Molecular Dynamics Study of Proton Transfer Reactions in GROMACS

Özge Yoluk

Degree project in applied biotechnology, Master of Science (2 years), 2011
Examensarbete i tillämpad bioteknik 45 hp till masterexamen, 2011
Biology Education Centre and Department of Cell and Molecular Biology, Uppsala University
Supervisors: David van der Spoel and Erik G. Marklund
Dedicated to my beloved parents;  
Emirali and Fidan Yoluk
Figures

Figure 1. Grothuss mechanism. Green dashed lines indicate the transfer. ........................................ 1
Figure 2. One dimensional potential energy profile of a proton-transfer reaction. ..................... 3
Figure 3. Schematic representation of the regimes in proton-transfer reactions. ................... 4
Figure 4. Schematic representations of the molecules and proton-transfer routes. ................. 8
Figure 5. Flowchart of a step in Q-Hop algorithm ....................................................................... 8
Figure 6. A simple flowchart for a molecular dynamics simulation with GROMACS .......... 10
Figure 7. Empirical cumulative distribution function plots with the first (right) and the
second (left) set of reactions ........................................................................................................ 12
Figure 8. Radial distribution function of inert hydronium in different water models with or
without proton-transfer .............................................................................................................. 13
Figure 9. RDF of Cl-OW (I) and OW*-OW (II) in simulations with or without Cl- ............. 14
Figure 10. Number of acetic acid molecules over time .............................................................. 14
Figure 11. E12 over time for glutamate dipeptide. .................................................................... 15
# Contents

Figures ..................................................................................................................................................... i

Abstract ................................................................................................................................................. iv

1. Introduction ....................................................................................................................................... 1
   1.1. Q-Hop Algorithm ....................................................................................................................... 2
   1.2. Notes on Q-Hop implementation to GROMACS: Improvements ......................................... 4
   1.3. Treatment of Transfer Probabilities ......................................................................................... 4
       1.3.1 Gillespie Algorithm ............................................................................................................. 5
       1.3.2. Randomized List Approach .............................................................................................. 5

2. Methods .............................................................................................................................................. 7
   2.1 Treatment of Transfer Probabilities: Method & Systems ............................................................. 7
   2.2 Study of Proton-transfer Reactions in GROMACS: Method & Systems ....................................... 8
       2.2.1. Excess Proton in Water ..................................................................................................... 9
       2.2.2. Hydronium/Cl\textsuperscript{-} System .................................................................................. 10
       2.2.3. Ammonium/Ammonia & Acetic Acid Simulation Systems ............................................... 10
       2.2.4. Dipeptides ......................................................................................................................... 11

3. Results .............................................................................................................................................. 12
   3.1. Treatment of Transfer Probabilities .......................................................................................... 12
   3.2. Excess Proton in Water ............................................................................................................. 12
   3.3. Hydronium/Cl\textsuperscript{-} Simulations ....................................................................................... 14
   3.4. Ammonium & Acetic Acid Simulations ..................................................................................... 14
   3.5 Dipeptide Simulations ............................................................................................................... 15

4. Discussion ......................................................................................................................................... 16

Acknowledgements .............................................................................................................................. 18

References ............................................................................................................................................ 19

Appendix .............................................................................................................................................. 21
Abstract

Proton-transfer is central for many biological phenomena and has an importance in many applications/methods. Several methods have been developed in order to analyze the proton-transfer reactions computationally. However so far developed methods were computationally expensive, limited for small scales or not sufficiently accurate. A fast algorithm to derive proton-transfer reaction rates, Q-Hop, is being implemented to fast, reliable molecular dynamics software, GROMACS by David van der Spoel’s group at Uppsala University. In this report, examples of oxygen-oxygen transfer and nitrogen-oxygen transfer reactions were analyzed with molecular dynamic simulations using GROMACS software. Also the distribution generated by randomized list approach, which was introduced in GROMACS implementation of Q-Hop, to treat the donor/acceptor networks has been tested. Radial distribution functions and hopping rates were in general agreement with previous studies. However, diffusion coefficients and estimated pKa values indicates the need for improvements. It was shown that randomized list approach generates distributions similar to distributions created by Gillespie algorithm and should be preferred to treat the donor/acceptor networks instead of sorted list approach.
1. Introduction

It has been more than 200 years since Theoders Grotthuss proposed a now widely accepted theory for proton-transfer in liquid water without being aware of the existence of the proton\(^1\). He was explaining the theory behind Volta’s electrolysis experiments when he drew a picture of the, as it is called today, Grotthuss mechanism (Marx 2006).

Today we know that proton-transfer in liquid water involves neither migration of $\text{H}_3\text{O}^+$ nor migration of $\text{H}^+$. Proton-transfer (in water) refers to a sequence of proton-transfer reactions between water molecules (proton hopping) (Figure 1). To be more specific it is a series of isomerization between $\text{H}_9\text{O}_4^+$ (Eigen) and $\text{H}_5\text{O}_2^+$ (Zundel) cations (Agmon 1995).

Proton-transfer is a common biological/chemical reaction that plays an important role in many aspects of life, from enzymatic processes to viral entry. Dengue virus undergoes a reversible conformational change at low pH. Those changes in conformation render furin cleavage, which is an important process in order for the virus to enter the host cell (Modis et al. 2004; Yu et al. 2008).

