5-2021

Mapping three dimensional interactions between biomolecules and electric fields.

Joseph Patrick Brian P.E.
University of Louisville

MAPPING THREE DIMENSIONAL INTERACTIONS BETWEEN BIOMOLECULES AND ELECTRIC FIELDS

By
Joseph Patrick Brian, P.E.
B.S. Chemical Engineering, 1980
P.E. Chemical Engineering, 1985
P.E. Electrical Engineering, 1988

A Dissertation
Submitted to the Faculty of the
J.B. Speed School of Engineering
of the University of Louisville
in Partial Fulfillment of the Requirements
for the Degree of

Doctor of Philosophy
in Chemical Engineering

Department of Chemical Engineering
University of Louisville
Louisville, Kentucky

May 2021

MAPPING THREE DIMENSIONAL INTERACTIONS BETWEEN BIOMOLECULES AND ELECTRIC FIELDS

By
Joseph Patrick Brian, P.E.
B.S. Chemical Engineering, 1980
P.E. Chemical Engineering, 1985
P.E. Electrical Engineering, 1988

Dissertation approved on

April 28, 2020

by the following dissertation Committee:

_____

(Vance Jaeger, Ph.D.) Chair

_____

(Joel Fried, Ph.D.)

_____

(Gautam Gupta, Ph.D.)

_____

(Kevin Walsh, Ph.D.)

DEDICATION

First and foremost, I would like to publicly thank my

Lord and Savior Jesus Christ

for saving me,

and orchestrating the events of my life to make the earning of this degree possible.

I couldn't have done it without him.

I thank him because
I know there will be no Ph.D.'s in heaven,
only children of God,
and that he gave his life on the cross
that I could be one of them,

and for allowing me to kneel in his presence
on the evening of 30 November 2009.

ACKNOWLEDGMENTS

I would like to thank my advisor, mentor and friend, Dr. Vance Jaeger for teaching me GROMACS, Unix, Metadynamics, PLUMED, CHARMM-GUI, AMBER and all the tricks he learned at the Max Plank Institute for Biophysical Chemistry in Gottingen, Germany.

My best friend Joe Mahoney for being there through thick and thin. There is a friend who sticks closer than a brother. Prov 18:24b

My parents Joe and Marlyn for bringing me into this world and caring for me.

My undergraduate professors at the University of Kentucky, especially Dr. Grieves.

My former employer, BFGoodrich / Westlake Chemical Corporation, who gave me a wonderful career, and indulged my quest to never stop learning.

My boss for most of my career, Jim Best, for being a fantastic boss, always having my back, never limiting my opportunities, and took the time to write one of my letters of recommendation.

Dr. Timothy Dowling, for his time and encouragement during my first serious time in a college classroom in 34 years and writing one of my letters of recommendation.

Dr. Mahendra Sunkara, for believing I still had plenty of gas in the tank and making a way for me to enter the program and earn my way through on such short notice.

Dr. Joshua Spurgeon, my first advisor, for always being accommodative of my needs to simultaneously deal with certain personal issues at that time related to the worst personal situation of my life.

The support of the faculty and staff of the Department of Chemical Engineering and the Conn Center for Renewable Energy Research even through the rough patches, especially Dr. Fried.

My first compatriot post-docs, Dr. Bijandra Kumar and Dr. Sudesh Kumari.

My classmates and friends at school, especially Farnaz Minooei, Tim Dubbs, Marzieh Moradi and Mike Martin.

Harrison Simrall, Assistant Director of IT Research Computing for all his help when I needed software packages updated on the Cardinal Research Cluster.

My committee members for their time and support.

ABSTRACT

MAPPING THREE DIMENSIONAL INTERACTIONS BETWEEN BIOMOLECULES AND ELECTRIC FIELDS

Joseph Patrick Brian

April 28, 2020

Newly developed molecular simulation techniques and post processing capabilities are enabling a deeper understanding of complex biophysical processes. Findings presented herein promise to improve the practice of electroporation, increase the stability of proteins in solution, and uncover the mechanisms nitrogenase uses to catalyze the conversion of atmospheric nitrogen to ammonia fertilizers. All-atom simulations of lipid membrane pore development as caused by electroporation have been investigated using finite transmembrane electric flux produced solely by charge imbalance across the membrane without an applied external field. A remarkable correlation between electric flux and pore size has been discovered that the results indicate is independent of pore size distribution. New post processing techniques, developed to measure pore size by efficiently counting water molecules in the pore "core", enable further frame by frame data refinement to remove artifacts arising from extraneous motion along all axes. These methods were used to create the first three dimensional maps of non-uniform electric fields that develop as biomolecules and electric fields interact. Rigorous validation of large data sets using these processing steps unexpectedly bore a second benefit, the ability to render visualizations of biostructures at the sub-nanometer scale that are inaccessible to direct experimental observation. Changes in free energy surfaces of proteins attributable to biomolecular force

fields, water models, ion models, salt species, and salt concentrations have also been characterized using an enhanced sampling method called metadynamics. Newly developed visualization techniques and electric field map tools have been applied to nitrogenase enzymes which catalyze the reduction of atmospheric nitrogen to ammonia through a network of electronic interactions coupled to large structural changes in the enzyme complex. The results presented herein provide researchers unprecedented insight into the interactions of biomolecules with electric fields, yielding improved knowledge of the mechanisms underlying their function.

TABLE OF CONTENTS

LIST OF FIGURES

CONTINUED

FIGURE PAGE

CONTINUED

FIGURE PAGE

CONTINUED

FIGURE PAGE

INTRODUCTION

Overview of molecular dynamics and structural biology. Molecular dynamics (MD) and its application toward biological systems was born in 1968 when Michael Levitt programmed the coordinates of one of only two known protein structures into a computer. The structure of the enzyme lysozyme had only been determined three years earlier from X-ray crystallography, after the structure of myoglobin had first been determined in 1959. Pictures of both had been published in the journal *Scientific American*. Levitt was inspired by these early articles, and he became fascinated by molecular structures. Working in Israel as a condition to be accepted to the Ph.D. program at Kings College in London, he had recently finished building the first physical model of a protein, lysozyme. Because he had the listing of all the atom coordinates needed to build the physical model, upon its completion he immediately set about building a computer model. In 1968 Shneior Lifson and Arieh Warshel published the first consistent physical-mathematical model (force field) (FF) of the bond strain energies and Lennard-Jones parameters for several n-alkanes and cycloalkanes. Levitt and Lifson applied this energy modeling technique to larger molecules of more general structure by including the elements hydrogen, carbon, nitrogen, oxygen, and sulfur, thus applying the FF to their computerized models of lysozyme and myoglobin. Their 1969 paper on macromolecular energy minimization established the foundation of what is now practiced in the much broader realm of MD for both biological and non-biological systems.

At the time, these proteins' structures had only recently been revealed by X-ray crystallography. To understand how the proteins function, researchers first needed to know the biologically active structures. However, X-ray crystallography requires proteins to be in a crystalline state. Biologically active structures exist within a narrow range of environmental constraints. Hydration, pH, salinity, salt species, temperature and electric fields all affect protein structure, but when the protein is in the crystalline form these variables cannot be adequately probed. For example, X-ray crystallography is undertaken at cryogenic temperatures and often in the presence of exotic chemicals required to induce crystal formation. Alternative methods have been developed to determine the structure of proteins that do not readily crystallize.

Transmission electron cryomicroscopy (cryo-EM) is a relatively recent method in structural biology, requiring hyperfast cryofixation of samples in a bath of liquid ethane to solidify structures before they have time to rearrange (sub-millisecond), driven by rapid changes in temperature. The technique is generally applicable to structures that are relatively static, and only a few recent attempts have shown promise in distinguishing distinct structural populations (e.g. open and closed states of a protein). But many biostructures are very dynamic, changing shape to induce functions such as to modulate membrane channels, to propagate membrane potentials along axions, to dock and undock cofactors to enzymes, and so on. These functions take place in the nanosecond to second time frame. The structures that underlie these critical functions will likely never be experimentally probed in the native environment at the atomistic level.

These inaccessible-to-direct-observation frontiers are beginning to be conquered at the all-atom level of detail using advances in supercomputers, force-fields, and simulation

algorithms. The fields of computational chemistry (CC) and MD have benefitted greatly in recent years due to the fortuitous development of graphical processing units (GPUs). These GPUs have been developed to render images with lifelike quality and with improved light raytracing algorithms. The mathematics of calculating ray traces are identical to those needed for calculating particle trajectories of atoms in MD simulations. GPUs are vastly superior to familiar computer central processing units (CPUs) for MD, and the concomitant increases in calculation speeds, now make viable all-atom simulations that were, until recently, impractical for all but the smallest systems. When studying molecular biophysics using MD, biostructures consisting of hundreds to millions of atoms (i.e. lipid membranes, proteins, ribosomes) are simulated to uncover the physical basis of life functions. In fact, the effects of external stimuli can also be probed. For the work presented herein, electric fields are of particular interest. All-atom MD simulations unlock, with some measure of fidelity, previously unknowable physical details of how non-uniform electric fields affect biomolecular conformations and functions. Knowledge gained from MD simulation can be used to engineer improved control of cellular functions. This dissertation explores frontiers made possible by immense advances in computational speed, new simulation software capabilities, and many hours of hard work.

Motivation. The genesis of this work was a study of lipid bilayer membrane (BLM) systems. For about a year and a half, we studied the behavior of BLMs during pore formation in the presence of transmembrane electric fields (i.e. electroporation). Electroporation is used most notably for cell transfection. Section 1 of this dissertation contains a recently submitted manuscript detailing our findings. Our research and a survey of the published literature on the practice of electroporation suggested three things. First,

there was no good molecular-level explanation of observed electroporation behavior. Second, electroporation is practiced mostly as a black art, somewhat blindly attempting to increase transfection efficacy without increasing the rate of cell lysis upon being subjected to the treatment. Third, modern simulation programs had no ability to investigate the role non-uniform electric fields play in the formation of pores through BLMs. I hypothesized non-uniform electric fields played a more significant and interesting role than anyone seemed to realize.

Two significant hurdles immediately presented themselves that would make analysis difficult. First, there was a need to develop an automated technique to remove the extraneous movement from simulation trajectories (i.e. a set snapshots from the dynamic simulation) caused by simulation features like center of mass motion removal and fluctuations in simulation box dimensions created by the semi-isotropic barostat. Second, since the calculation of the electric field map scales as N^6, where N is the number of particles in the simulation box (N ~ 100,000 for a BLM system), a brute force method of creating the map at high resolution was out of the question. Solutions for both problems were eventually developed, paying some unexpected dividends, as previewed in **Fig. 1,** and presented in Section 2.

Even with recent advances in simulation speeds, many problems still require simulation times vastly larger than can be achieved by modern computers. One class of these problems, namely calculating free energy landscapes of protein folding, is of particular interest to structural biologists who are interested in understanding how proteins work and how protein function can be improved through rational design. Free energy is a quantity of particular interest, because it provides insight into a protein's prefered folded

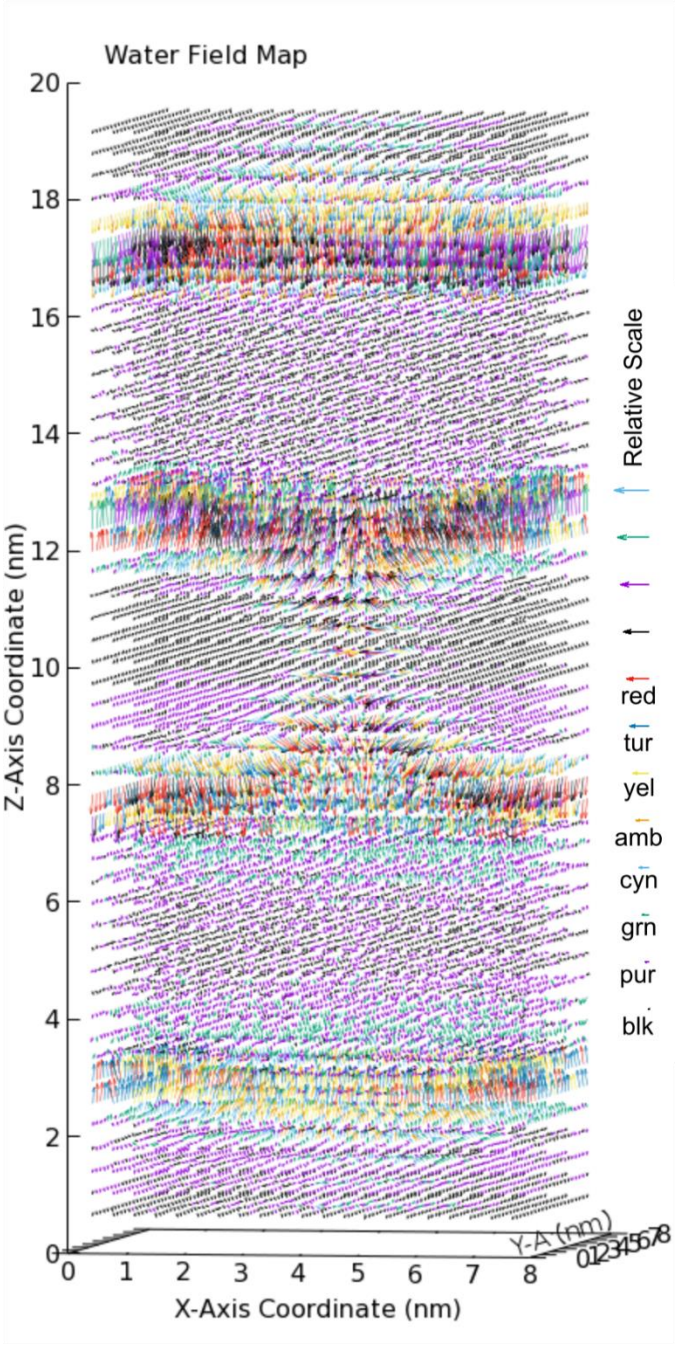


Fig. 1: *One of the first electric field vector maps produced, this specifically for the field attributable soley to common alignment of the the water molecule dipoles over biostructure dimensions.*

and unfolded conformations. However, even small "fast" folding proteins fold and unfold on the timescale of microseconds. An accurate estimate of folding thermodynamics from standard MD simulations is immensely costly and has only previously been achieved by using distributed computing for explitly solvated all-atom systems. So-called "enhanced sampling techniques" have been developed to accelerate protein folding and unfolding in MD simulations while providing an accurate estimation of folding thermodynamics. One such method is known as metadynamics, and it will be applied herein to explore the folding of a model protein. Because the early work of this dissertation focused on the effects of ions and electric fields on BLMs, a natural extention of this was to study the response of protein structures to changing environmental conditions that effect

electrical potentials, like salt species and concentration variations. Simulations have been performed to determine the effects of salt species, salinity, water, and protein FFs on protein folding thermodynamics in molecular simulations. Best practices for the selection of accurate, compatible salt, water, and protein models have thus been developed. Section 3 of this dissertation contains a soon-to-be submitted manuscript relating to this work.

