T]$
  ▷ simplified 3-state trajectory with corrections from close bulk state if involved in a PBC jump

  for all N water molecules **do**
    Compute the history of a single molecule
    history $\leftarrow 0$ ▷ to store history state and time
    memory $\leftarrow 0$ ▷ intermediary variable
    for t over T frames **do**
      history$[t] \leftarrow S[t]$
      if $S[t] = 0$ **then** ▷ the water molecule is in the membrane
        history$[t] \leftarrow$ memory + sign(memory) ▷ cumulate history
      end if
      memory $\leftarrow$ history$[t]$
    end for
    Compute state variation
    $\Delta S[T] \leftarrow S[1:T] - S[0:T-1]$
    Detect permeation events
    events$[T] \leftarrow$ ▷ A permeation event occurs when :
    $\Delta S[T]$ ▷ the water molecule cross a region boundary
    $\cdot(S[T] - \Delta S[T] == 0)$ ▷ the molecule goes from membrane to bulk
    $\cdot(\Delta S[T] \cdot (\text{history}[T]) < 0)$ ▷ the direction of movement points towards the opposite of the last visited bulk region
    for t over T frames **do**
      if $events[t] \neq 0$ **then** ▷ permeation event detected
        Store molecule id i
        Store permeation time t
        Store permeation duration |history$[t]$|
        Store permeation direction diff$[t]$
      end if
    end for
  end for
end for