Apart from being central to many biological phenomena, proton-transfer has an importance in many applications/methods as well. Mass spectrometry (MS) methods, measure the characteristics of molecular ions based on their mass-to-charge ratio in the gas phase. Hence, generation of (macro)molecular ions in the gas phase is an essential part of MS experiments. It was shown that in electrospray ionization (ESI) method non-covalent interactions stay intact (Patriksson et al. 2007). Therefore MS is becoming more popular amongst structural biologists. A recent review on macromolecular structures in the gas phase by Van der Spoel et al. (2011), draws a picture of the field by analyzing outcomes of the experiments together with observations from simulation studies. They highlight the conserved structural similarity of proteins in vacuo and in solution, but point out differences in terms of hydrogen bonding fraction. Methods like ESI, works through lowering the pH and this will most likely result with different protonation states for amino acid side chains (Patriksson et al. 2007; van der Spoel et al. 2011). In order to predict the structural differences of proteins between gas phase and in solution computationally, proton-transfer reactions should be allowed in simulations.

\[^1\] Translation from Grotthuss memory 1805: “It is clear that in the whole operation the molecules of water, situated at the extremities of the conductor wires, will alone be decomposed, whereas all those placed intermediately will change reciprocally and alternatively their component principles without changing their nature. … all the molecules of the liquid situated in this circle would be decomposed and instantly recomposed …” (Marx 2006)
Study of proton-transfer reactions has seen a large interest for several decades. Molecular dynamics simulations are mainly performed with a fixed protonation state, but several computational methods are present today to study proton-transfer; Car-Parinello molecular dynamics (CPMD) method is perhaps the most important one among them (Tuckerman et al. 1997; Marx et al. 1999).

Despite the fact that ab initio studies with those methods made major contributions to our knowledge of proton-transfer reactions, many of those methods are computationally expensive and usually restricted to small systems. In 2001, Lill and Helms, proposed an algorithm called Q-Hop, to derive proton-transfer rates quantum mechanically (Lill et al. 2001a; Lill et al. 2001b; Lill et al. 2001c). A fast and reliable approach allows study proton-transfer reactions in larger systems. An implementation of Q-Hop algorithm in the NWChem molecular dynamics package is present and there have been several studies (Lill et al. 2002; Gu et al. 2007; Wei et al. 2007).

In order to make the simulations faster and more reliable based on the studies of Lill and Helms, members of David van der Spoel’s group, have been working to incorporate the Q-Hop algorithm with major improvements to the fast molecular dynamics software package GROMACS. Implementation is currently being finalized and a manuscript is in the making. The aim of this study was to study the most common biological reaction, proton-transfer, using GROMACS software starting from simple systems such as excess proton in water and later moving on to more complex systems.

1.1. Q-Hop Algorithm

A proton-transfer reaction typically has a double-well shape potential energy surface (Figure 2). $E_{\text{min.1}}$ and $E_{\text{min.2}}$ represent the states where proton is bound to the donor ($E_{\text{min.1}}$) or the acceptor ($E_{\text{min.2}}$). $E_{12}$ is the difference between two minima and $E_{b}^{\rightarrow}$ is the energy barrier (forward). The energy barrier is highly affected by the distance between donor and acceptor atoms (Lill et al. 2000). Starting from the double-well shape potential energy surface, Lill and Helms showed that a fast estimation of proton-transfer rates is possible (Lill et al. 2001a). A brief explanation of the Q-Hop algorithm is given below.

$E_{12}^0$ is computed with measured R(DA); donor-acceptor distance, using three fitted parameters ($\alpha$, $\beta$, $\gamma$) that are driven from quantum chemistry calculations for many possible donor-acceptor pairs, Equation I. After adding the environmental effect $E_{\text{env}}$ to $E_{12}^0$, the probability of each possible transfer reaction is calculated.

$$E_{12}^0 = \alpha + \beta R(DA) + \gamma R(DA)^2$$  \hspace{1cm} \text{Eq. I}

Energy barrier of proton-transfer reactions tends to disappear at small distances (Lill et al. 2000). Therefore one needs more than one approach to calculate probabilities, since energy barrier is expressed as a function of $E_{12}$ and $R(DA)$, simple transition state theory is not sufficient to cover all transfer reactions. In Q-Hop proton-transfer probabilities are grouped into three regimes based on $R$ and $E_{12}$ (Figure 3): Transition state theory (TST), Schrödinger Equation (SE) and the regime in between the validity limits of TST and SE. For every regime parameters $\alpha$, $\beta$ and $\gamma$ is needed.

---

*Environmental effect is the term that takes the effect of the donor-acceptor pair’s surrounding molecules into account.*
1. **TST regime**: Transfer probability is calculated with Eq. II for high R and E12 values. Transfer rate is calculated with 
\[ P_{TST} = \kappa(T) \left( \frac{k_B T}{2\pi \hbar} \right)^{3/2} \exp\left(-\frac{E_b - \hbar \omega/2}{k_B T}\right) \] 
where \( \hbar \omega/2 \) is quantum correction). Enhancement in classical reaction rate due to tunneling effect is accounted in the term \( \kappa(T) \) (Lill et al. 2001a; Lill et al. 2001c).

2. **SE regime**: For small donor-acceptor distances, barrier and second minima decrease and sometimes disappear; transition state theory is not valid anymore. In those cases, probabilities are calculated with Eq. III. Quantum mechanically computed energy profiles for proton transfer are used to calculate the initial wave function of Schrödinger equation. Using the initial wave function, time evolution of probability density to find the proton on the right side of the midpoint between donor and acceptor is examined to determine the proton transfer rate (Lill et al. 2001c). Shortest time recorded for probability to reach its maximum is 11fs; therefore 10fs window is used with proton transfer simulations in Q-Hop. Fitted parameters \( K \) and \( M \) are dependent on \( R(DA) \).

3. **Gap**: Regime where none of the theories above applies. To fill this gap, a linear interpolation on logarithmic scale has been applied. \( E_{12}^L \) and \( E_{12}^R \) are the validity limits of the two approaches. (Lill et al. 2001c)

\[ \log P_{GAP} = \log P_{TST}(E_{12}^L) + \frac{\log P_{TST}(E_{12}^R) - \log P_{SE}(E_{12}^L)}{E_{12}^R - E_{12}^L} \log \left(\frac{E_{12}^R}{E_{12}^L} - 1\right) \] Eq. IV

First, the probabilities are calculated with one of the three methods above, and then a sorted list of probabilities is compared against a generated random number. If the probability is higher than the random number, protonation states for that donor-acceptor pair is altered such that proton moves from donor to acceptor. It should be noted that in Q-Hop algorithm a lag.
time is introduced, time after a successful hop when no hopping is allowed. However this is not used in GROMACS implementation.