Chemical engineers seek to create catalysts to increase reaction rates and make commercially valuable compounds. Maximization of product selectivity and minimization of energy expenditure are major economic drivers of this work. In biology, these catalysts are proteins known as enzymes, and nature often uses yet unknown mechanisms to guide highly selective and efficient reactions. Nitrogenase is an enzyme produced by certain bacteria, algae, and archaea that reduces atmospheric dinitrogen to ammonia, a critical component for synthesis of all amino acids and nucleotides. The synthetic Haber-Bosch (HB) process performs the same reaction but requires pressures of 2,200 to 3,600 psi and temperatures between 752 and 932°F, consuming 1-2% of the world's entire energy production and 3-5% of its natural gas production. Without the fertilizers produced by the HB process, insufficient food could be grown to keep the world's population fed. Researchers have been exploring the secrets of nitrogenase activity for the last 50 years in hopes that they can discover an improved bio-inspired synthetic catalyst for nitrogen reduction. Again, this inaccessible-to-direct-observation frontier is one where MD and CC have much to offer to help unravel these secrets.

Briefly, nitrogenase works in tandem with a co-enzyme, dinitrogenase reductase, that binds adenosine-triphosphate (ATP) molecules, before docking with nitrogenase. ATP is cleaved to form adenosine-diphosphate (ADP) and a phosphate ion, pumping electrons

and energy toward the substrate. The co-enzyme undocks and exchanges ADP for ATP in a repeating cycle. Collectively these enzymes use three types of metal-containing cofactors to transport and retain free electrons. The docking/undocking cycle operates through physical changes in docking region's shape and charge. Nitrogenase must accumulate eight electrons to reduce one molecule of dinitrogen. These electrons change the local electric field and hence also influence the enzyme's conformation and dynamics. To study the effects of nucleotide (e.g. ADP and ATP) binding and charge states of co-factors on the proteins' conformational dynamics, we have constructed a model of the nitrogenase enzyme complex and its constituent parts. Building a model of a large organometallic system is no small challenge, since commonly used FFs do not contain parameters for metals and metal complexes. CC methods were used to calculate the charge distribution in the metal cofactors in their base (uncharged) state. In the enzyme, most of these cofactors are ligated to the protein. To achieve additional electronic isolation of the FeMo cofactor, a chelated molecule, homocitrate, is also entirely held in place by ligand bonds. The smallest version of this system consists of over 300,000 atoms. The new nitrogenase model and initial findings about the effects of nucleotide binding on protein dynamics are contained in Section 4 of this dissertation, which will form the basis of a manuscript for later publication.

There are many more detailed results from these four sections than can reasonably be individually discussed. Hopefully, this introduction has provided an effective outline and perspective for appreciating the skills and techniques I have mastered as well as those I have personally developed to produce this body of work.

SECTION 1

INSIGHTS INTO THE MOLECULAR MECHANISMS OF ELECTROPORATION FROM COMPUTATIONAL ELECTROPHYSIOLGY

STATEMENT OF SIGNIFICANCE

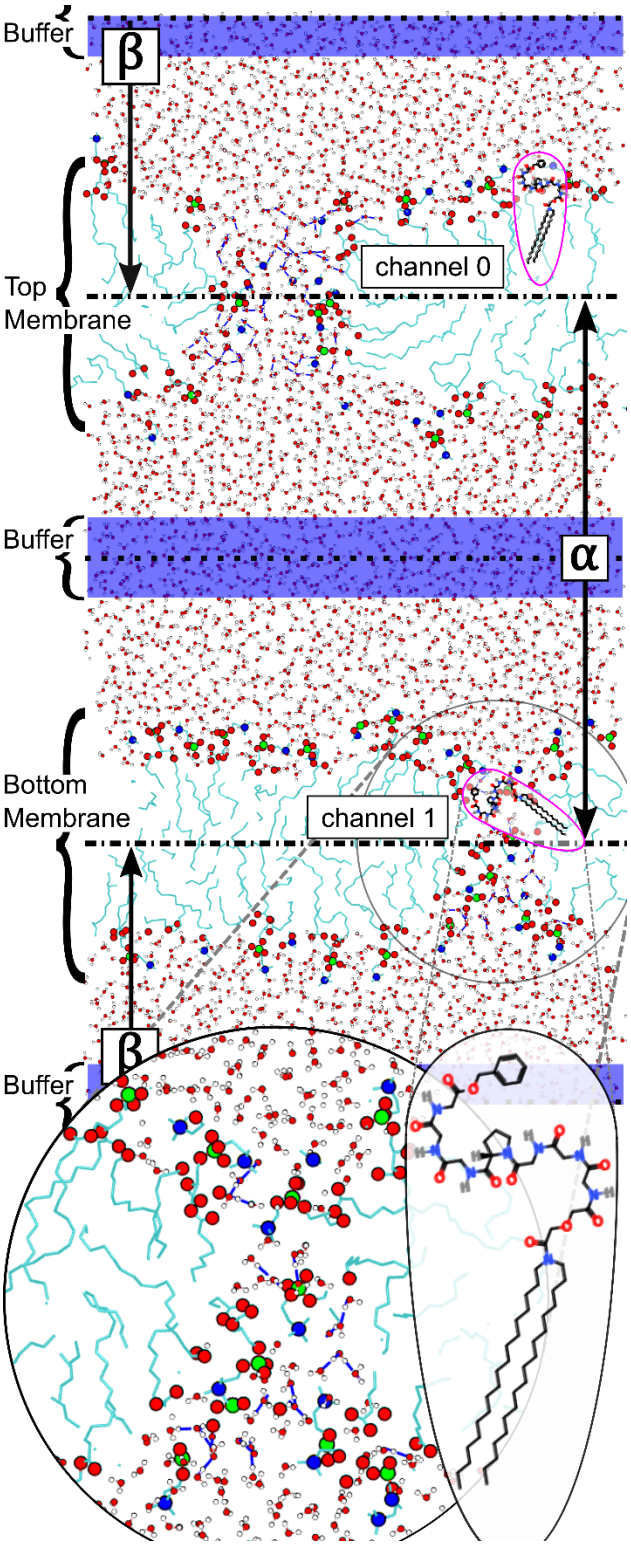
Though electroporation is widely practiced in research and clinical environments, the fundamental theoretical molecular level mechanisms of pore formation, stability and longevity are poorly understood. Advances in multiple facets of molecular dynamics have enabled simulations of previously unobtainable fidelity. All atom simulations incorporating a recently developed method known as computational electrophysiology demonstrate convincingly that pore size is established mostly by the magnitude of the

applied electric flux. This insight suggests certain logical modifications to electroporation protocols and instrument design to increase treatment efficacy while simultaneously decreasing cell mortality.

INTRODUCTION

Electroporation is often used to induce pore formation(1-3) in lipid membranes to facilitate the transfection of cells with DNA and to deliver macromolecules such as proteins(4, 5). Although the efficacy of electroporation has been repeatedly demonstrated(6-11), molecular level structures during and after pore development, pore characteristics, transport properties, stability, lifetime, decay and disintegration are not well understood(12). Since electroporation involves the interaction of many thousands of molecules dynamically rearranging at once in an electric field, it is difficult to study the molecular biophysics experimentally at high resolution(13). MD simulations, on the other hand, are well suited to provide molecular-level structures of dynamically rearranging biomolecules.

It is well known that molecular simulations of lipid membranes are sensitive to several factors(14-18). Consequently, modeling complex behavior in lipid bilayer systems requires techniques that accurately reflect real life conditions. Previously published MD simulations of electroporation have induced pore formation by the addition of an electric field term in the Hamiltonian(19, 20). While this method is easily implemented in many simulation packages, it is an imperfect solution to the problem for two main reasons. First, the applied electric field causes ions to irreversibly permeate through the membrane, which changes the effective transmembrane potential during the simulation. Second, to simulate



a concentration gradient, a vacuum layer must be introduced to separate the ion baths. This vacuum layer causes changes in the structuring of water near the interface, and the simulation of a vacuum layer requires the Canonical Ensemble (NVT) rather than the preferred Isobaric-Isothermal Ensemble (NPT).

To overcome these disadvantages, we have used a recently developed method called computational electrophysiology(21) that modifies the behavior of an MD simulation. CompEL uses a resourceful simulation geometry in which two membranes are present, denoted as the "double membrane" geometry as illustrated in **Fig. 2**. This double-membrane construction creates two independent ion baths (regions α and β)

Fig. 2: *Double membrane geometry. The α region is a volume in the center of the simulation box between the membranes. Phosphorus is green, nitrogen is blue, and oxygen is red. Faint blue lines in the pore are hydrogen bonds.* ***Oval inset:*** *SCMTR molecule. Magenta ovals show typical positions in the membrane and in pores.*

with no vacuum layer between. Thus, the simulation boundaries are free of discontinuities, and no vacuum layer

must be introduced. Therefore, a barostat can be used to control the lateral pressures on the membrane with a semi-isotropic algorithm unlike many previous simulations(22-24). Because of these advantages, CompEL has previously been used to study ion permeation through protein ion channels(25-27). We hypothesize that the NPT ensemble coupled with the CompEL algorithm will model more realistic behavior during simulations of electroporation, because it allows for the expansion or contraction of the membrane surface area, which is important in the evolution, maintenance, and fidelity of pore size and formation/disintegration kinetics.

The ability to modulate membrane behavior in the presence of electric fields and ion gradients could be useful in enhancing electroporation efficacy. To study how membrane anisotropy affects electroporation, a synthetic ion transporter known as SCMTR has been included. SCMTRs are a class of molecules designed to mimic the structure and function of natural membrane chloride ion channels. Several SCMTR structures have been synthesized and tested with different efficacy in encouraging anion transport(28). Typically, their structures consist of a heptapeptide capped on one end with two lipid-like tails and on the other end with a moiety selected to impart hydrophobicity. The class of SCMTRs includes variations on this theme with varying tail lengths, tail saturation, tail cis-trans geometry, amino acid sequences and headgroup types. When added to liposomes with a high internal chloride ion concentration, SCMTRs rapidly and selectively induce the release of chloride ions from the liposome into the surrounding bath (10:1 Cl-/K+ ion permeations(29)). Recent studies on SCMTRs in planar lipid membranes indicate that

channel formation at low transmembrane voltages is a relatively rare event, sometimes requiring several minutes for channels to form(30). SCMTRs have only been studied computationally once before(31) and kinetic mechanisms for their observed behavior have yet to be postulated. Prior experimental experience with SCMTR molecules(30) led us to hypothesize that their ion transport enhancing characteristics might extend to electroporation as well. The oval inset of **Fig. 2** and **Fig. S1** in the Supporting Material (pg. 116) show the SCMTR type used in this work. The chemical formula for this SCMTR is ((C18H37)2NCOCH2OCH2CO-GGGPGGG-OCH2Ph).

Reversible electro-permeabilization of lipid membranes has revealed a 10:1 specificity for cations(32), but has not described the selectivity mechanism. Several classes of molecules such as SCMTRs(29), hydraphiles(33) and lariats(34) promote membrane pore formation based upon ionic concentration and charge gradients.

