

Supporting Information: Time Series Modeling of Solute Transport in an H_{II} Phase Lyotropic Liquid Crystal Membrane

Benjamin J. Coscia

Michael R. Shirts

November 22, 2019

S1 Setup and analysis scripts

All python and bash scripts used to set up systems and conduct post-simulation trajectory analysis are available online at https://github.com/shirtsgroup/LLC_Membranes. Documentation for the `LLC_Membranes` repository is available at <https://llc-membranes.readthedocs.io/en/latest/>. Table S1 provides more detail about specific scripts used for each type of analysis performed in the main text.

| Script Name | Section | Description |
|---|---------|---|
| <code>/setup/parameterize.py</code> | 2.1 | Parameterize liquid crystal monomers and solutes with GAFF |
| <code>/setup/build.py</code> | 2.1 | Build simulation unit cell |
| <code>/setup/place_solutes_pores.py</code> | 2.1 | Place equispaced solutes in the pore centers of a unit cell |
| <code>/setup/equil.py</code> | 2.1 | Equilibrate unit cell and run production simulation |
| <code>/analysis/solute_partitioning.py</code> | 2.1 | Determine time evolution of partition of solutes between pores and tails |
| <code>/timeseries/msd.py</code> | 2.2 | Calculate the mean squared displacement of solutes |
| <code>/analysis/sfbm_parameters.py</code> | 2.2 | Get subordinated fractional Brownian motion parameters by fitting to a solute's dwell and hop length distributions and positional autocorrelation function. |
| <code>/timeseries/ctrwsim.py</code> | 2.2 | Generate realizations of a continuous time random walk with the user's choice of dwell and hop distributions |
| <code>/timeseries/forecast_ctrw.py</code> | 2.2 | Combines classes from <code>sfbm_parameters.py</code> and <code>ctrwsim.py</code> to parameterize and predict MSD in one shot. |
| <code>/analysis/markov_state_dependent_dynamics.py</code> | 2.3 | Identify frame-by-frame state of each solute, construct a transition matrix and simulate realizations of the MSDDM model. |

Table S1: The first column provides the names of the python scripts available in the `LLC_Membranes` GitHub repository that were used for system setup and post-simulation trajectory analysis. Paths preceding script names are relative to the `LLC_Membranes/LLC_Membranes` directory. The second columns lists the section in the main text where the output or usage of the script is first described. The third column gives a brief description of the purpose of each script.