### 1.2. Notes on Q-Hop implementation to GROMACS: Improvements

The implementation of Q-Hop in GROMACS includes major improvements to deal with issues when original algorithm was not sufficiently accurate:

- Treatment of donor/acceptor pairs: Using a randomized list instead of a list in decreasing order (see 1.3).
- Correction of the validity limits for TST regime: Validity limit of TST regime is changed from \( \exp(\beta E_b) > 100 \) to \( \exp(\beta E_b - \hbar \omega / 2) > 100 \) since the actual energy barrier to overcome is \( E_b - \hbar \omega / 2 \).

- To prevent double counting of energies, calculation of \( E_{12} \) is altered so that coulomb interactions which effects the term \( E_{env} \) can be scaled
- No lag time used.

### 1.3. Treatment of Transfer Probabilities

As mentioned in section 1.2, in Q-Hop algorithm; next proton-transfer reaction to fire is chosen from a list of probabilities which are sorted on a decreasing order. This approach decreases the actual probability of the other reactions that comes later on the list.

To create a correct distribution, another approach has been introduced in GROMACS. The list of transfer probabilities is randomized before the comparison against a random number. This is a shortcut approach to the Gillespie algorithm.

The Gillespie algorithm was introduced by Daniel Gillespie in 1976 to generate correct distribution of the master equation by numerical simulation which involves Monte-Carlo techniques (Gillespie 1976; Gillespie 1977):

\[
\frac{dP_k}{dt} = \sum_l T_{kl}P_l
\]

where \( P \) stands for the probability. \( k \) and \( l \) shows the state of the system and \( T \) stands for the transfer probability from \( k \) to \( l \).

Gillespie algorithm has two methods to implement the Monte-Carlo step: the “first reaction method” and the “direct method”. Below, a brief explanation of both the Gillespie algorithm and the randomized list approach can be found.
1.3.1 Gillespie Algorithm

Both methods was shown to yield similar results (Gillespie 1976). The difference is that first-reaction method can be computationally more expensive for complex systems due to generation of random numbers for every possible reaction. As for direct method, only two random numbers are generated in each step. Although first-reaction method requires more time/memory usage, for Q-Hop method generation of probabilities is far more computationally expensive.

First-Reaction Method

First reaction method describes, for a system with reactions \( \{1, 2, 3 \ldots M\} \) at time \( t \), the probability of a reaction \( R_v \) to occur at time interval \( (\tau + t, \tau + t + d\tau) \) as:

\[
P(\tau) d\tau = \exp(-a_v \tau) a_v d\tau
\]

where \( a_v \) is the reaction propensity. From this equation a tentative reaction time \( \tau_v \) is generated for all reaction \( \{R_v\} \):

\[
\tau_v = (1/a_v) \ln(1/r_v) \quad (v = 1, 2, 3 \ldots M)
\]

where \( r_v \) stands for generated random number.

From these \( \{1, 2, 3 \ldots M\} \) reactions the one that occurs first, thus with smallest \( t_v \), is chosen as the next reaction. \( t_v = \tau \) and system time is updated with by \( t + \tau \). \( v \) of the reaction with smallest \( t \) corresponds the integer value \( (\mu) \) of next reaction. It should be noted that, random number is generated for every reaction with this method.

Direct Method

The direct method uses a process called “conditioning” and expresses the probability of a \( \mu \) reaction to occur at time \( t \) as:

\[
P(\tau|\mu) = P_1(\tau). P_2(\mu|\tau)
\]

where \( P_1(\tau) \) stands for the probability that the next reaction will occur between time interval \( (\tau + t, \tau + t + d\tau) \) and \( P_2(\mu|\tau) \) stands for that next reaction to be \( \mu \). From \( P(\tau) \) a random \( t \) value is generated by drawing a random number and solving equation like in first reaction method but using only one random number for all reactions:

\[
\tau_v = (1/a) \ln(1/r_1)
\]

Where \( a \) is the sum of all the reaction rates and \( r_1 \) is the random number.

Then with a second random number, the reaction integer \( \mu \) generated according to \( P_2(\mu|\tau) \):

\[
\sum_{v=1}^{\mu-1} a_v < r_2 a \leq \sum_{v=1}^{\mu} a_v
\]

1.3.2 Randomized List Approach

One of the improvements to Gillespie algorithm was in the step where the next proton-transfer reaction was chosen.

In Q-Hop as developed by Lill and Helms, probabilities are sorted in decreasing order before the comparison with a random number. The reaction with the highest probability would have a higher actual probability against other reactions. By not generating the right distribution between possible reactions, a problem arises when two similar possible transfer reactions are present since probability for those two reactions are expected to be similar as well. In
GROMACS implementation of Q-Hop, randomized list approach has been introduced in order to avoid this problem. Intrinsic probabilities are expressed as a function of reaction rate $k$ for possible transfer reactions in a 10fs window:

$$P(10) = 1 - e^{-k10fs}$$

Eq. XI

After that, instead of following Gillespie algorithm, since the only interest is the state at 10fs, the list of probabilities is processed in a random order for the comparison with a random number.

In this study to demonstrate the distributions generated by Gillespie algorithm, randomized list and sorted list approaches, a python script was written (see Appendix I).
2. Methods

2.1 Treatment of Transfer Probabilities: Method & Systems

*Gillespie Algorithm:* Both methods of Gillespie algorithm, direct method and first-reaction methods were used for comparison of the observed distributions with randomized and sorted list approaches.