The CompEL double membrane geometry, while eliminating vacuum boundary discontinuities, doubles the number of atoms in a simulation, causing the simulation to run at about half the speed, since the simulations presented here were run near the linear scaling regime. However, in the optimal geometry with two pores there are twice as many locations for membrane-crossing events to occur, meaning CompEL does not diminish overall computational efficiency. CompEL tracks the concentration of ions in each of the baths separated by the lipid membranes. When an ion crosses a membrane, CompEL detects a change in the concentration gradient, and reestablishes the set-point concentrations in each bath by removing an ion of the same species from the buffer region and placing it back in the buffer region of the chamber with the depleted ion concentration. Thus, the concentration gradient over the membrane is controlled, and a quasi-steady state is

maintained. With CompEL's ability to address shortcomings of previous simulation methods, we aim to study the formation, structure, and dynamics of pores formed during electroporation and the transport of ions through these pores.

CompEL is therefore is an invaluable method that we believe will enable the first development of a fully three-dimensional theoretical understanding of the role the non-uniform electric field plays in stabilizing the equilibrium membrane structure that contains pores. Establishing the initial (uniform) electric field (and finite flux) with a finite transmembrane charge imbalance ensures that the resulting non-uniform electric field must consist of that same initial finite electric flux. For a brief review of the relevant electrostatics discussed in this paper, see the **electrostatics** section in the Supporting Material (pg. 114) where it is explained that electric flux is proportional to transmembrane charge imbalance. With a better understanding of the role electric flux has in determining pore size and stabilizing membrane structure, new electroporation methods could potentially exploit this knowledge to grow pores to an equilibrium size for the applied flux.

Finally, electroporation as practiced, involves timescales beyond current computational capabilities, especially in cases of macromolecule transport. However, by enhancing pore formation with additional electrical potential, MD simulations can elucidate transitions between membrane and pore states and the equilibrium characteristics of these states(35, 36). With a first principles theoretical approach, a clearer understanding of the molecular biophysics underlying pore formation, membrane permeation, pore dynamics, and the specific interactions that lead to experimental observables can be gained.

Molecular mechanisms of pore formation are conceptualized here as the competition between forces that stabilize and destabilize membrane structures. The lateral compressive and tensile forces between atoms in lipid bilayer membranes vary more than two orders of magnitude through the thickness of the membrane(37-42). The compressive and tensile forces in equilibrium with one another, establish the planar geometry of the membrane(43). Membrane structures experience a time-averaged uniform electric field (normal to the membrane surface) due to equal distributions of oppositely charged ions on either side of the membrane. In the presence of sufficient transmembrane electrical potential, (~2.0 V) we hypothesize the following driving forces for electroporation:

1) The external electric field torques the lipid headgroup dipoles in the anti-aligned leaflet away from anti-alignment, thus strengthening the effective magnitude of the external electric field locally.

2) This local disruption of the normal headgroup dipole alignments reduces the energy barrier for the intrusion and growth of continuous water columns into the membrane hydrophobic region. In these columns, the water molecule dipoles are aligned with the external field, thus conducting, intensifying and extending it non-uniformly in the direction of the opposite leaflet(44-46).

3) The intensified electric field induces further self-reinforcing interactions that evolve to develop a continuous hydrophilic region between the outer surfaces of the two leaflets, commonly referred to as a pre-pore (i.e. water penetrates the membrane).

4) The primary driving force is the concentration of the electric flux in the center of the pore by the progressive alignment of water molecule dipoles.

5) The intensification of electric flux in the center of the pore is balanced by a commensurate decrease in the electric flux in the hydrophobic region beyond the annulus of the pore.

By this process, the membrane structure rearranges to form a new, stable pore in equilibrium with the non-uniform electric field. Around the pore's center, the lipid leaflets are drawn into a toroidal geometry as a portion of the relaxed electrical stress transforms into increased leaflet mechanical stress. The pore geometry that minimizes the Gibbs free energy of the conformation while balancing the electric and mechanical stresses is toroidal(12, 47, 48), and as the diameter grows, an open pore forms in the membrane. Ultimately, the equilibrium size of the pore is established by the electric flux through the membrane.

METHODS

Build CHARMM lipids membrane and SCMTR

To begin our simulation campaign, a simulation box with a single membrane geometry was created using CHARMM(49) lipids. The CHARMM-GUI web server(50) was used to create a single lipid bilayer membrane of 174 DOPC (Dioleoyl phosphatidylcholine) molecules and 11,506 water molecules with the CHARMM36(51) FF for lipids and mTIP3P(52) for water. Steepest descent minimization was performed for up to 5000 steps, and the single membrane was equilibrated in the NPT ensemble with semi-isotropic pressure coupling for 10 ns in GROMACS2018(53-62). CHARMM-GUI was also used to create a topology file for the SCMTR molecule using the CHARMM general FF(63, 64) (CGenFF) (see **Fig. S1**).

MD parameters

GROMACS2018 was used for all MD simulations. The standard GROMACS simulation parameters for CHARMM36 lipids(65) were used as indicated by the CHARMM-GUI webserver. Hydrogen-heavy atom bond lengths were constrained with the LINCS algorithm. A Verlet integrator was used, and Lennard-Jones interactions were cutoff with a force-switch modifier between 1.0 and 1.2 nm. Electrostatics were calculated using the particle mesh Ewald (PME) method with a real space cutoff of 1.2 nm. No dispersion corrections were used. Production runs were performed using a 2-fs integration timestep. Temperature was controlled at 303.15 K with the Nose-Hoover thermostat(66) and pressure was controlled at 1 bar with the Parrinello-Rahman barostat(67) unless otherwise noted. A representative MD parameter file for GROMACS has been included in the Supplemental Information.

Place SCMTR in membrane

Several modifications to the geometry produced by CHARMM-GUI were needed. First, the GROMACS utility *genion* was used to replace 22 water molecules with 11 Na ions and 11 Cl ions using a standard CHARMM ion force-field(68). The salt molarity of the resulting bath solutions was about 50 mM (ionic molarity of about 100 mM) since cations and anions were always swapped 1:1 between baths to produce the span of charge gradients. These values were selected to match a previous study by Burkhardt et al.(31) The *-membed* option of GROMACS *mdrun* program was used to insert a dimer of SCMTR molecules into the membrane in the active conformation suggested by previous experimental results(69, 70). This step resulted in the removal of 14 lipids from the bilayer to make room for the SCMTR dimer, 6 from the upper leaflet and 8 from the lower leaflet.

This system was equilibrated for 20 ns with semi-isotropic pressure coupling using the Berendsen barostat(71) to converge the density of the simulation box. After this step, vacuum layers of 3.0 nm were added in the *z*-dimension and ions were rearranged to create a +/- 6 e^- charge imbalance across the membrane (an excess of 6 Na ions on one side of the membrane and an excess of 6 Cl ions on the other). Then steepest descent energy minimization was performed for up to 50,000 steps in GROMACS.

Build double membrane system

To produce the double membrane geometry, the energy minimized system described above was copied, rotated 180 degrees about the x axis and translated along the z-axis while removing any vacuum layer to create the continuous double membrane system. Ions were swapped between the α and β regions (see **Fig. 2**) to set an initial charge imbalance of +/- 6 e^-. Up to 50,000 steps of energy minimization and 100 ps of NPT equilibration were used to collapse any remaining gap between the two halves of the box. Finally, 40 ns of NVT equilibration was conducted to ensure a good starting configuration for the CompEL algorithm.

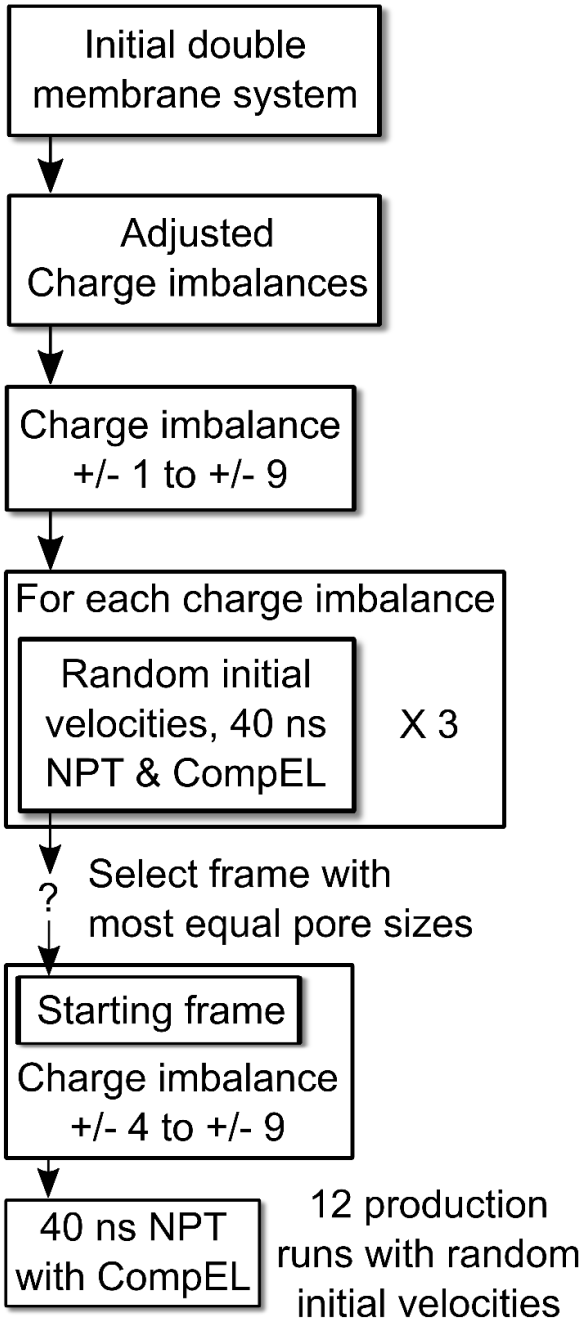
Build system with pore in each membrane

Five copies of the double membrane configuration were assigned independent initial velocity distributions. In this geometry, pore formation can be induced by introducing a relatively high transmembrane voltage (~2.0 V) caused by an ion charge gradient of +/- 6 e^-. This voltage was selected, because the timescale of pore formation at lower voltages is too long to consistently observe pore formation. For this work we coined and use the term headgroup channel to define any continuous column of lipid hydrophilic moieties forming between the membrane leaflets, which we then differentiate as pores or

pre-pores as described later. Simulations were run for up to 40 ns using the NVT ensemble. Once a stabilized pore formed in the simulation, a frame was selected that had no ions in the center of the pore. The half of the double membrane system without a pore after this step was then removed. The remaining half of the system with the pore in the membrane was copied, rotated 180 degrees about the x axis and translated along the z-axis to create the continuous double membrane system with a pore in each membrane. Ions were swapped between the α and β regions (see **Fig. 2**) to keep the charge imbalance at +/- 6 e^-. Energy minimization and 100 ps of NPT equilibration were applied to collapse any remaining gap between the two halves of the box.

Production simulations workflow

This double membrane open-pore system became the initial frame used to create simulations spanning a larger range of transmembrane ion charge gradients of +/- 1 to +/- 9 e^-. After appropriately swapping ions to create the nine initial configurations, three instances of each were simulated in the NPT ensemble for 40 ns to allow the pores to reach their equilibrium size. For each charge imbalance, a frame from near the end of one of these simulations was selected in which the pores were nearly equal in size and no ions were present in the center of the pores. Visual analysis of simulations of this system with smaller charge imbalances, (+/- 1 to +/- 3 e^-), prompted us to classify them as pre-pore. For the systems with headgroup channels that we considered to be at the pore stage of development (+/- 4 to +/- 9 e^-), 12, independent 40 ns simulations were performed in the NPT ensemble using random initial velocities. These 72 simulations were duplicated



identically in all respects except that the SCMTR molecules, while still present, were not associated with the pore. This second set of simulations allowed us to identify changes to pore behavior induced solely by the presence of SCMTR molecules. **Fig. 3** illustrates this workflow once the starting system had been assembled.

Count water molecules

A method was developed to accurately count the water molecules in the core of a pore on a frame by frame (one every 20 ps) basis from the trajectories. Because of fluctuations in the coordinates of the center of a pore caused by the semi-isotropic pressure control and center of mass motion removal, some challenges

Fig. 3: *Simulation workflow to generate CompEL production runs from assembled double membrane system.*

arose which were overcome using the GROMACS *select* routine. The *select* routine statement that enabled the extraction of this data along with a full explanation of its operation is presented in the Supporting Material. Our definition of the pore's core is

bounded vertically (+/- 0.75 nm) from the center of geometry of the headgroup nitrogen, phosphorous and oxygen atoms within a horizontal radius of the pore center. To determine a proper value for this radius, the average radius of the core water molecules, which also define the radial extent of lipid headgroups in the core, is increased by 50%. Without this limit, water molecules associated with membrane curvature away from the pore would be counted as well, increasing water molecule counts an average of ~6.5%. For systems with no developed pore, the +/- 0.75 nm setting created a so-called "live zero" pore water count in the typical range of 1-3 molecules. Live zero measurements are implemented by setting selection criteria parameters to maximize the explicit discernment between noise and the anticipated signal when it is at a known value of zero. Using a live zero setting permits any development in the membrane interior, like water wires or initial movement of headgroups toward the interior to immediately be detected and located for accurate quantification to produce the graph of pore core water molecules.