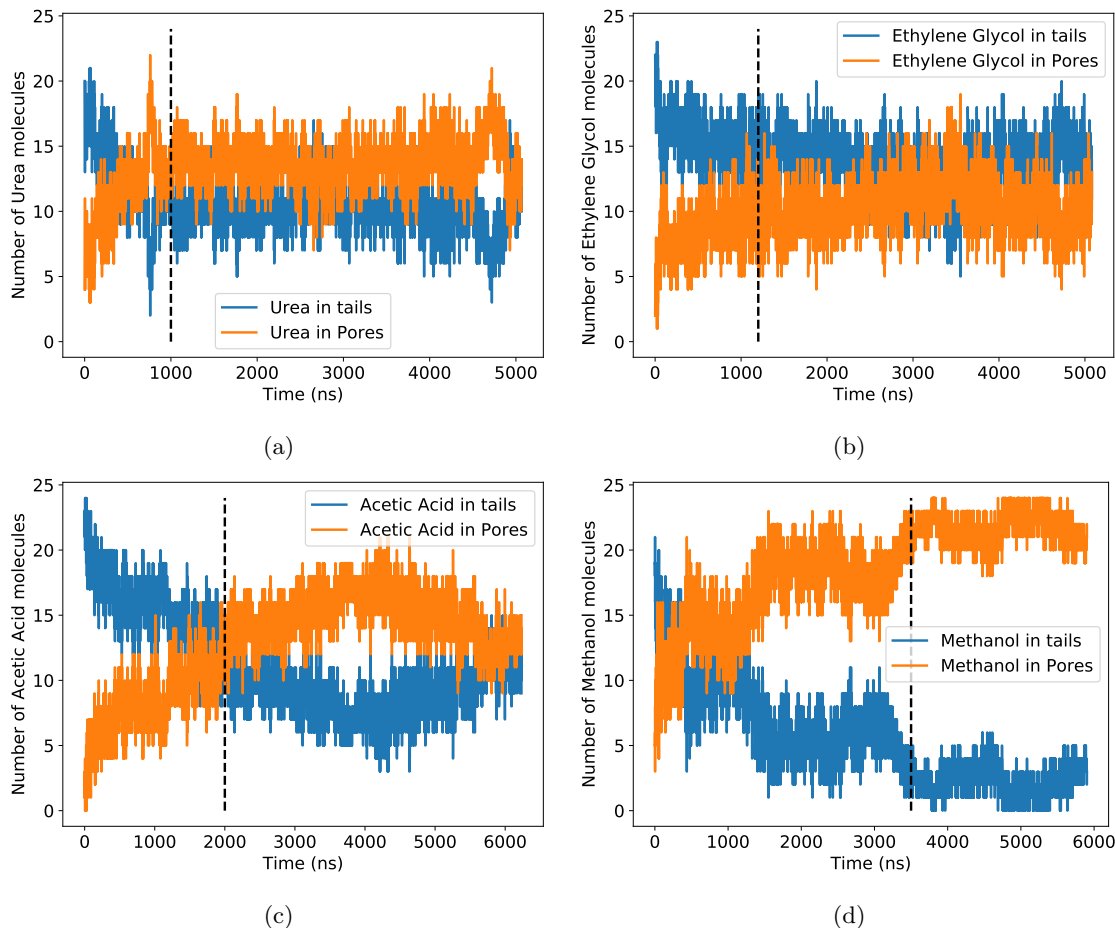


Figure S1: We considered a system to be equilibrated when the partition of solutes between the tails and pore plateaued. Our chosen equilibration point for each solute is indicated by the vertical black dashed line. (a) Urea equilibrates the fastest, after 1000 ns. (b) Ethylene glycol equilibrates after 1200 ns (c) Acetic Acid does not equilibrate until 4400 ns. (d) We considered methanol to be equilibrated after 3500 ns. Methanol nearly completely partitions into the tails.

S2 Solute Equilibration

We collected all data used for model generation after the solutes were equilibrated. We assumed a solute to be equilibrated when the partition of solutes in and out of the pore region stopped changing. The pore region is defined as within 0.75 nm of the pore center. We've plotted the partition versus time in Figure S1 and indicated the chosen equilibration time points.

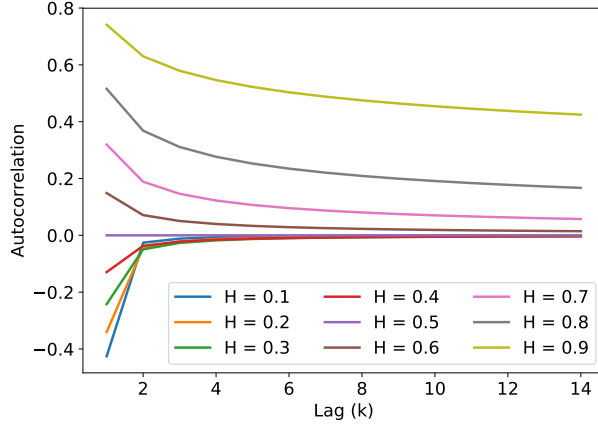


Figure S2: The analytical autocorrelation function of FBM decays to zero faster when $H < 0.5$ compared to when $H > 0.5$.

S3 Estimating the Hurst Parameter

We chose to estimate the Hurst parameter, H by a least squares fit to the analytical autocorrelation function for fractional Brownian motion (the variance-normalized version of Equation 5 in the main text):

$$\gamma(k) = \frac{1}{2} \left[|k-1|^{2H} - 2|k|^{2H} + |k+1|^{2H} \right] \quad (1)$$

In Figure S2, we plotted Equation 1 for different values of H . When $H > 0.5$, Equation 1 decays slowly to zero meaning one needs to study large time lags with high frequency in order to accurately estimate H from the data. Fortunately, all of our solutes show anti-correlated motion, so most of the information in Equation 1 is contained within the first few lags.