The steps that had been followed for Gillespie algorithm are as follows:

1. Quantity of each molecule and reaction propensities are initialized.
2. Random numbers are generated to compute the time interval and determine the next reaction to happen.
3. System time is updated with the computed time interval. Molecule quantities are updated based on the chosen reaction in step 2.
4. If not all the reactants are consumed, system goes back to step 1 with new molecule quantities and reaction rates.
5. System continues to run from step 1 to 4 until it exceeds the specified certain time.

In Gillespie algorithm reaction propensities are used in order to choose the next reaction. Therefore randomly drawn probabilities are converted to reaction rates based on Equation XI and used as reaction propensity assuming that the reaction propensity is equal to reaction rate in a first order reaction.

Since only the situation in 10fs is our interest and shortest transfer event is 10fs when selected reaction had a time interval greater than 10fs, molecule quantities were not altered and system time updated with 10fs.

*List Approaches:* Randomly drawn probabilities were stored in a list. For randomized list approach, list of the probabilities shuffled using the random module of python. For sorted list approach, list of the probabilities sorted in decreasing order. Then shuffled/sorted list was compared with a random number, starting from the first element on the list. If the reaction is successful (probability being greater or equal than the random number) system continues with altered molecule quantities and thus generating new probabilities for the new set of reactions. If the reaction is not successful, next element on the list tested against a new random number. Probabilities for each reaction were drawn randomly, from random module of python once and were kept same for all methods. All the methods were started with the same molecule quantities. System time updated with 10fs after each step which allows only one trial for a reaction to be chosen.

Two sets of reactions were used in order to generate examples of proton-transfer (Figure 4). In the first set, donor atom was kept the same, and several acceptors could accept proton from donor. Ending time was set to 10fs for the first set which means maximum one trial for a reaction to be chosen is allowed. In the second set a small reaction network is used (see Appendix II for the complete set of reactions, with randomly generated probabilities for each
reaction). Script was run with ending time 30fs for the second set which allows maximum three attempts for a reaction to be chosen.

2.2 Study of Proton-transfer Reactions in GROMACS: Method & Systems

A schematic representation of the Q-Hop algorithm steps is shown in flowchart Figure 5. In every specified transfer frequency (can be set in parameters file) molecular dynamic step, distances between possible donor-acceptor pairs are measured. Environmental effect ($E_{env}$) is computed. Based on $E_{12}$ and $R(DA)$, probability of transfer is calculated with one of the three methods described in 1.1. The probability of each possible reaction in current step is compared against a random number starting with the first element of the list. If all the probabilities for that step are lower than the random number, simulation goes back to the md steps without any alterations to the system. If the probability is higher than the random number, a hop takes place between that donor-acceptor pair. Protonation state of that pair is changed and simulation goes on with new protonation state.

The parameters that are needed in order to perform proton-transfer simulations were obtained from: (Herzog et al. 2006); (Lill et al. 2001a; Lill et al. 2001c).

---

**Figure 4.** Schematic representations of the molecules and proton transfer routes. I) Starting condition, one donor atom (A) and several acceptors. Only one transfer event is allowed. II) Starting condition, donor atom A and acceptors B,C,D. Max. 3 transfer events are allowed. After the first transfer, possible acceptors vary depending on the chosen acceptor in the first step.

**Figure 5.** Flowchart of a step in Q-Hop algorithm
A flowchart for a typical MD simulation with GROMACS is shown in Figure 6. For a simulation with Q-hop, coordinate files and topology files should be constructed for protonated state where all the hydrogen atoms present on donor/acceptor atoms. Topologies used in simulations were constructed based on AMBER99SB force field including simulations with proton-transfer reactions. Simulation box type is cubic for all systems and quippable SPC water used as a solute, unless otherwise stated.

All test systems were energy minimized using steepest descent with 1000 steps. All simulations performed on ANFINSEN\textsuperscript{iii}, with .mdp options below, unless otherwise stated:

- plain cut-offs for electrostatics and VdW interactions (with distance 1.2 nm)
- userreal1=0.7 (Coulomb scaling option, see section 1.2)
- 0.0005ps timestep
- Temperature coupling: Velocity-rescale (0.5ps, 300K)
- qhopmode=randlist (Randomized list; option to choose how to treat donor/acceptor networks)
- qhopfreq=20 (option to set the frequency for the computation of proton-transfer)

To tell the mdrun (MD program)\textsuperscript{iv} in which protonation state systems are H\_exist group in an index file must be provided to grompp. H\_exist group contains the hydrogen atom indices that are present on donors/acceptors in a certain state (protonated or deprotonated).

\[
\text{NH}_4^+ + \text{H}_2\text{O} \rightleftharpoons \text{NH}_3 + \text{H}_3\text{O}^+
\]

In example, for the reaction above if a simulation of ammonium (\text{NH}_4\textsuperscript{+}) in water is wanted to performed, H\_exist group would contain four hydrogen of ammonium, plus the two hydrogen of water. But if the desired state is ammonia (\text{NH}_3), then three hydrogen of ammonia, plus the three hydrogen of hydronium should be present in H\_exist group.

Proton-transfer reactions can be grouped based on donor and acceptor atoms: oxygen-oxygen (O-O), nitrogen-oxygen (N-O) (or vice versa) and nitrogen-nitrogen (N-N). In this study examples of O-O transfer and N-O transfer reactions have been studied. An excess proton in bulk water and glutamate dipeptide were chosen as examples for O-O transfer reactions; ammonium (\text{NH}_4\textsuperscript{+}) and lysine dipeptide were chosen as examples for N-O, O-N transfer reactions. Amino and carboxyl groups of amino acids can be (de)protonated, in order to prevent the (de)protonation event in the termini of the peptide both dipeptides were capped with N-methyl and acetyl groups. Also not all the amino acids are supported as terminal residues in Amber force field.