SCMTR-pore and pore-flux interactions

To quantify SCMTR's effects on pore formation kinetics, 240 simulations of 40 ns on double membrane systems starting with no pores at +/- 6 e^- charge imbalance were conducted. To quantify SCMTR effects on pore stability, starting frames (equal pore sizes in each membrane) were selected from the production runs at +/- 4 e^- charge imbalance both with and without SCMTR's associated with the pore. Ions were swapped to eradicate the charge imbalance. Then ten production runs of 40 ns, each with random initial velocities, were conducted for both starting frames. The trajectories were analyzed in VMD(72) (Visual Molecular Dynamics) to determine pore closure times. To quantify the effects of charge imbalance on simulation box *x*-*y* dimensions (area per lipid), the average

area of each of the 144 production runs was calculated and then the standard deviation was calculated for each of the 12 types of production runs (six different charge imbalances with and without SCMTR's).

RESULTS AND DISCUSSION

Transmembrane charge imbalance

Three initial 40 ns simulations were conducted for each of the nine charge imbalances. Many simulation animations across the span of charge imbalances were visualized in VMD. Visual inspection prompted us to classify the channels in simulations with charge imbalances of +/- 3 e^- or less as pre-pores. The pre-pore structure comes into being once the water and headgroup intrusions from each leaflet meet in the hydrophobic region, forming a headgroup channel. We distinguish the pre-pore and pore regimes as well as pore size by counting the number of water molecules in the central (1.5 nm) transmembrane region. For now, we define pores as headgroup channels where half or more of the water molecule hydrogen bonds are with other water molecules, thus establishing an objective criterion for classifying water channels as pores or pre-pores in the computation realm. Subsequently, a set of 144 simulations of 40 ns each was conducted with transmembrane charge imbalances spanning from +/- 4 to +/- 9 e^-. Pore size, pore size distribution, ion flux and ion charge selectivity were quantified as a function of transmembrane ion charge imbalance. Then the variance attributable to the presence of the SCMTR molecules in the pores was determined for each of these observables. **Fig. 4** illustrates the pore size distribution over the span of simulated transmembrane charge imbalances.

The average number of water molecules present in the pores is a linear function of the charge imbalance. No matter how water molecules are distributed between the two membrane pores, the total number of water molecules over the two pores (i.e. the sum of the number of waters in both membranes) remains the same. However, **Fig. 4** obfuscates some detail from our simulations by averaging over the full 40 ns trajectory. In some simulations, one of the two pores close, yet the closure is always balanced by equivalent growth in the remaining pore. **Fig. S2** in the Supporting Material (pg. 117) illustrates a case where a pore closes, yet somewhat counterintuitively, the average water molecules for each pore in that simulation are nearly identical.

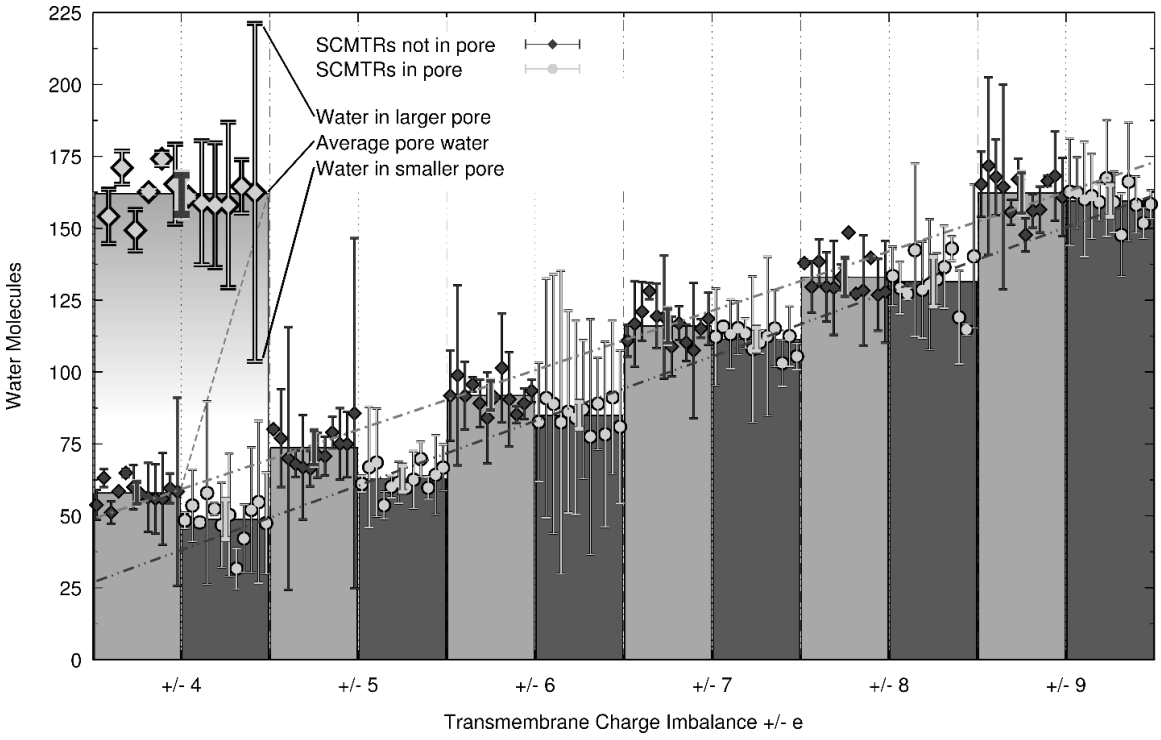
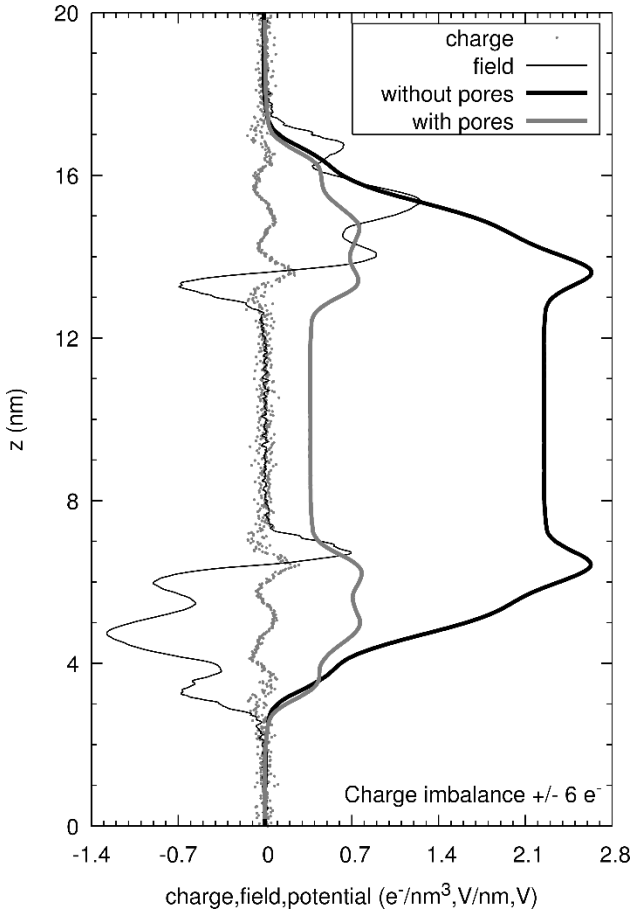


Fig. 4: *Distribution of core water molecules in induced pores. The magnified inset explains how the data are presented. Bars represent the average number of water molecules in both pores for a given charge imbalance. Light bars indicate a SCMTR-free pore, and dark bars indicate SCMTR is associated with the pore. The central whisker is the standard deviation of this average over 12 observations. The dashed lines show the linear regression for each pore type with regression coefficients of* R^2*=0.9930 (SCMTRs not in pore) and* R^2*=0.9932 (SCMTRs in pore).*

Another curiosity in **Fig. 4** is the clear and consistent difference in the pore size over the span of charge imbalances attributable to the presence of SCMTR molecules in the pore. During visual inspections of animations, we developed a hypothesis to explain this difference. While the simulations started with the SCMTR molecules dimerized, the structure of the dimer evolves when it is coupled with a pore. Often the heptapeptide chains are wrapped like two belts around sections of the circumference of the pore, hydrogen bonding to headgroups and occupying headgroup sites that would normally be hydrogen bonded to water molecules. This substitution explains the very consistent difference in pore water molecule counts. In later discussion, other results indicate that this heptapeptide conformation with the pore headgroups also enhances pore stability and longevity.

The initial double membrane geometry (before pore formation) exposes each membrane to a roughly uniform electric flux because ions freely diffuse through the solvent bath. The average transmembrane potential resulting from this uniform flux was calculated using the GROMACS *potential* utility as shown in **Fig. 5** for a system with a transmembrane charge imbalance of +/- 6 e^-. Pore formation kinetics are enhanced by using a high transmembrane voltage of about 2.0 V. With this voltage pores tend to form in under 40 ns. With smaller voltages, pores take significantly longer to form. Since no external electric field is applied in our simulations, pore formation and growth can only develop to a point of equilibrium with the finite electric flux created by the ion charge imbalance between the baths. For a more detailed view of simulation, pre- and post-pore electrical properties see **Figs. S3** and **S4** in the Supporting Material (pgs. 118-119). That the total pore size was so highly correlated with the total electric flux irrespective of the pore size distribution was not anticipated. Visual inspection of simulation trajectories also



revealed a pronounced alignment of the choline moieties (lipid headgroups) in the pore in the direction of the electric field (see **Fig. S5** in the Supporting Material (pg. 120)) even at transmembrane potentials ostensibly as small as 0.4 volts. The explanation for this misleading and possibly unappreciated disparity lies in how membrane potentials are currently calculated in GROMACS.

Fig. 5: *Membrane charge distribution, electric field and potential (before pore development) and potential (after development). The flat character of the potential curve in the water bath regions reveals there is no external electric field present except for that produced by the ion charge imbalance that exists between the two baths.*

Analytical solutions for determining the electric field of uniform charge distributions only exist for conformations that are highly symmetrical. Averaging the analysis of electric properties over the x-y plane (as implemented in the GROMACS *potential* utility, being predicated on the assumption of planar symmetry) cannot elucidate important details about the three-dimensional evolution of pore structures as their coulombic interactions are transitioning from a uniform to a non-uniform electric field. We are not aware of a simulation analysis tool that permits a full three-dimensional determination of the non-uniform electric field and its interaction with the membrane to

form pores by analysis of the simulation trajectory data. We address the need for such a utility in our conclusions.

Ion fluxes

During each simulation, ion permeation events were recorded for each pore. These data were matched with the actual pore size in each membrane and plotted in three dimensions to visualize interactions and dependencies/correlations. The results for pores without SCMTRs are displayed below in **Fig. 6**, illustrating relationships between transmembrane charge imbalance, pore size, and ion permeation events. The corresponding figure for pores with SCMTRs is shown in **Fig. S6** in the Supporting Material (pg. 121).

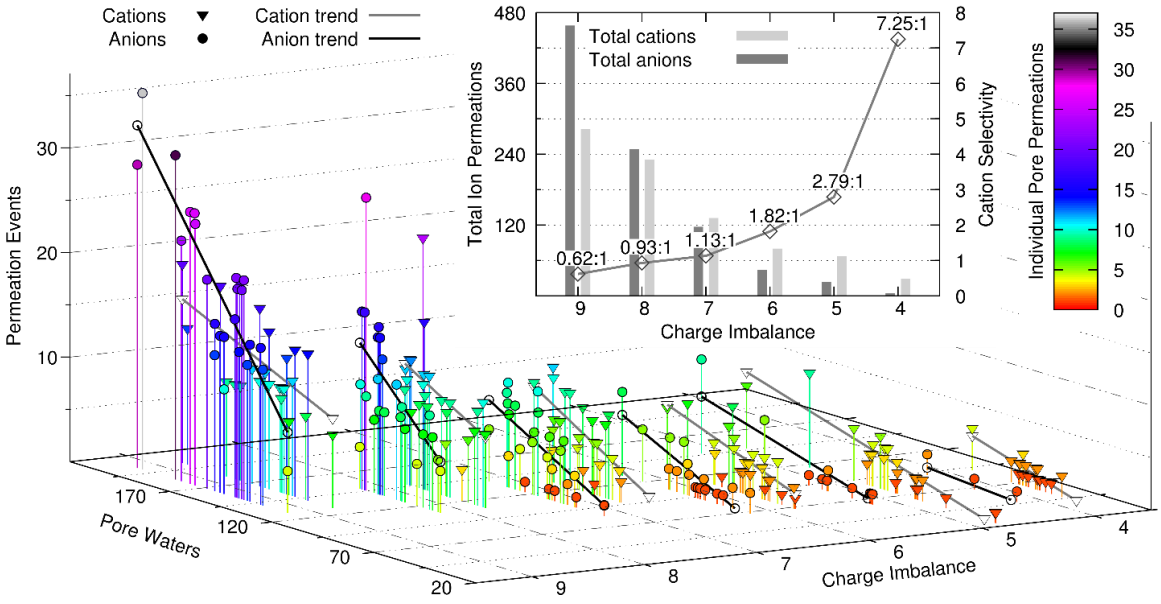


Fig. 6: *Ion permeation rates vs. pore size and charge imbalance for plain pores (no SCMTR present). Data for each charge imbalance (integer values) are split left and right to separate permeation of cations and anions. Note relatively constant trend of cation diffusion vs. pore size compared to steadily increasing trend of anion diffusion. Inset plot displays the total ion permeations by charge for each charge imbalance and associated variance in cation selectivity.*

Along the y-axis (charge imbalance), the integer values are split left and right to separate the permeations by ionic charge, right for cations and left for anions. At charge imbalances of +/- 4 e^- and +/- 5 e^-, some small pores with cation permeations had no corresponding anion permeations. This is the reason for the apparent lack of anion points for some pore sizes where a plotted cation data point is visible. By summing all the permeation events by ion type for each of the charge imbalances, the total ion flux versus charge imbalance can be compared, which the regression coefficients of **Fig. 4** demonstrated, strongly correlates with total pore size. These total permeations are plotted as bars on the inset plot. The ratio of the total of the cation to anion permeations is the pore cation selectivity, plotted with the diamond points.