The autocovariance function of fractional Lévy motion is different from fractional Brownian motion (see Equations 5 and 7 of the main text), but their autocorrelation structures are the same. The autocovariance function of FLM is dependent on the expected value of squared draws from the underlying Lévy distribution, $E[L(1)^2]$. This is effectively the distribution's variance, which is undefined for most Lévy stable distributions due to their heavy tails. As a consequence, one should expect $E[L(1)^2]$ to grow as more samples are drawn from the distribution with the autocovariance function responding accordingly. However, we are only interested in the autocorrelation function. In order to predict the Hurst parameter from the autocorrelation function, we must show that it has a well-defined structure and is independent of the coefficient in Equation 7 of the main text. In Figure S3, we plot the average autocorrelation function from an FLM process with an increasing number of observations per generated sequence. For all simulations we set $H=0.35$ and $\alpha=1.4$. The variance-normalized autocovariance function, i.e. the autocorrelation function, does not change with increasing sequence length. Additionally, the autocorrelation function of FBM, with the same H , is the same.

S4 Simulating Fractional Lévy Motion

S4.1 Truncated Lévy stable hop distributions

Determining where to truncate the hop distribution: A pure Lévy stable distribution has heavy tails which can lead to arbitrarily long hop lengths. Our distribution of hop lengths fits well to a Lévy distribution near the mean, but under samples the tails. In Figure S5 we compare the empirically measured transition emission distribution of the MSDDM for Urea to its maximum likelihood fit to a Lévy stable distribution. The ratio between the two distributions at each bin is nearly 1 close to the center, indicating a near-perfect fit, larger than 1 slightly further from the center, suggesting that we slightly over sample intermediate hop

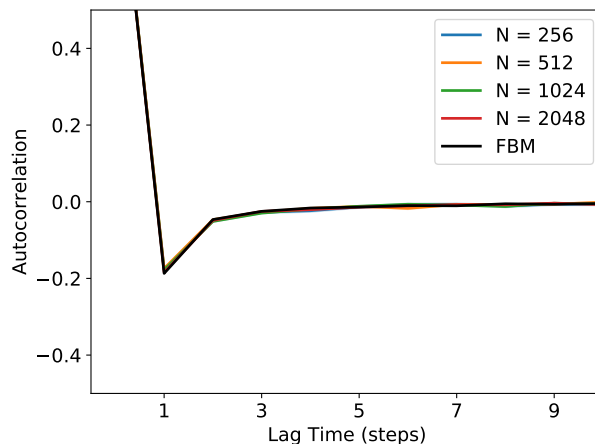


Figure S3: The autocorrelation function of an FLM process does not change with increasing sequence length (N). It shares the same autocorrelation function as fractional Brownian motion (FBM). All sequenced used to make this plot were generated using $H=0.35$ and, for FLM, $\alpha=1.4$.

lengths, and below 1 far from the center, indicating under sampling of extremely long hop lengths. Based on the plot, we chose a cut-off of 0.8 nm in order to compensate for over sampled intermediate hop lengths. We used the same procedure to determine truncation parameters for all solutes.

Generating FLM realizations from a truncated Lévy distribution: To generate realizations from an uncorrelated truncated Lévy process, one would randomly sample from the base distribution and replace values that are too large with new random samples from the base distribution, repeating the process until all samples are under the desired cut-off.

This procedure is complicated by the correlation structure of FLM. At a high level, Stoev and Taqqu use Riemann-sum approximations of the stochastic integrals defining FLM in order to generate realizations. They do this efficiently with the help of Fast Fourier Transforms. In practice, this requires one to Fourier transform a zero-padded vector of random samples drawn from the appropriate Lévy stable distribution, multiply the vector in Fourier space by a kernel function and invert back to real space. The end result is a correlated vector of fractional Lévy noise.