Analysis of the simulations was done mainly with tools available in GROMACS.

\subsection*{2.2.1. Excess Proton in Water}

Three provided water models (SPC, SPC/E, and TIP3P) that were modified for Q-Hop by addition of 3 virtual sites for hydronium hydrogens were used in a small test system to compare the radial distribution function (rdf) and diffusion coefficient (D) with or without Q-Hop (For the differences between water models (van der Spoel et al. 1998) ). To measure rdf 10ps simulations, to measure D 500ps simulations were performed. The simulation box contained 863 water molecules and one hydronium. Density: 1019g/l

\textsuperscript{iii} Linux cluster of ICM, Uppsala University
\textsuperscript{iv} Main computational chemistry engine within GROMACS to run MD simulations.
To measure diffusion coefficient; two sets of 500ps length 16 simulations each with different starting velocities were prepared one with Coulomb scaling (CS) set to 0.7 and the other one with CS set to 1. In both setups qhopfreq was set to 5 fs and timestep of 0.001ps was used. The simulation box contained 863 water molecules and one hydronium. Density: 1019g/l

![Flowchart for molecular dynamics simulation with GROMACS](image)

Figure 6. A simple flowchart for a molecular dynamics simulation with GROMACS

To demonstrate the effect of electrostatic cut-offs system with CS 0.7 was run with two different cut-off setups: Particle-Mesh Ewald (PME) and plain cut-off. Cut-off distance was set to 1.2 in both simulations.

10ps simulation with plain cut-offs and down scaled coulomb was prepared to obtain rdf peaks of qhoppable hydronium oxygen (OW*).

### 2.2.2. Hydronium/Cl⁻ System

1 Cl⁻ was solvated in a box contains 862 water molecules and one hydronium. Simulation time was 500ps with time step 0.001ps. Cut-off distance was set to 0.9nm and qhopfreq to 5 steps. Density: 1019 g/l

To measure rdf, 10ps simulations were prepared using the same parameters as longer run.

### 2.2.3. Ammonium/Ammonia & Acetic Acid Simulation Systems

#### Ammonium/Ammonia

One ammonium (NH₄⁺) molecule placed in a simulation box contains 1 Cl⁻ and 498 water molecules. Simulation time was 1ns. Density: 962 g/l

Same simulation box was used as in NH₄⁺ system, except hydrogen atoms provided in H_exist group in index file was changed so that simulation start with ammonia (NH₃) and one hydronium. System simulated for 100ps.
Acetic Acid

5 acetic acid molecules were placed in a simulation box contains 656 water molecules. Concentration of acetic acid in the simulation box was 0.53M. Simulation time 100ps. Density: 865 g/l

2.2.4. Dipeptides

Dipeptides were constructed using PyMol version 1.2r2. They were capped in both ends, using residues N-methyl (NME) for C-terminus and acetyl (ACE) for N-terminus. To compensate total charges of the system ions were added, when it is necessary.

Table 1. Dipeptide Simulation Systems

<table>
<thead>
<tr>
<th>Dipeptide</th>
<th>Simulation Time (ps)</th>
<th>Ions</th>
<th>Density (g/l)</th>
<th>Nr. Of Water Molecules</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-E</td>
<td>500</td>
<td>None</td>
<td>988</td>
<td>1143</td>
</tr>
<tr>
<td>E-K</td>
<td>500</td>
<td>1 Cl</td>
<td>970</td>
<td>1250</td>
</tr>
</tbody>
</table>
3. Results

3.1. Treatment of Transfer Probabilities

It is expected for the number of times where a reaction \( \mu \) is preferred to be proportional to the reaction probabilities. To demonstrate the difference/similarity between different methods, how many times reactions were chosen, was counted. Counted data was analyzed with empirical cumulative distribution function in R. Results of 10000 runs of the script are shown in Figure 7. As can be seen in ecdf plots, randomized list approach generates similar distributions as Gillespie algorithm. On the other hand sorted list approach yields quite different results.

![Empirical cumulative distribution function plots with the first (right) and the second (left) set of reactions.](image)

*Figure 7.* Empirical cumulative distribution function plots with the first (right) and the second (left) set of reactions. x: number of times a reaction was chosen. Random L.: Randomized list approach, Sorted L.: Sorted list approach, Direct M.: Direct method of Gillespie algorithm, F-R M.: First-Reaction method of Gillespie algorithm (distributions of both Gillespie methods overlaps).

Using time as a limiting factor for Gillespie methods created a similar situation with list approaches when no reaction was chosen due to the random numbers being higher than the reaction probabilities.

3.2. Excess Proton in Water

Without Q-hop, radial distribution function of inert hydronium oxygen(OW(i)) with water oxygens (OW) were similar in all three water models with a maximum around 2.5Å (Figure 8). Although the peaks were slightly shifted (with +/- 0.4Å) for SPC/E and TIP3P,
simulations with Q-hop showed similar maxima as without Q-hop. Also values of the peaks differed between simulations with or without Q-hop. Difference for TIP3P water model was 0.3 and for SPC/E and SPC water models were 0.5, though the peak area itself is more important to compare with the experimental data. Rdf of hoppable hydronium oxygen (OW*) with OW was similar with OW(i) peak in SPC water simulation, peak around 2.5Å with a value of ~5 (Figure 9).

**Table 2.** D and hopping rates of excess proton

<table>
<thead>
<tr>
<th>Simulation System</th>
<th>D(x10^5 cm^2/s)</th>
<th>Hopping rate (hop/ps)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cut-off/CS1</td>
<td>25</td>
<td>0.88</td>
</tr>
<tr>
<td>Cut-off/CS0.7</td>
<td>16</td>
<td>2.3</td>
</tr>
<tr>
<td>PME/CS0.7</td>
<td>21</td>
<td>2.1</td>
</tr>
</tbody>
</table>

Figure 8. Radial distribution function of inert hydronium in different water models with or without proton transfer. Np: without proton transfer, p: proton transfer.