The difference in the total permeations of opposite charged ions is the net flow of electrical current through the pore. By averaging the total permeations of each ion type for each charge imbalance these currents can be calculated and another nuance of the pore cation selectivity becomes more apparent. Over the span of these charge imbalances, the direction of the net current flow reverses. Since each simulation was 40 ns in length, each net pore permeation accounts for a current of about four picoamperes. The distribution of the charge imbalance in the simulations was such that the electric field always pointed from the α-region (positive charge) to the β-region (negative charge) (see **Fig. 2**). Positive electric current flow is defined as the flow of positive charges in the direction of the electric field. Knowing the direction of the electric field, the net currents were calculated at charge imbalances from +/- 4 e^- to +/- 9 e^- and averaged +4.2, +7.2, +6, +2.5, -5.2 and -29.2 pA respectively. The last value was the difference between an anionic current of -76.3 pA and the cationic current of 47.1 pA. These values correlate well with experimentally measured

values(73). If this simulation scenario were carried out in a planar bilayer experiment, the patch clamp amplifier would show a current reversal between the +/- 7 e^- and +/- 8 e^- charge imbalances. This indicates that the anion flux has exceeded the cation flux. Since the CompEL algorithm automatically swaps crossed ions back, thereby maintaining the established charge imbalance, these simulations provide insight into the mechanisms establishing the steady state fluxes. In experimental conditions, no net current flow simply means equal and opposite counter ion diffusion, which also means decreasing concentration and charge gradients.

Nevertheless, clearly pore size and charge imbalance effect a significant change in pore selectivity, yet a larger challenge is to identify and understand the all the mechanisms responsible for this change. We believe the feature that stands out in **Fig. 6**, the constantly increasing slopes of the anion trend lines, are an important clue. Compared to the nearly invariant slopes of the cation trend lines over the span of increasing charge imbalance, determining the causes for these differences is essential.

All other things being equal, considering ions of equal concentration interacting only with water in the absence of electric fields, differences in fluxes through any surface should vary only with respect to their individual diffusion coefficients. Animations of pre-pores and pores mechanics reveal cations entering the pores freely, whereas there is clearly an energy barrier for anion entry into smaller pores. At larger pore sizes, diffusion appears to be unhindered for both ion species, since larger pore dimensions make it possible for ions to pass through the pore interacting only with water molecules and the electric field. In the absence of electric fields and molecular interactions, different ions would exhibit selectivity for the species with the higher diffusion rate. To understand by what degree this

effect is present in our simulation data, a system with 11,480 water molecules, 11 chlorine ions and 11 sodium ions was simulated for 100 ns to determine the diffusion rates of the ions in this system. The GROMACS *msd* program calculated diffusion coefficients of 2.33×10^{-5} cm^2 sec^{-1} for sodium and 4.0×10^{-5} cm^2 sec^{-1} for chlorine at 303.15K. This equates to chloride ions having a ~73% higher diffusion rate in the simulations with our set of FFs. Thus, for 283 cations observed to cross the membrane, about 489 anions would be expected to diffuse through the pore if it has reached the purely diffusive regime. The actual value of 458 is about 94% of 489, so the steadily increasing trend of the anion permeations is not inconsistent with the difference in the ion diffusion rates. Applying the same analysis to the permeation data for pores with SCMTRs, (**Fig. S6**) 597 anion and 226 cation permeations are observed at the same charge imbalance. For 226 cation permeations, 394 anion crossings would be predicted when the system is in the purely diffusive regime. This is well short of the 597 observed crossings. We speculate the higher electric field in the pore resulting from the pore centric concentration of the electric flux will be identified as the cause of these transport rates in excess of a purely diffusion-based regime.

Potential simulation size artifacts

To assess how the finite size of the simulation box affected our results, a set of simulations were conducted with a system four times the size of the original (doubled in the *x*- and *y*-dimensions and identical in the *z*-dimension). Since pore formation in the original system required a charge imbalance of +/- 6 e^-, the larger system was created with a charge imbalance of +/- 24 e^-, anticipating an equivalent electric flux would lead to similar pore formation dynamics. While the large and small systems had the same transmembrane voltage, pores were surprisingly formed much less often in the larger

systems. As expected, the +/- 24 e^- charge imbalance gave rise to a pore with equivalent increases in area. The pore area is proportional to the charge imbalance and thus the electric flux. The higher relative flux of chloride ions through the pore compared to sodium ions moving in the opposite direction caused a significant net flux of water molecules from one bath to the other, because solvated ions moving through the pore are accompanied by their water solvation shell. **Fig. S7** in the Supporting Material (pg. 122) depicts the extreme outcome of this scenario in which one bath shrinks while the other grows. Relative chloride flux between the larger and smaller systems is significantly greater than a factor of four. We hypothesize that the chloride ion flux increases due to changes in the ion concentration gradient caused by the net flux of water. Therefore, CompEL may not be a viable method to observe quasi-steady state ion flux through sufficiently larger pores. However, modifications to the CompEL protocol could be made in the future to control not only the number of ions in each bath but also the number of water molecules.

SCMTR effects on pore formation

While investigating pore formation kinetics and the effects of SCMTR on pore formation, we unexpectedly observed that NPT simulations exhibit a clear pattern of much lower pore formation rates compared to the NVT ensemble, even though the NPT ensemble is needed for simulations to reach their equilibrium box dimensions. The campaign of 240 simulations that were used to probe pore formation kinetics were each 40 ns in length. The initial 40 simulations, conducted in the NPT ensemble, developed only one pore over the total 1600 ns. Next, the same 40 simulations were repeated in the NVT ensemble using the equilibrium box size for membranes with pores (so that pore development would not be constrained by available area per lipid), and seven pores were formed. Of these seven pores,

four formed with SCMTR molecules incorporated and three formed without. Another set of 40 NVT simulations were run using the equilibrium box size for membranes without pores (at the risk of making lipid molecules and pores compete for area), and counterintuitively produced 16 pores. The next 40 NVT simulations were run at the largest box dimensions seen in a frame in an NPT trajectory and again counterintuitively produced five pores. Finally, 80 NVT simulations were performed at the original equilibrium box size (for membranes with pores), but the charge imbalance was reduced from +/- 6 e^- to +/- 5 e^-, producing three pores. In total, the data indicate that when simulating pore formation, the choice of ensemble and the surface area per lipid are both critical variables. The NVT ensemble, more crowded lipid membranes, and higher transmembrane potentials all encourage pores to form more quickly.

SCMTR effects on pore stability

To quantify stability of pores formed by electroporation, pore disintegration kinetics were explored. Ten simulations were conducted for systems containing two equally sized pores at equilibrium with a +/- 4 e^- charge imbalance for pores with and without SCMTRs. At time zero the charge imbalance was eliminated and all but one of the twenty pores without SCMTRs disintegrated within 12 ns. The average lifetime for the non- SCMTR pores was 6.2 ns. The decay of the SCMTR-containing pores was slower by a factor greater than three, with an average pore lifetime not less than 22 ns, and two pores were still open after 40 ns. For these simulations, the pore water molecules were counted as an indicator of pore size, and pore closure events were tabulated. These data are presented in **Fig. 7** which clearly indicates enhanced pore stability and longevity attributed to the SCMTR molecules' inclusion in the pore structure. As previously discussed, we also

attribute this increased stability to the SCMTRs hydrogen bonding to multiple lipid headgroups in the pore. We hypothesize that SCMTR's enhancement of ion permeations in previous experiments(29) can be explained by the stabilizing effects observed in simulations. However, we see insufficient evidence to confirm or dispute that SCMTR enhances pore formation.

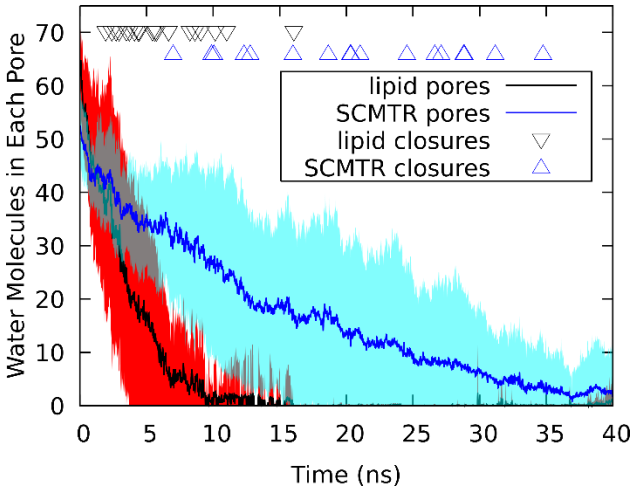


Fig. 7: *Pore disintegration dynamics after removal of charge imbalance. We hypothesize hydrogen bonding between SCMTR molecules and lipid headgroups in the pore play a key role in slowing the disintegration of membrane pores in the absence of any electric field. Shaded regions indicate the standard deviation over 10 simulations and the trendlines are averages.*

Area per lipid of membranes with pores

We had hypothesized that the NPT ensemble would more realistically simulate all phases in the lifetime of membrane pores formed in response to transmembrane electric fields. Semi-isotropic pressure control permits simulated membranes to expand or contract as necessary to reach an equilibrium surface area with their surroundings. These fluctuating areas were extracted from the full trajectories of all 144 production runs and plotted in **Fig. 8**.

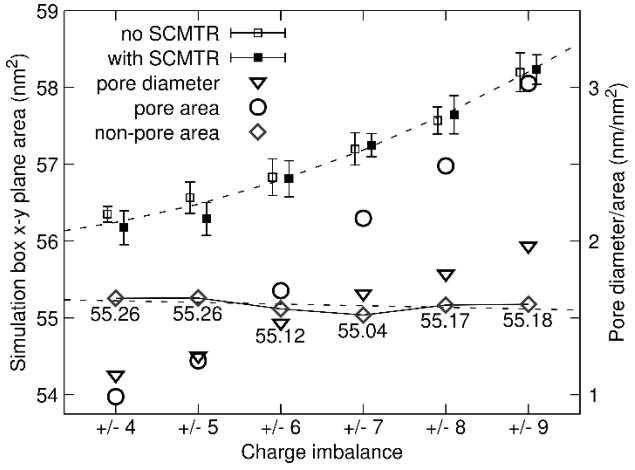


Fig. 8: *Simulation x-y plane box area vs. charge imbalance. Error bars are standard deviations of the average x-y plane area for each of 12 simulations. Note how the total pore area as measured by water molecules fully accounts for changes in simulation cross-sectional area.*

As hypothesized, our simulations show that different simulation box sizes are necessary to allow pores to reach their equilibrium sizes without being constrained as the charge imbalance changes.

To determine if counting core water molecules is an accurate measure of the pore size, we checked for the self-consistency of multiple measurements. First, we used the counts of core water molecules to calculate pore area. We define the pore area as the number of water molecules in the core divided by the bulk density of water divided by the height of the core, which was defined as 1.5 nm in our counting algorithm. In **Fig. 4**, we demonstrate that the number of core water molecules is extremely highly linearly correlated with the electric flux. Second, we calculated the difference in box x-y area that is caused by the formation of a pore, which we use as a second measure of the pore area as in **Fig. 8**. The consistency of these two measures indicates that counting core water molecules is in fact an accurate measure of pore size. Furthermore, this supports our earlier claim that the equilibrium pore size is a function of the electric flux which in our simulations is proportional to charge imbalance.

For simulations with charge imbalances of +/- 4 and +/- 9 e^- (averages of 50 and 152 water molecules), the pore areas are calculated to be 0.99 and 3.03 nm^2. A quadratic regression of the data in **Fig. 8** predicts box x-y areas of 56.2 and 58.2 nm^2 at charge imbalances +/- 4 and +/- 9 e^-. If the area of the pore completely accounts for differences in box x-y area, then the remaining area containing lipids is found by subtracting the pore area from the x-y box dimensions, and this value should be nearly constant. We thus calculate the lipid area to be 55.3 and 55.2 nm^2 (at +/- 4 and +/- 9 e^-), which are statistically

the same when considering the error bars. This result is consistent for all intermediate charge imbalances.

CONCLUSIONS

MD simulations of lipid membrane pore formation in response to electric fields induced by transmembrane charge gradients have been conducted using the CompEL method. These simulations have provided molecular-level insight into restructuring of lipid membranes in response to electric fields. The CompEL method is a step forward in more realistically modeling membrane structures, dynamics, and steady state properties. Aside from the ability to model steady state transport through membranes it enables studying membrane interactions with controlled, steady, finite amounts of electric flux. A key finding of our work is that the electric flux determines the equilibrium pore size.