If one is to truncate an FLM process, one can apply the simple procedure above for drawing uncorrelated values from the marginal Lévy stable distribution, but after adding correlation, the maximum drawn value is typically lower than the limit set by the user. Additionally, the shape of the distribution itself changes. Analogous to the database used to correct the Hurst parameter, we created a database to correct the input truncation parameter (the maximum desired draw). The database returns the value of the truncation parameter that will properly truncate the output marginal distribution based on H , α and σ (the width parameter). Figure S5 shows the result of applying our correction. Note that generating this database requires a significant amount of simulation and still likely doesn't perfectly correct the truncation parameter. The output leads to a somewhat fuzzy, rather than abrupt, cut-off of the output distribution. This is likely beneficial since we observe a small proportion of hops longer the chosen truncation cut-off. When the cut-off value is close to the Lévy stable σ parameter, as it is in our anomalous diffusion models, we observed that the tails of the truncated distribution tend to be undersampled. In order to maintain the distribution's approximate shape up to the cut-off value we recommend ensuring that the cut-off value is at least 2 times σ . However, this may lead to a slight over-prediction of the MSD.

S4.2 Achieving the right correlation structure

We simulated FLM using the algorithm of Stoev and Taqqu [1]. There are no known exact methods for simulating FLM. As a consequence, passing a value of H and α to the algorithm does not necessarily result in the correct correlation structure, although the marginal Lévy stable distribution is correct. We applied a database-based empirical correction in order to use the algorithm to achieve the correct marginal distribution

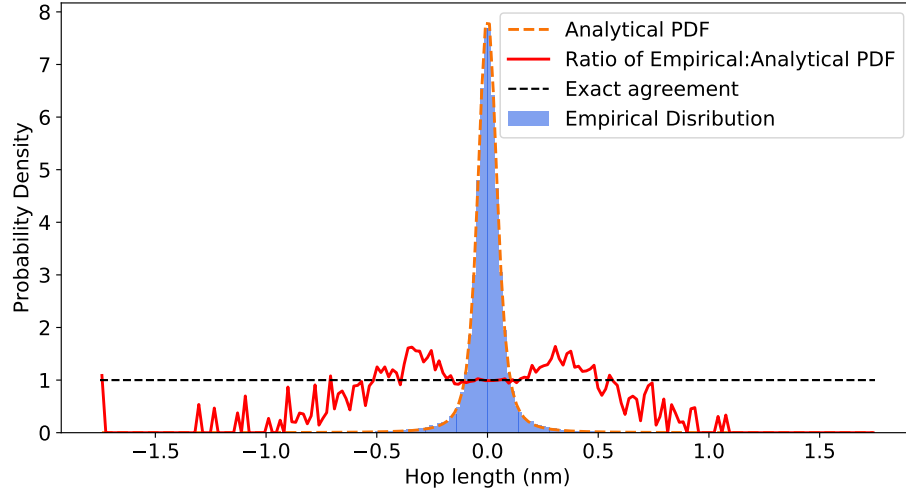


Figure S4: The ratio between the empirical and maximum likelihood theoretical distribution quantifies the quality of fit as function of hop length. The fit is near-perfect close to the mean. Intermediate hop lengths are over sampled, and the tails are under sampled. We used this type of plot to determine the appropriate place to truncate the Lévy stable distributions.

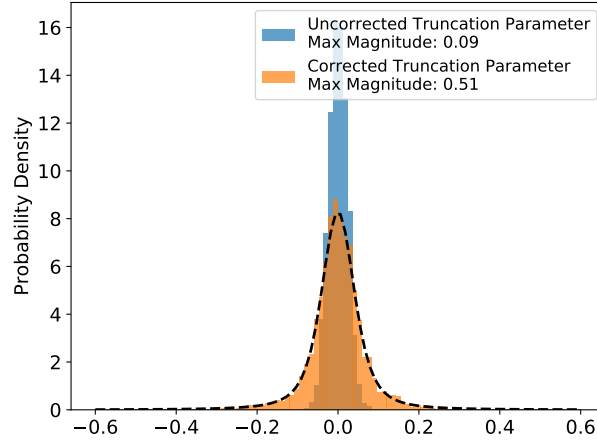


Figure S5: We can accurately truncate the marginal distribution of FLM innovations by applying a correction to the input truncation parameter. We generated FLM sequences and truncated the initial Lévy stable distribution (before Fourier transforming) at a value of 0.5. After correlation structure is added, the width of the distribution of fractional Lévy noise decreases significantly. We corrected the input truncation parameter with our database resulting in a distribution close to the theoretical distribution (black dashed line) with a maximum value close to 0.5.