Diffusion coefficient was averaged over 16 simulations each 500ps length with Einstein equation:

\[ D = \frac{\langle [r(t) - r(0)]^2 \rangle}{6 \tau} \]

Eq. XII

Observed diffusion coefficients and hopping rates for simulations with different electrostatic cut-offs and coulomb scaling are summarized in Table 2. Decrease in D by a factor 0.6 was observed in simulations with coulomb scaling but hopping rate was higher compared to simulations without coulomb scaling.
3.3. Hydronium/Cl⁻ Simulations

Radial distribution function of Cl⁻-OW and Cl⁻-OW* showed a maximum around 3Å, and a second peak was observed between Cl⁻-OW* at 5Å (Figure 9). Rdf value of OW*-OW was around 4.5 which is slightly higher compared to simulations without Cl⁻, but the peak distance were the same 2.5Å (Figure 9).

3.4. Ammonium & Acetic Acid Simulations

Ammonium/Ammonia

In total 4 hops between NH₄⁺ and water happened during 1ns simulation. 0.8% percent of the time proton was bound to a water molecule. Estimated pH was 3.04 (pH = -log(0.008/(Na*Volume)) which gave pKa of 5.13 based on the the Henderson–Hasselbalch equation. In simulation with NH₃, proton travels in water molecules, until it gets closer to NH₃. After the hop from water to NH₃, proton remained in this new state for the rest of the simulation. 60% of the time proton was on water which gave an estimated pH of 1.19 and pKa of 1.33.
Acetic Acid
During 100ps simulation, number of acetic acid (COOH) molecules in simulation box for a certain time step is usually 3 or 4 (Figure 10). After 95ps, for a short time, all the acetic acid molecules in the simulation box were deprotonated. Estimated pH of the system was 0.76 and pKa was 1.05.

3.5 Dipeptide Simulations

Glutamate Dipeptide
Number of hops between each glutamate residue and water molecules was similar: 60 hops for GLU1 and 54 hops for GLU2. (De)Protonation was observed throughout the whole simulation. 31% of the time net charge on dipeptide was -2 (both residues were deprotonated) and 37% of the time it was -1 (only one residue was deprotonated). Experimental pKa of glutamate side chain is 4.4. Although deprotonated glutamate residues for a long time suggest a pH above the pKa value, estimated pH was 1.33. (De)Protonation energies (E_{12}) of the 2nd residue vary between from 10kJ to -80kJ, however much lower E_{12} values were observed for the 1st residue (Figure 11).

Glutamate-Lysine Dipeptide
Number of hops between glutamate and water molecules was quite low; 5. Number of hops between lysine and water molecules was 28. (De)Protonation of glutamate residue was only observed for the first 55ps frame and it was deprotonated for the rest of the simulation. (De)Protonation event of lysine was observed throughout the whole simulation. 31% of the time net charge on dipeptide was -1 (both residues deprotonated) and 32% of the time it was +1 (both residues protonated). Estimated pH of the system was 1.38.

![Figure 11. E12 over time for glutamate dipeptide: only E12 energies of successful hops between water-GLU (and vice versa) were shown.](image-url)
4. Discussion

One of the tasks to be clarified was to make sure that a random distribution would be generated by the method of use to treat donor/acceptor networks. It is important to have statistically correct distributions in order to generate randomness in simulations. So when two similar possible transfer reactions (i.e. both oxygens in carboxyl group might have similar probabilities to accept the proton) are present they will both have equal chances. The number of times each reaction is chosen should be proportional with reaction probabilities. Therefore one should prefer to use randlist mode to perform proton-transfer simulations. Although results with randomized list approach were better than sorted list approach, with randomized list approach lower probabilities had higher chance compared to Gillespie methods. It is possible that this is due to an error made during the construction of the script and not yet spotted. If this is the actual outcome, with randomized list approach, transfer event might happen more often between a donor-acceptor pair with low probability while there are other pairs that are more favorable. Indeed further investigation is needed.

Examples of proton-transfer simulations in small molecules and dipeptides were demonstrated. Although some systems were small or short simulation times were used, one should note that these kinds of tests are often necessary to compare the outcomes of the current status of the implementation, before moving into more complex systems.

Different setups with excess proton in bulk water resulted with difference in diffusion coefficients as well as different hopping rates. It shouldn’t be forgotten that hopping rates presented in results section, were not filtered. Meaning that hops lasts shorter than 10fs (back and forth hops) were included in the data. Thus one would expect to have lower actual hopping rates in many cases, especially when qhop frequency was set to 5fs which was the case for excess proton in water simulations. Coulomb scaling resulted with higher hopping rates and lower diffusion coefficients. Observations with PME were also higher compared to plain cut-offs indicating that proton-transfer is highly effected by the setup parameters.

Observed diffusion coefficients was higher compared to experimental value $9 \times 10^{-5} \text{cm}^2/\text{s}$, more importantly it was different from what was observed in reference study (Lill et al. 2001b): $9.5 \times 10^{-5} \text{cm}^2/\text{s}$. Considering the improvements that had been introduced, $9.5 \times 10^{-5} \text{cm}^2/\text{s}$ should not be considered as an accurate observation of D. I think that observation of D for SPC water model reflects the actual outcome of the method; $16 \times 10^{-5} \text{cm}^2/\text{s}$. Results in water systems indicate the need for improvements in algorithm. However in general all the results obtained in this study also points a need for a better water model to be used in proton-transfer simulations.