We have developed and implemented a new method of determining pore size by counting pore core water molecules. However, our method requires significant proficiency with the GROMACS *select* command syntaxG11, and these commands are much more complicated than any examples we have seen published. This method has enabled us to verify a compelling correlation in MD between equilibrium pore size and the electric flux. Because pore growth to equilibrium size is not instantaneous, we speculate initial formation times of individual pores significantly effect pore size distributions observed in practice. If so, the connection of pore size to electric flux could be useful for improving the experimental and clinical practice of electroporation, because to the best of our knowledge previous work has not considered this variable to be a contributing factor to pore size distributions.

At smaller pore sizes chloride ions seem to encounter an energy barrier at the pore entrance. As pore sizes increase, lower probabilities of ion-pore wall interactions allow observed chloride ion fluxes to approach purely diffusive transport. We hypothesize the unquantified non-uniform electric field gradients are key variables affecting pore permeation dynamics when fluxes exceed rates consistent with unperturbed diffusion. We believe accurate electric field mapping in and around pores will be required before a better theoretical explanation of these observed ion fluxes can be postulated.

In future work, we plan to develop a tool to accurately reconstruct the exact nature of interactions among electric fields and biomolecules from the existing trajectory data. An accurate, three-dimensional description of these interactions as well as the three-dimensional electric field map would be important theoretical developments, leading to useful insights that could be exploited in practical applications. Furthermore, since coulombic interactions are known to be used by nature for control and modulation of many essential membrane specific processes (endo- and exocytosis as well as transport of species via membrane proteins), a more complete understanding of how the membrane itself interacts with the local electric field is beneficial.

Although the NPT ensemble was crucial to establishing the link between electric flux and pore size, we noted it also has a profound effect on pore formation rates. Trying to account for the extreme variation in observed formation rates accounted for the bulk of the 240 simulations performed hoping to ascertain pore formation kinetics. Our results indicate that close to the equilibrium area per lipid, pore formation rates observed using the NVT ensemble counterintuitively increase as that area becomes more constrained. We speculate being more constrained could be creating simulation artifacts manifesting as if

the membrane is experiencing more extreme pressure fluctuations in the x-y plane, contributing to larger fluctuations of the initial pore formation energy barrier.

SECTION 2

POST-PROCESSING AND VISUALIZATION OF ELECTRIC FIELDS AND COMPUTATIONAL X-RAYS

INTRODUCTION

Two significant hurdles to mapping electric fields in three dimensions have presented themselves, specifically extraneous bulk system motions and exorbitant electric field calculation requirements. Capabilities developed for high accuracy measurement of pore core water molecules from electroporation, despite bulk system motions (see Section 1), have laid the foundation of a method to completely remove nearly all system motions relative to a selection of molecules such as those surrounding a membrane pore. Analysis is performed frame by frame, efficiently enough to make the approach viable for larger scale analysis of MD trajectory data. The calculation of the non-uniform electric fields

developed by pores, the interrogation of interactions between biomolecules and non-uniform electric fields, and the elucidation of pore formation mechanics and equilibrium conformations has become tantalizingly ponderable. The exorbitant cost of calculating electric fields with no symmetry is however a significant hurdle.

While studying electroporation, **Fig. S4** (page 119) was produced to understand the contributions of different membrane regions to the overall electric field. Since the GROMACS electric *potential* utility was written assuming planar symmetry (which massively simplifies the solution of Gauss' equation to have an analytical result), the utility produced results that were not valid in all dimensions for a pore-containing membrane, lacking planar symmetry. Therefore, I started investigating the possibility of calculating the real non-uniform electric field map in three-dimensions. The traces in **Fig. S4** results were calculated using 1000 slices along the z-dimension of the simulation box. Knowing the scaling issues, tests were performed at 200-layer intervals between 200 and 800 slices to assess the number of slices really needed. Statistical noise became evident at 200 slices, indicating that the field should be calculated at a resolution of 0.5 Å or smaller. Benchmarks of preliminary code indicated that it would take 12 years to produce one frame of three-dimensional field data at the target resolution! It was not a viable plan, so it was dropped. Later, I postulated that one could exploit the orthogonality along the axes of the charge coordinates to reduce the scale of the calculation by two orders of magnitude, thus resurrecting the idea. The demonstration of this concept has born much fruit, in the form of the electric field maps, covered in Part 1 of this Section, and the unanticipated computational X-rays, covered in Part 2 of this Section.

PART 1 – MAPPING ELECTRIC FLUX FROM SIMULATION TRAJECTORIES

PROGRAMS and UTILITIES

To map three-dimensional interactions between biomolecules and electric fields, MD trajectories had to be stabilized to remove all extraneous motion relative to the biostructure of interest, and the biostructure had to be centered in the simulation box in every frame. The first two steps in this process were developed to extract accurate pore core water molecule counts for electroporation, the details of which are documented in the Supporting Materials of Section 1. The details of the remaining steps are documented in the Supporting Materials of Section 2. In **Fig. 9**, we first present a high-level flowchart of the sequence of processing steps. These steps are achieved by four general techniques.

The first class of techniques includes all the GROMACS programs and utilities required to produce and manipulate native GROMACS simulation files. The second class includes all *Bash* (Bourne Again Shell) scripts written to automate post-processing activities. The third class includes scripts written for the plotting and visualization programs *gnuplot* (an open source plotting program) and VMD, to verify that simulations finish without major errors, to verify data integrity at intermediate steps and to produce the final graphs, plots and animations. The fourth class includes all the *AWK* (programming language) routines when custom numerical analysis of data is required to generate result files. Documentation including a master script for the use of the GROMACS, *Bash*, *AWK* and *gnuplot* commands developed to do this work is in the Supporting Materials for Section 2 with full listings of all routines published on GitHub.

To appreciate the significance of these results, a review of the standard tools before this work, which analyzed the electric field in only one dimension, is called for. Membrane

potentials play a critical role in the maintenance of ion concentration gradients within cells, the conduction of nerve signals to and from the brain, and in inducing pore

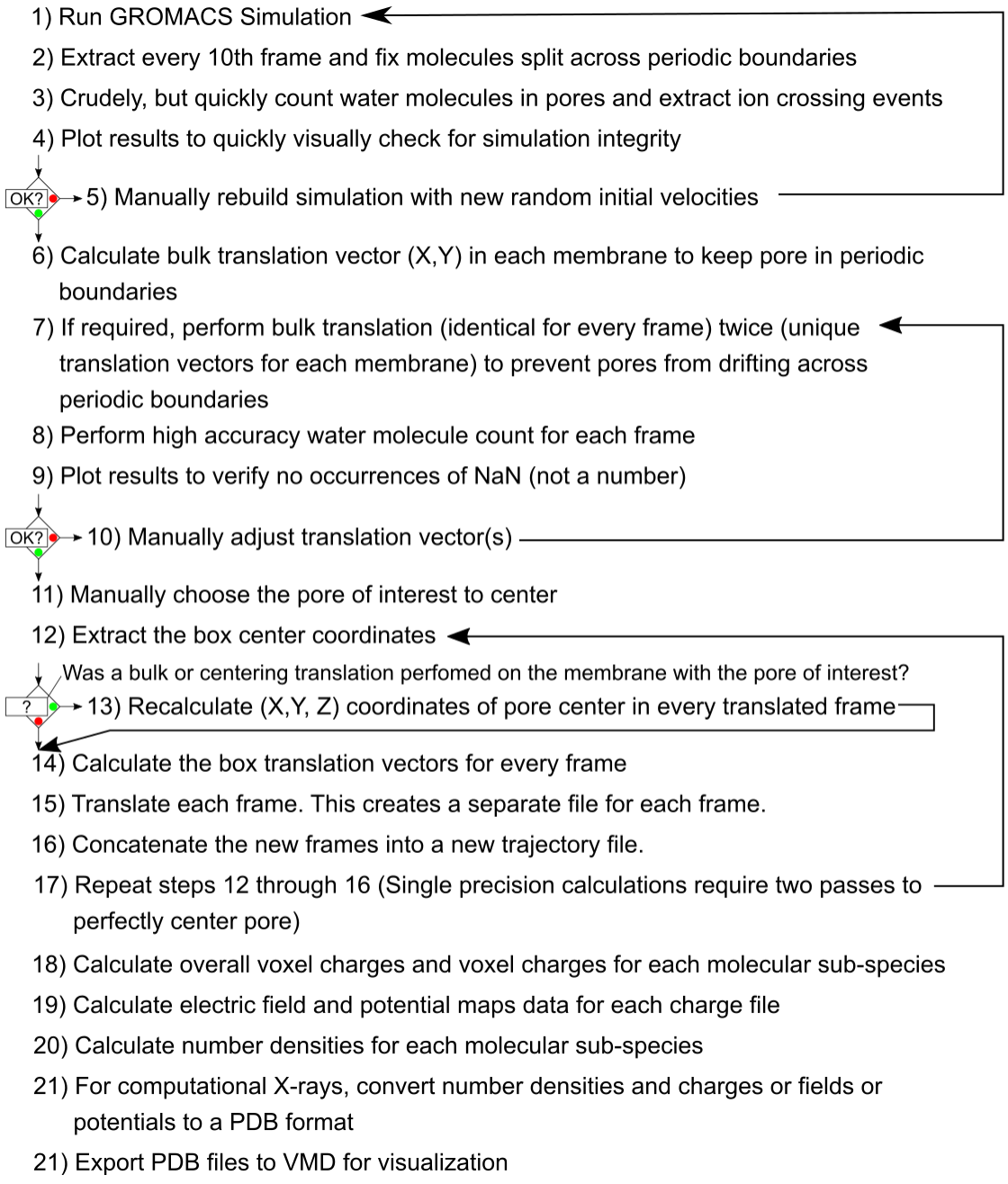


Fig. 9: *Sequence of post processing steps applied to simulation trajectories to produce electric field maps and computational X-rays. Steps 1-10 were developed initially to extract results from electroporation simulations. Steps 11-21 were developed to achieve the goals of the dissertation proposal.*

formation as practiced in electroporation when quite large electric fields are experienced by the membrane. In GROMACS membrane potentials are calculated by the *potential* utility. However this utility assumes planar symmetry of charge distribution along any given axis (i.e. it produces a one-dimensional potential profile). This limitation is completely understandable, since analytical solutions of Gauss' law (eq. 1) only exist for conformations where the charge distribution is highly symmetric (infinite plane, infinite cylinder, spherical surface). **Figs. 10** and **11** show the results of these one-dimensional calculations for membranes with and without pores. Poreless membranes possess charge distribution symmetry in the x-y plane, making field and potential calculations along the Z axis straightforward.

$$\Phi = \oint E \cdot \cos(\theta)\, dA \qquad (1)$$

Fig. 10 accurately represents the electric field, and a membrane centered in the simulation box (Z = 10 nm) experiences a roughly 2.0 V transmembrane potential, plotted as the thick green trace. The other six traces show the electric field and potential profiles for the ions, lipids and water molecules. Note that the individual moieties are exerting and interacting with fields and potentials many times larger that the overall field. These strong coulombic interactions are the forces that give the membrane its amazing ability to self assemble and its self-healing properties. The traces presented in **Fig. 10** give insight into the potential energy available to drive the incorporation of membrane protiens. A pair of the interactions that are easily distinguished are the sharp black and cyan traces that show the opposite orientations of the lipid and water dipoles.