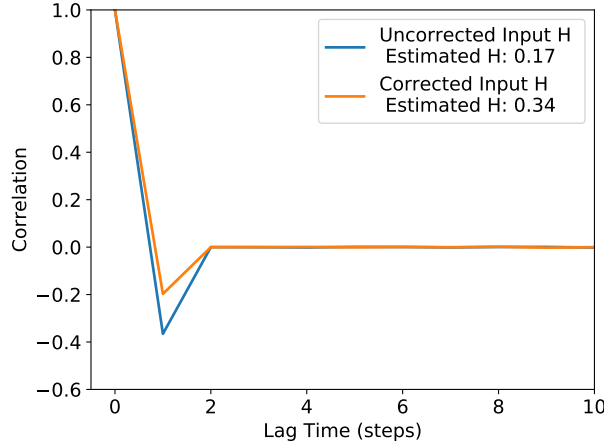


Figure S6: Correcting the Hurst parameter input to the algorithm of Stoev and Taqqu results in an FLM process with a more accurate correlation structure. We generated sequences with an input H of 0.35. We estimated H by fitting the autocorrelation function. Without the correction, H is underestimated, meaning realizations are more negatively correlated than they should be.

and correlation structure.

Stoev and Taqqu note that the transition between negatively and positively correlated draws occurs when $H = 1/\alpha$. When $\alpha = 2$, the marginal distribution is Gaussian and the transition occurs at $H = 0.5$ as expected from FBM. We corrected the input H so that the value of H measured based on the output sequence equaled the desired H . We first adjusted the value of H by adding $(1/\alpha - 0.5)$, effectively recentering the correlation sign transition for any value of $1 \leq \alpha \leq 2$. This correction alone does a good job for input H values near 0.5, but is insufficient if one desires a low value of H . The exact correction to H is not obvious so we created a database of output H values tabulated as a function of input H and α values. Figure S6 demonstrates the results of applying our correction. Without the correction, FLM realizations are more negatively correlated. This would result in under-predicted mean squared displacements when applying the model.

S5 Verifying Markovianity

We verified the Markovianity of our transition matrix, T , in two ways. First we ensured that the process satisfied detailed balance:

$$T_{i,j}P_i(t = \infty) = T_{j,i}P_j(t = \infty) \quad (2)$$

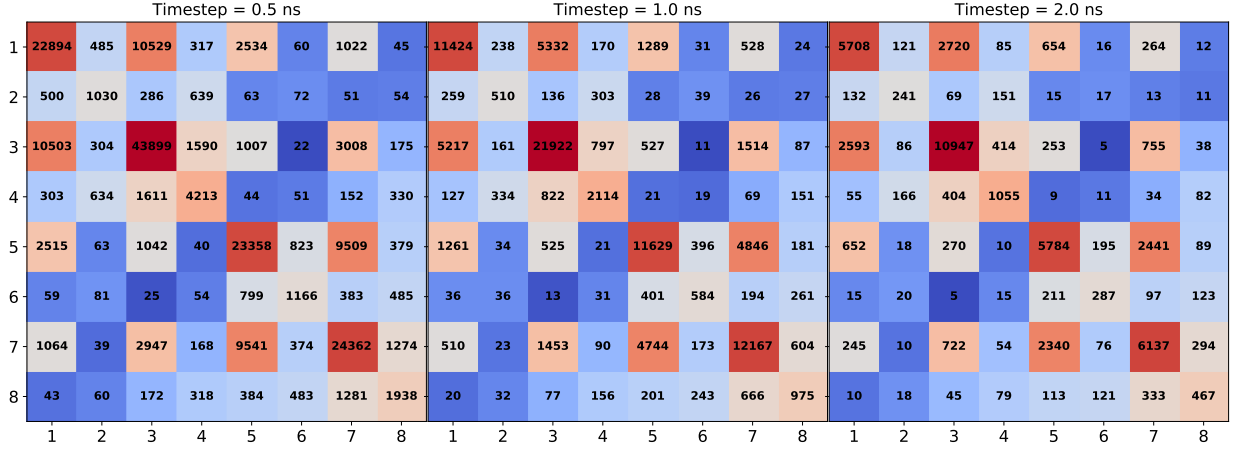
where P is the equilibrium distribution of states. This implies that the number of transitions from state i to j and from state j to i should be equal. Graphical representations of the count matrices show that this is true in Figure S7.