Radial distribution functions of both inert and hoppable hydronium oxygen, correlates with the observations of reference study (Lill et al. 2001b). Rdf peak of Cl– with water oxygens or hydronium oxygen were around 3Å. Studies suggest similar results to 3Å (Laasonen et al. 1997; Botti et al. 2004).
In simulations with ammonium, hopping rate between ammonium and water was quite low; suggesting a pH value lower than the pKa value of ammonium during simulation. To find the pH one needs to compute the average number of $\text{H}_3\text{O}^+$ molecules over time. Estimated pH of the ammonium system with average $\text{H}_3\text{O}^+$ concentration was 3.04 with a pKa value of 5.13 which is much lower than the expected pKa value of 9. Even lower pKa value observed in simulations with ammonia which points the importance of simulation time to reach steady state and but it should be noted that in simulations with ammonia since OH groups was not taken into account, results are not trustworthy. pKa value of acetic acid simulation was also lower than the expected value as well.

In both dipeptide simulations, most of the time glutamate residues were deprotonated. Similar to acetic acid and ammonium simulations, low pH values were observed. Even though an improved water model is needed to reach as close as possible to the experimental pKa values, longer simulations are necessary in order to get a good average. Also tests with an improved water model would give better results, pKa values closer to the experimental values.

So far all proton-transfer reactions were between water and different molecules. But it is also possible to study proton-transfer between different molecules than water. An example of this was tested between $\text{NH}_4^-$ - acetate ion (COO$^-$) and between $\text{NH}_3^-$ – COOH (data not shown). Soon, using GROMACS it will be possible to study proton-transfer reaction in complex systems with longer simulation runs. It might be interesting to analyze protein conformations with molecular dynamics studies at low pH in order to generate experimental conditions like in NMR or ESI methods.
Acknowledgements

Special thanks from lost spirit to…
David Van der Spoel and Erik Marklund for their guidance and patience throughout the whole project.
Staffan Svard for his guidance and helps throughout her master studies.
Her dear family for their support throughout her whole life.
Van der Spoel’s group; Daniel Larsson, Carl Caleman, Jochen Hub, Christian Wennberg and Hajdu’s group together with Åqvist’s group whom she happened to meet.
Her friends; new and old friends who were always there and always will be.
Her elementary school teacher, Suham Akıllı who had a great influence on her life.

And the most of the special thanks goes to her beloved nephew, Laşer Emirali Yoluk, who should be on his way to Sweden from Turkey on foot.
References

APPENDIX I

Python script that was used to analyze the distributions of Gillespie methods, Randomized List and List approaches:

```python
out=open('mrnd.txt','w')
out.write('attempt\trandom\tlist\tfIRST\tg\tdirect\n')
probs=open('mrnd_prob.txt','w')

reaction_list=[]

class Box:
    def __init__(self, molecules, quantity):
        self.molecules=molecules
        self.quantity=quantity

class Reaction(object):
    def __init__(self, reactants, products, probability, r_index):
        self.reactants=reactants
        self.products=products
        self.probability=probability
        self.r_index=r_index
        probs.write(str(r_index)+'\t'+str(probability)+'\n')

box=Box(['AH','A','BH','B','CH','C','DH','D','EH','E','FH','F','GH','G','HH','H','IJ','I','JH','J'],[1,0,0,1,0,1,0,1,0,1,0,1,0,1,0,1])
reactions_list.append(Reaction(['AH','B'], ['A','BH'], 0.687969, 'R0'))
#reaction_list.append(Reaction(['AH','C'], ['A','CH'], random.random(),'R1'))
#reaction_list.append(Reaction(['AH','D'], ['A','DH'], random.random(),'R2'))
#reaction_list.append(Reaction(['AH','E'], ['A','EH'], random.random(),'R3'))
#reaction_list.append(Reaction(['AH','F'], ['A','FH'], random.random(),'R4'))
#reaction_list.append(Reaction(['AH','G'], ['A','GH'], random.random(),'R5'))
#reaction_list.append(Reaction(['AH','H'], ['A','HH'], random.random(),'R6'))
reaction_list.append(Reaction(['AH','B'], ['A','BH'], 0.68796884691, 'R0'))
reaction_list.append(Reaction(['AH','C'], ['A','CH'], 0.92190998092, 'R1'))
reaction_list.append(Reaction(['AH','D'], ['A','DH'], 0.806865011717, 'R2'))
reaction_list.append(Reaction(['AH','E'], ['A','EH'], 0.375419769804, 'R3'))
reaction_list.append(Reaction(['AH','F'], ['A','FH'], 0.247488635553, 'R4'))
reaction_list.append(Reaction(['AH','G'], ['A','GH'], 0.821933185569, 'R5'))
reaction_list.append(Reaction(['AH','H'], ['A','HH'], 0.962516362937, 'R6'))

```

---

21
#reaction_list.append(Reaction(['CH','H'], ['C','HH'], random.random(), 'R6'))
#reaction_list.append(Reaction(['DH','J'], ['D','JH'], random.random(), 'R7'))
#reaction_list.append(Reaction(['DH','I'], ['D','IH'], random.random(), 'R8'))
#reaction_list.append(Reaction(['EH','J'], ['E','JH'], random.random(), 'R9'))
#reaction_list.append(Reaction(['FH','G'], ['F','GH'], random.random(), 'R10'))
#reaction_list.append(Reaction(['IH','H'], ['I','HH'], random.random(), 'R11'))

def population():
    rl_p=dict(zip(box.molecules, box.quantity))
    l_p=dict(zip(box.molecules, box.quantity))
    direct_p=dict(zip(box.molecules, box.quantity))
    first_p=dict(zip(box.molecules, box.quantity))
    return rl_p, l_p, direct_p, first_p