Fig. 11 shows a similar set of traces in addition to charge distributions that cause the fields and potentials for a membrane containing a pore. While **Fig. 11** is interesting and informative, it is flawed. Note for instance that the calculated transmembrane potential,

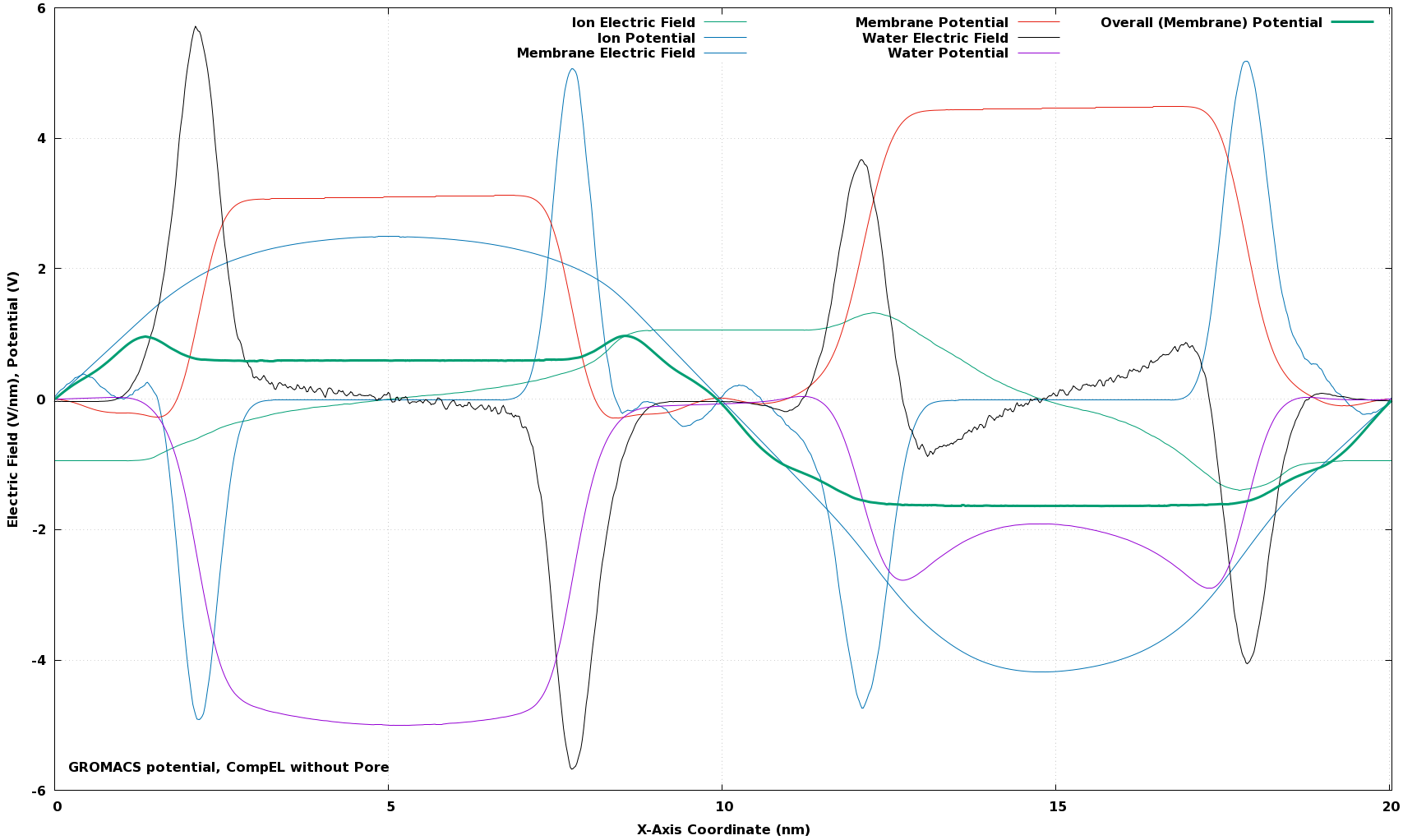


Fig. 10: *Composite electric field and potential plot for centered membrane with no pore. Thick green trace shows overall potential. Y-scale is volts, x-scale is nm. The membrane potential is the difference in the overall potential between 5 and 15 nm; ~2.0V. Note the overall electric field magnitude is a fraction of the fields present due to individual moieties. This significance is discussed in the text.*

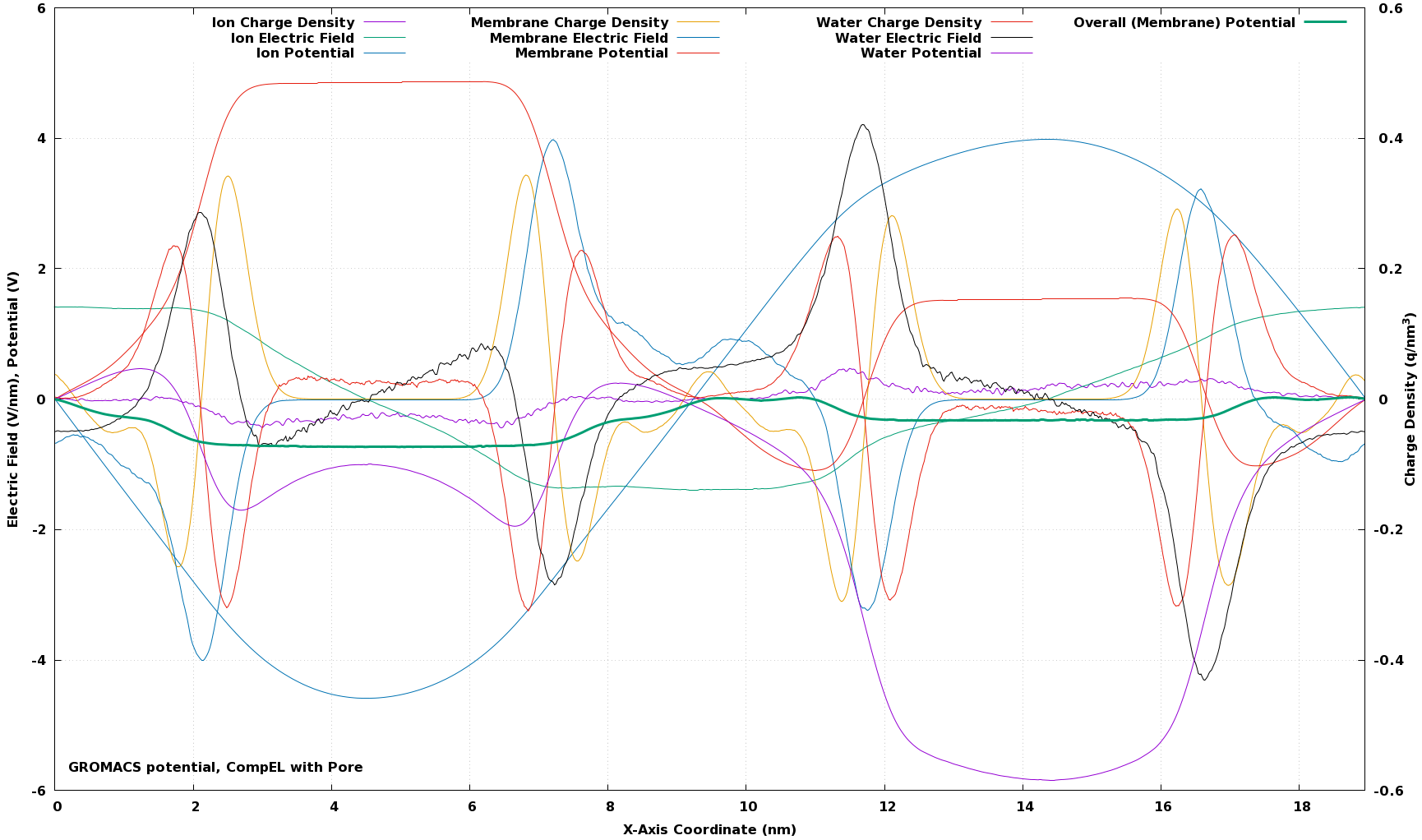


Fig. 11: *Composite electric field and potential plot for centered membrane with pore, generated by the GROMACS potential utility. The first figure is valid, because of the system symmetry along the axis analyzed. This figure is not valid, because pore formation disturbs the planar symmetry that the potential utility assumes is present.*

indicated by the difference in value of the thick green trace on either side of the central membrane, shows a transmembrane potential of ~0.45V. Both plots were from trajectories that had equal charge imbalances maintained by the CompEL algorithm during a simulation. How can they have such apparently different transmembrane potentials? In **Fig. 10** all the electric field vectors are parallel to the z-axis, along which the fields and potentials are being evaluated. In **Fig. 11** the field vectors are not all parallel to the z-axis. Some vectors are parallel, others are orthogonal, and others occupy nearly every angle between these two. The fields are non-uniform, and a one-dimensional profile cannot possibly convey (visualize) reality. Furthermore, the biomolecules and non-uniform electric fields are not acting independently of each another. Many molecules have either permanent dipoles or are zwitterionic, as with DOPC. Dipoles are interacting with the electric fields, changing molecular orientations, which in turn changes the fields in an intricate balancing act. The rearrangment of molecules occurs as they move toward the lowest free energy arrangement, except briefly when overcoming energy barriers. How do we reckon the complexity of this situation in all three dimensions from the data, as dictated by the equipartition theorem of energy distribution?

THEORY

Coulomb's law (eq. 2) describes the forces (F) that electrical point charges (q) separated by a distance (r) exert on one another in a vacuum where k (eq. 3) is a constant ($\sim 9 \times 10^9\ N \cdot m^2/C^2$) (Coulomb's constant) derived from the permitivitty of free space (ε_0) for geometries with spherical symmetry, and $\hat{r}_{12}$ is the unit vector between the two point charges. In electrostatics, force also equals the electric field times the charge.

$$\vec{F}_{12} = k \cdot \frac{q_1 q_2}{r^2} \hat{r}_{12} \tag{2}$$

$$k = \frac{1}{4\pi\varepsilon_0} \tag{3}$$

Note that Coulomb's law gives the equal and opposite (direction) force for only two particles. A more practical way to check calculated electric fields makes use of Gauss' law (eq. 1) to check the validity of the results. To do so, the electric flux is calculated over the surface of an enclosed volume produced by a distribution of N point charges within that volume. The electric field $\vec{E}$ (eq. 4) must be calculated at every point (at target resolution) of the surface for the enclosed charges. Note that equation 4 is simply Coulomb's law with one of the charge terms removed. Note also that every term summed into $\overrightarrow{E_j}$ requires the absolute distance between points *j* and *i* calculated from 6 cartesian coordinates. Equation 4 (the brute force method) inconviently scales as N^6.

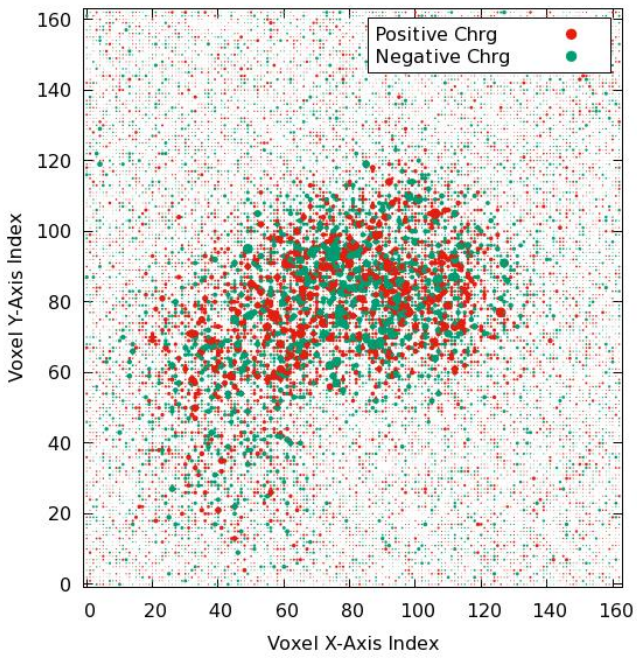
$$\overrightarrow{E_j} = \sum_{l=1}^{N} k \frac{q_l}{(d_{jl})^2} \hat{r}_{jl} \tag{4}$$

For our purpose, since the simulation calculations take into account the periodic boundaries, we can consider the periodic boundaries to be the Gaussian surface to which we will apply equation 1. Gauss showed that any distribution of charges that have zero net charge within a surface will have zero net flux throught that surface; Φ in equation 1 will be equal to zero. Since the simulation has no net charge, equation 1 provides a simple method to check the integrity of any routine designed to apply equation 4 to the simulation results.

DEVELOPMENT and DEMONSTRATION

To address previously mentioned scaling issues, I postulated that with charge voxels distributed equally along rectilinear coordinates orthogonal to each other, the scaling factor of the brute force method could be reduced. The initial attempt to implement a voxel-based method failed when results indicated net flux through the simulation boundaries, violating Gauss' law. The voxel method involves a highly complex indexing scheme to work through the volume space and properly reckon the relative position of every voxel in the space index-wise. Pitfalls abounded in the possible mishandling of the ever-changing relative position of the periodic boundaries to a given voxel during any given iteration of the calculations. In problems of this complexity, without validated tools, nothing can be assumed.

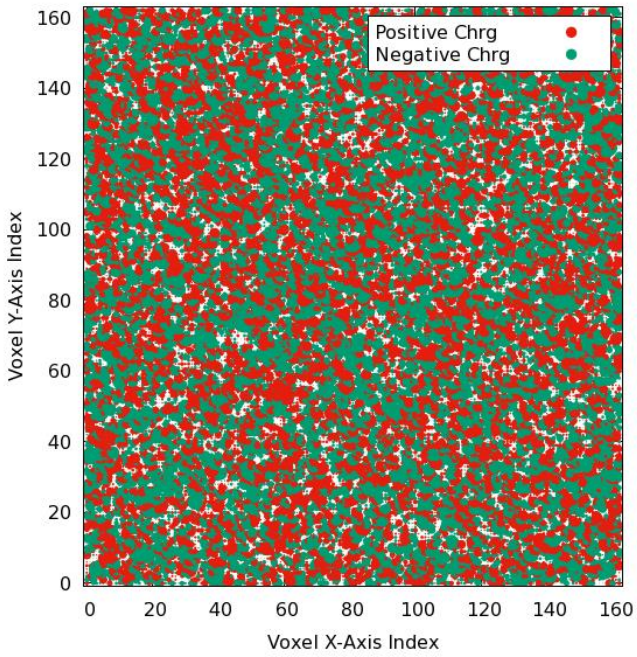
An extensive troubleshooting session followed. Were errors caused by bad data, or was the algorithm flawed? Was everything integrated correctly? I used graphical visualizations to check the data. However, the data set contained over 77,000,000 voxels and was 100MB in size. The graphing program could not plot the data because it could not read a file that large. There is a reason scientific computing is done on supercomputers in a UNIX operating system. The problem of calculating electric fields on this scale verges on the realm of "big data", and it comes with its own special set of difficulties. Record length limits, matrix formats, utility program limitations, disk I/O throttling on computer clusters, and coding errors are some of the challenges that must be overcome. In simulation work, any of these challenges that compromise the data integrity give rise to the well-known problem of "garbage in / garbage out". In other words, bad data produces bad



models. To avoid wasting time with garbage data requires constant vigilance, testing, and verification of even the most mundane aspects of data processing.