Second, we ensured that the transition matrix did not change on coarser time scales. In Figures S7 and S8, we show that increasing the length of time between samples does not change the properties of the count or probability transition matrices.

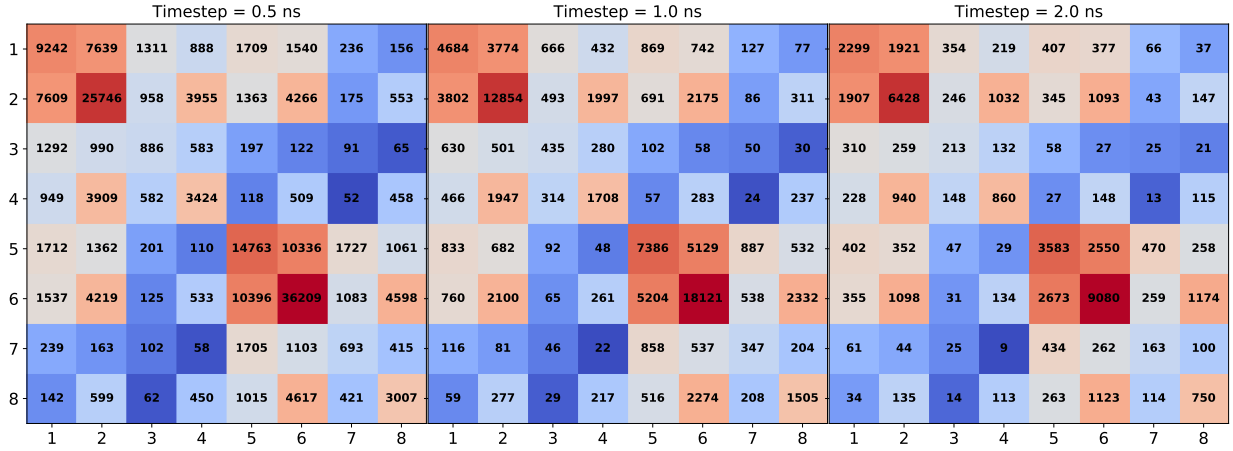
S6 Stationarity of Solute Trajectories

S7 2-state MSDDM parameters

We used a simple 2-state model to illustrate how the self-transition probability effects simulated MSDs generated from the MSDDM. We used the following parameters:

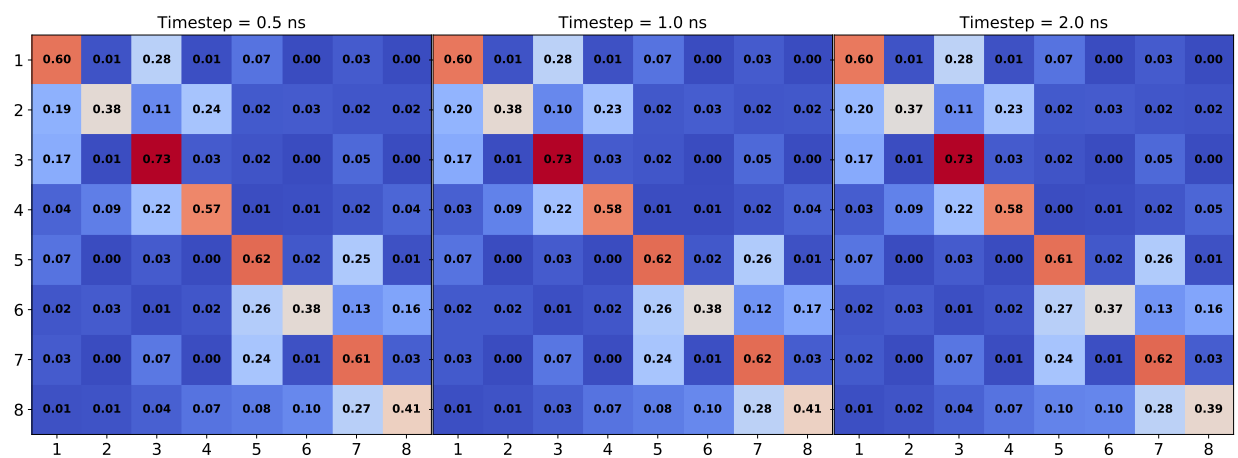


(a) Urea

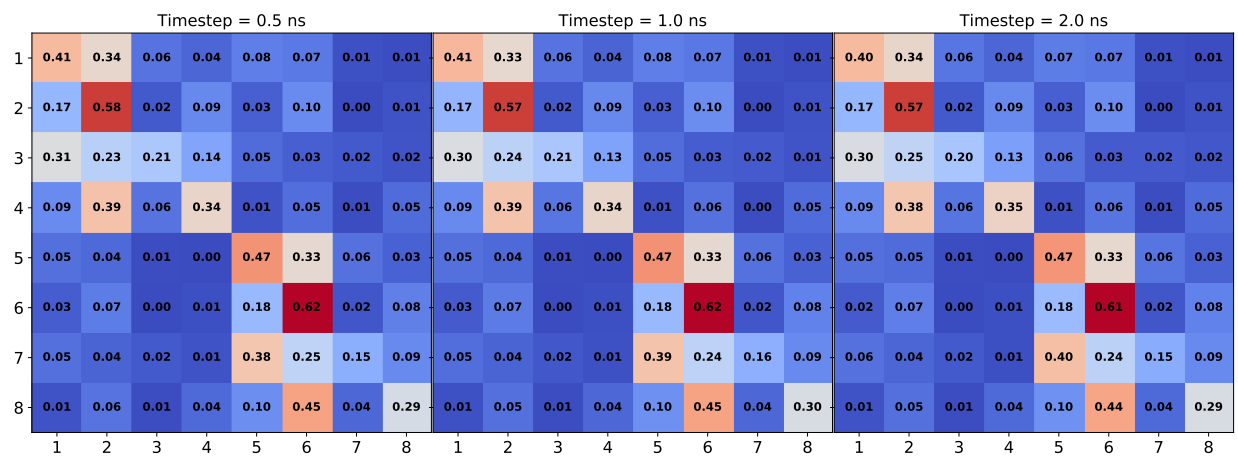


(b) Ethylene Glycol

Figure S7: The number of transitions from state i to j and j to i are very close indicating that our process obeys detailed balance. Detailed balance is conserved for different sized time steps.



(a) Urea



(b) Ethylene Glycol

Figure S8: As the timestep between observations increases, the probability transition matrix does not change significantly.