def probability_list(mode):
    p_list={}
    p_v=[]
    r_v=[]
    for i in range(len(reaction_list)):
        if mode[reaction_list[i].reactants[0]] and mode[reaction_list[i].reactants[1]] > 0:
            x=reaction_list[i].probability
            y=reaction_list[i].r_index
            p_list[x]=y
            p_v.append(x)
            r_v.append(y)
    sum_p=sum(p_v)
    return p_v, p_list, sum_p

def change_population(mode, mu):
    mode[reaction_list[mu].reactants[0]]=mode.get(reaction_list[mu].reactants[0])-1
    mode[reaction_list[mu].reactants[1]]=mode.get(reaction_list[mu].reactants[1])-1
    mode[reaction_list[mu].products[0]]=mode.get(reaction_list[mu].products[0])+1
    mode[reaction_list[mu].products[1]]=mode.get(reaction_list[mu].products[1])+1

def randomlist(rl_p):
    t_start=0
    t_end=30
    while t_start < t_end:
        p_v, p_list, sum_p= probability_list(rl_p)
        random.shuffle(p_v)
        for p_mu in p_v:
            r=random.random()
            if p_mu > r:
                r_mu=p_list.get(p_mu)
                mu=int(r_mu.split('R')[1])
                change_population(rl_p, mu)
                break
        t_start=t_start+10
    try:
        mu
except NameError:
    mu='NONE'
out.write(str(mu)+'

def helms_lill(l_p):
    t_start=0
    t_end=30
    while t_start < t_end:
        p_v, p_list, sum_p= probability_list(l_p)
        p_v.sort()
        p_v.reverse()
        for p_mu in p_v:
            r=random.random()
            if p_mu > r:
                r_mu=p_list.get(p_mu)
                mu=int(r_mu.split('R')[1])
                change_population(l_p,mu)
                break
        t_start=t_start+10
        try:
            mu
            except NameError:
                mu='NONE'
out.write(str(mu)+'

def direct_m(direct_p):
    t_start=0
    t_end=30
    while t_start < t_end:
        p_v, p_list, sum_p= probability_list(direct_p)
        if sum (p_v) == 0:
            break
        p_a_list={} 
        a_list=[]
        for i in p_v:
            a_v=math.log((1-i))/(-10)
            p_a_list[a_v]=i
            a_list.append(a_v)
        r1=random.random()
        r2=random.random()
        r2a0= r2 * sum(a_list)
        tao=math.log(1/r1)/sum(a_list)
        if tao <= 10:
            sub_sum_list=[]
            for i in a_list:
                sub_a_v=a_list[0:(a_list.index(i)+1)]
                if sum(sub_a_v) > r2a0:
                    r_mu=p_list.get(p_a_list.get(i))
mu=int(r_mu.split('R')[1])
change_population(gil_p,mu)
break
else:
    break
t_start=t_start+10
try:
    mu
except NameError:
    mu='NONE'
out.write(str(mu)+'

def first_g(first_p):
    t_start=0
    t_end=30
    while t_start < t_end:
        p_v, p_list, sum_p= probability_list(first_p)
        if sum(p_v) == 0:
            break
        p_a_list={} 
        a_list=[]
        for i in p_v:
            a_v=math.log((1-i))/(-10)
            p_a_list[a_v]=i
            a_list.append(a_v)
        tao_list={} 
        for i in a_list:
            r=random.random()
            tao_v=math.log(1/r)/i
            tao_list[tao_v]=i
        tao_mu=min(tao_list.keys())
        if tao_mu <= 10:
            r_mu=p_list.get(p_a_list.get(tao_list.get(tao_mu)))
            mu=int(r_mu.split('R')[1])
            change_population(first_p,mu)
            break
        else:
            break
t_start=t_start+10
try:
    mu
except NameError:
    mu='NONE'
out.write(str(mu)+'

def x_times(nr):
    for x in range(nr):
        out.write(str(x)+'

24
rl_p, l_p, gil_p, first_p = population()
randomlist(rl_p)
helms_lill(l_p)
first_g(first_p)
direct_m(direct_p)
x_times(10000)
out.close()
probs.close()
## APPENDIX II

<table>
<thead>
<tr>
<th>Set</th>
<th>Reaction</th>
<th>Donor</th>
<th>Acceptor</th>
<th>Probability</th>
<th>Nr. Of Observations (after 10000 run)</th>
<th>Direct Method</th>
<th>First-Reaction Method</th>
<th>List</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Random List</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>First Set</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First Set</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R0</td>
<td>AH</td>
<td>B</td>
<td>0.443</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>AH</td>
<td>C</td>
<td>0.199</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R2</td>
<td>AH</td>
<td>D</td>
<td>0.309</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R3</td>
<td>AH</td>
<td>E</td>
<td>0.979</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R4</td>
<td>AH</td>
<td>F</td>
<td>0.732</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R5</td>
<td>AH</td>
<td>G</td>
<td>0.771</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R6</td>
<td>AH</td>
<td>H</td>
<td>0.934</td>
</tr>
<tr>
<td>Second Set</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R0</td>
<td>AH</td>
<td>B</td>
<td>0.914</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R1</td>
<td>AH</td>
<td>C</td>
<td>0.347</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R2</td>
<td>AH</td>
<td>D</td>
<td>0.540</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R3</td>
<td>BH</td>
<td>E</td>
<td>0.043</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R4</td>
<td>BH</td>
<td>F</td>
<td>0.812</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R5</td>
<td>CH</td>
<td>G</td>
<td>0.424</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R6</td>
<td>CH</td>
<td>H</td>
<td>0.758</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R7</td>
<td>DH</td>
<td>I</td>
<td>0.278</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R8</td>
<td>DH</td>
<td>J</td>
<td>0.644</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R9</td>
<td>EH</td>
<td>J</td>
<td>0.300</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R10</td>
<td>FH</td>
<td>G</td>
<td>0.927</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>R11</td>
<td>IH</td>
<td>H</td>
<td>0.127</td>
</tr>
</tbody>
</table>