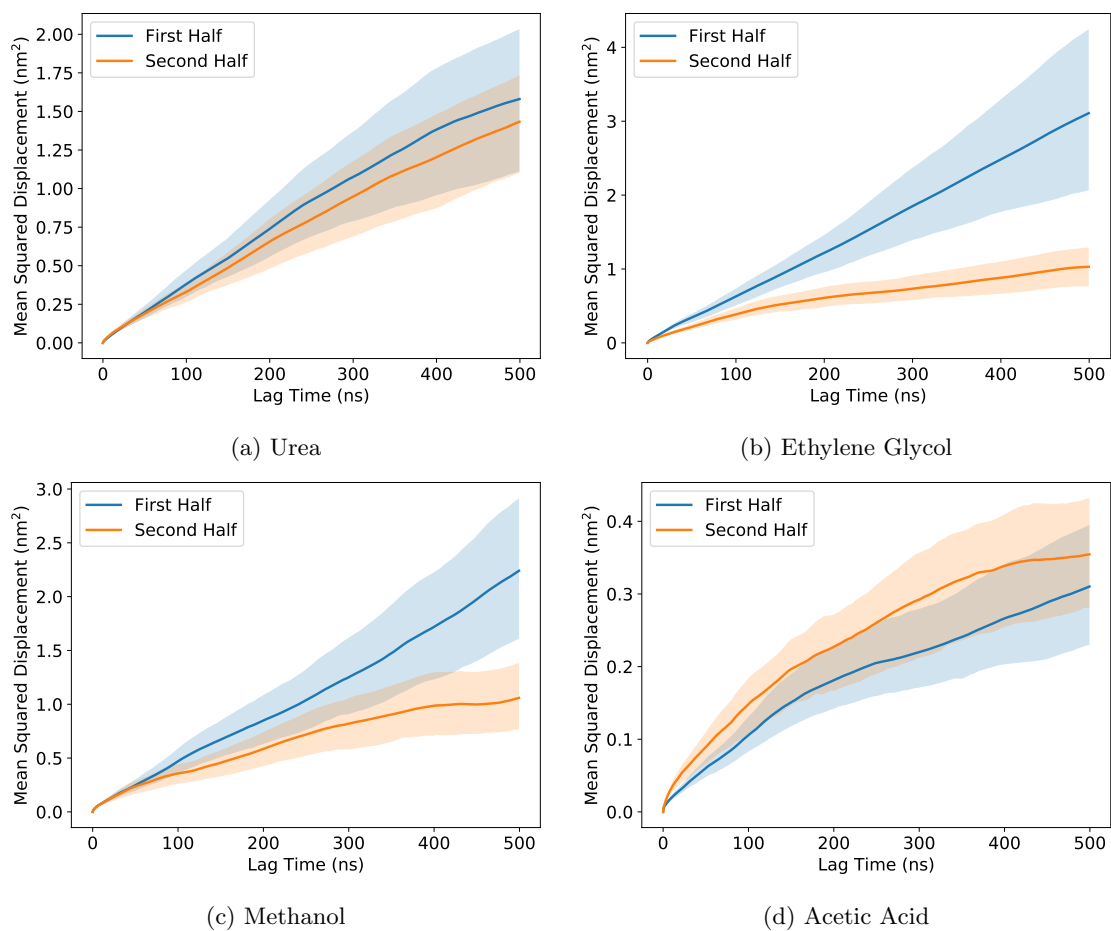


Figure S9: The ensemble of solute trajectories may be stationary if the MSD calculated from different portions of the trajectory are the same. Here we plot the MSD calculated up to a 500 ns time lag of the first and second halves of the equilibrated solute trajectories. Urea and acetic acid have similar MSDs, providing evidence of stationarity, while the MSDs of ethylene glycol and methanol are different.

| State | H | α_h | σ |
|-------|-----|------------|----------|
| 1 | 0.1 | 1.7 | 0.04 |
| 2 | 0.2 | 1.5 | 0.04 |
| T | 0.4 | 1.4 | 0.04 |

Table S2: Values of H , α_h and σ used to simulate realizations of the simple MSDDM whose MSDs are shown in Figure 9 of the main text. The values are chosen to be similar to those in Table 6 of the main text.

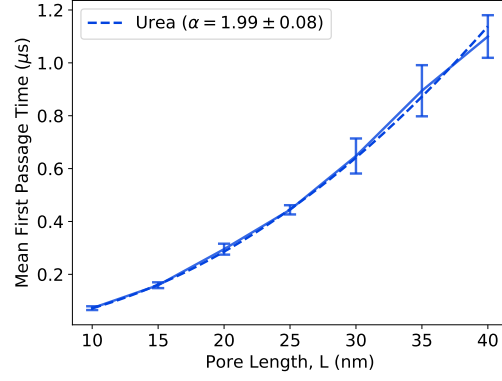


Figure S10

S8 Mean first passage time simulations

References

- [1] S. Stoev and M. S. Taqqu, “Simulation methods for linear fractional stable motion and farima using the fast fourier transform,” *Fractals*, vol. 12, pp. 95–121, Mar. 2004.