It was while I was developing a method to visually check the integrity of the data that I inadvertently created a visualization

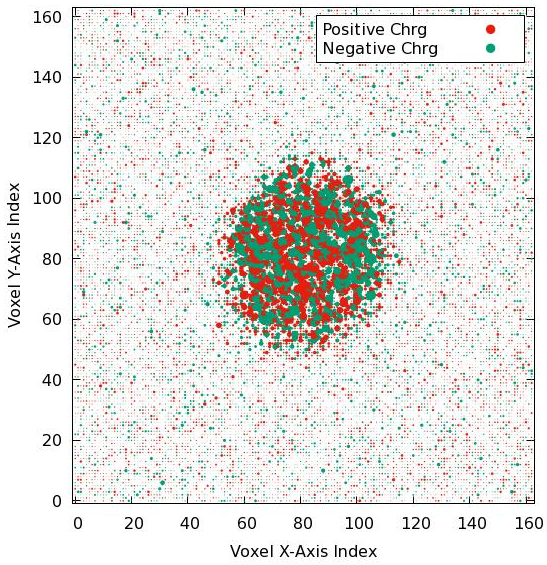
Fig. 12: *First quartile slice. The higher density indicates the presence of something other than lipid tails*



technique that I dubbed the "computational X-ray" (compXR). Because the data file was too large, only charge polarities in select x-y slices of the simulation box were plotted. A region with a BLM pore and a region with ions solvated in a water bath are presented in **Figs. 12** and **13**. **Fig. 13** appeared to be

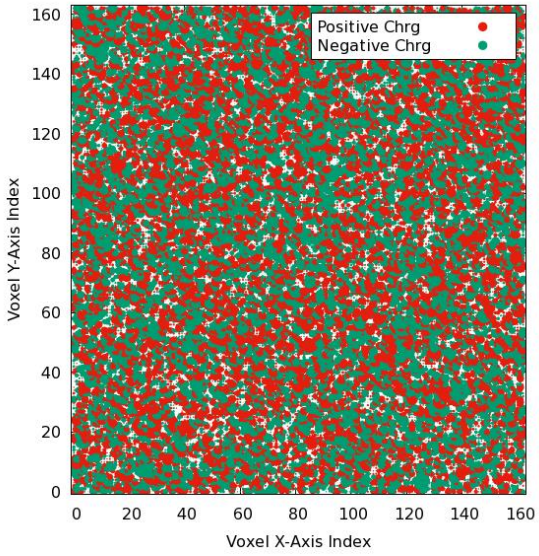
Fig. 13: *Second quartile slice. In this region, the higher density should be caused only by ions.*

correct based on physical intuition and experience from visualizing the trajectory. However, **Fig. 12** was unimpressive, until the realization that the trajectory stabilization



algorithm might have been performed on another portion of the simulation box (i.e. a second pore in a separate membrane). After plotting **Figs. 14** and **15**, the real picture came into focus, and these images were the first demonstration of the knowledge that could be gained from an analysis of the electric fields within these BLM

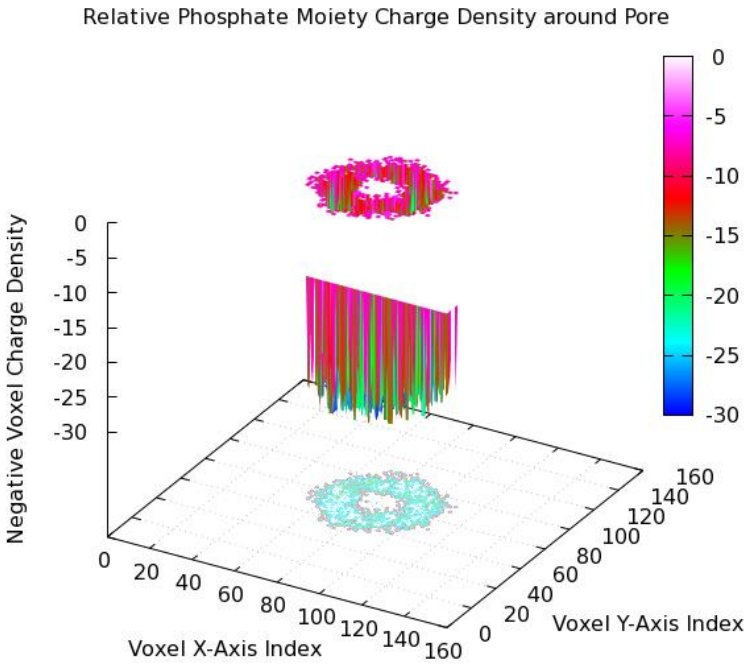
Fig. 14: *Third quartile slice. The well centered and sharp-edged dense region hinted at better things to come.*



simulations. Early images were rather crude, so more effort was spent to develop a way of getting a better visualization of the data. To create a superior visualization throughout the volume of the simulation box, I needed to be able to plot transparent pixels whose opaque fraction would cumulatively add as other transparent pixels were plotted on top

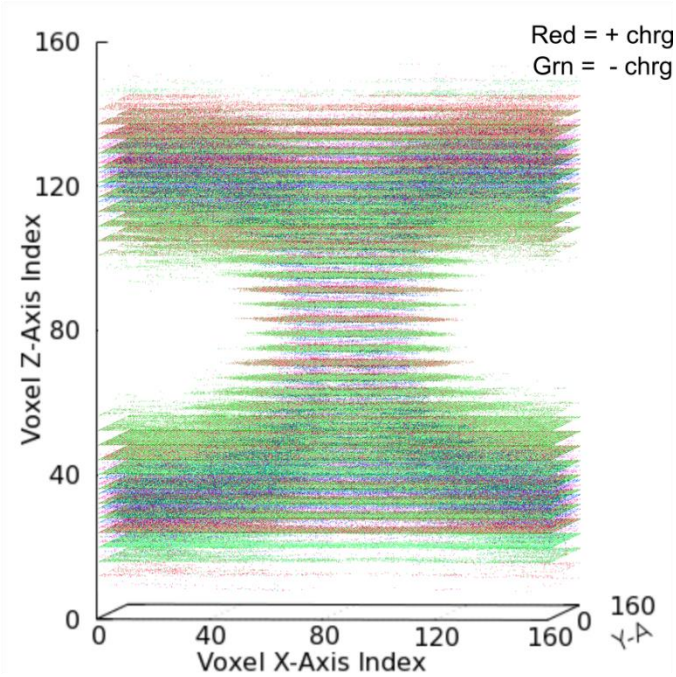
Fig. 15: *Fourth quartile slice. Exactly like **Fig. 13** and exactly what was wanted.*

of them. The native plotting program on UNIX systems, *gnuplot*, did not work well for this specific problem since its native transparency does not act in the desired fashion. The initial plots attempting to exploit pixel transparency showed a throated pore as in **Fig. 16**, but it was clear *gnuplot* had not been designed to natively express the transparency functionality



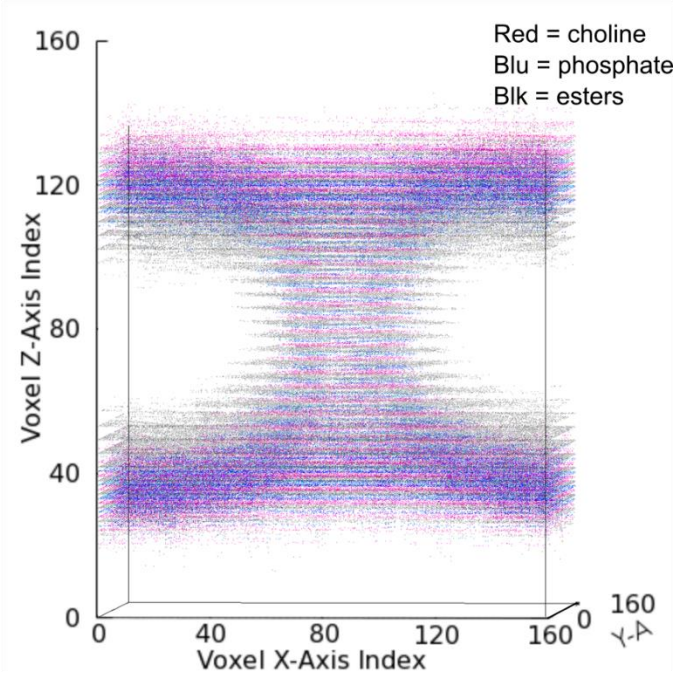
needed. Non-transparent pixels were adopted and the viewing angle was adjusted to minimize the number of overlapping pixels. The resulting figures, a representive of which is illustrated as **Fig. 17**, had some promise except charge colors should not be changing sign. Instead, in the

Fig. 16: *Final attempt to plot integrated charges using transparent pixels. The phosphate moiety shown here with the sharp pore edges further confirms precise frame centering.*



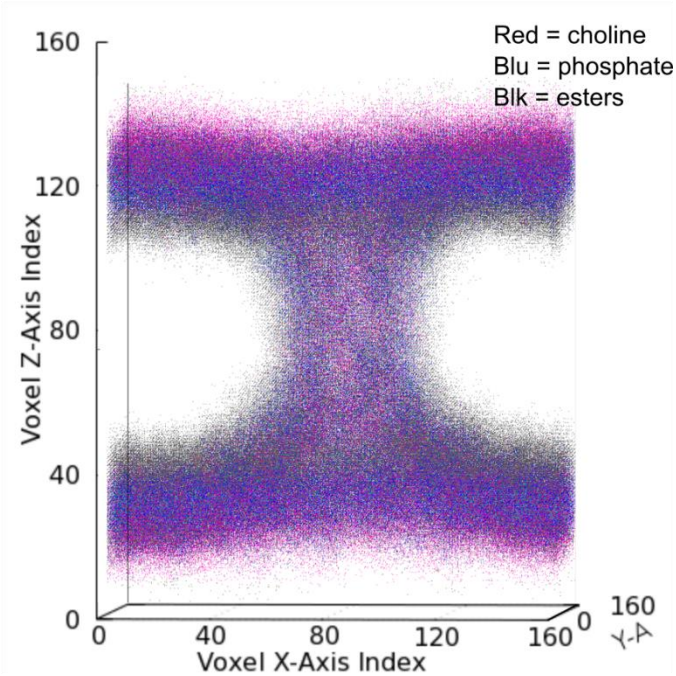
next iteration the color was set by the moiety of the molecules instead, initially using the cholines, phosphates and esters as in **Fig. 18**, which showed even more promise. For these figures, a method had been developed to read and plot the data layers individually into a multiplot file type, circumventing the initial

Fig. 17: *First attempt at plotting integrated data with non-transparent pixels. At this point I was trying to still use the color scheme used in* ***Figs. 12-15****.*



file size problems. Although I knew it would take a while to process, there was no longer an obsticle to plotting every layer of the data. When **Fig. 19** was about 40% rendered, it became apparent that this method was wildly successful. The potential of an improved method of visualization had emerged, and it immediately

Fig. 18: *Second attempt at plotting integrated data with non-transparent pixels. Now pixels represent any non-zero charge and the colors are set by moiety.*



spawned new hypotheses about the exact molecular interactions that underlie electroporation. The density profiles it revealed could be combined with the moiety data for unprecedented visualization and analysis.

After verifying the quality of the data, the electric field

Fig. 19: *Image that sparked the concept of Computational X-ray's. Seeing the detail appear in the core region is what gestated the insight to calculate and plot number densities of the various moieties of interest*

visualization could proceed. Two approaches were undertaken for the rendering of the electric field map. To generate electric field data for validating the newly developed orthogonal based approach, a brute force routine was programmed at a much lower resolution.

The results of this lower resolution brute force method were initially checked by calculating the net electrix flux through the simulation periodic boundaries along each axis. Gauss' law dictates that for a charge neutral system, these net fluxes need to sum to zero, which they did within floating point rounding errors. While working on a way to visually render the first brute force based results (**Fig. 1**), electric field calculations by the brute force method were repeated at twice the resolution since they could be completed in ~48 hours per moiety using non-optimized programs and scripts.

Validation Versus Standard Practices. The electric field and electrical potential in a non-pore system were calculated by brute force to validate against the GROMACS *potential* utility. Though the results were similar, initially they did not match. This was eventually traced to an obscure bug in the GROMACS *potential* utility. A non-pore system has no water molecules in the center of the membrane hydrophobic region, hence the water charge file used by the *potenita*l utility has sections where the charge number is zero. A plot of GROMACS and new brute force algorithm results revealed the problem as shown in **Fig. 20**. When the charge reads zero (cyan), notice the GROMACS electric field (purple) abrubtly changes, which is impossible in the absence of proximate charge. However this errant field value propogates into the GROMACS integration to calculate the potential (green). Notice the plot has different values for potential at the left and right simulation boundaries, which is also impossible in a charge neutral periodic system. The error in the

GROMACS code was corrected, and I logged a bug report with the developers. Thereafter, the brute force routine and the GROMACS potential utility produced the same results.

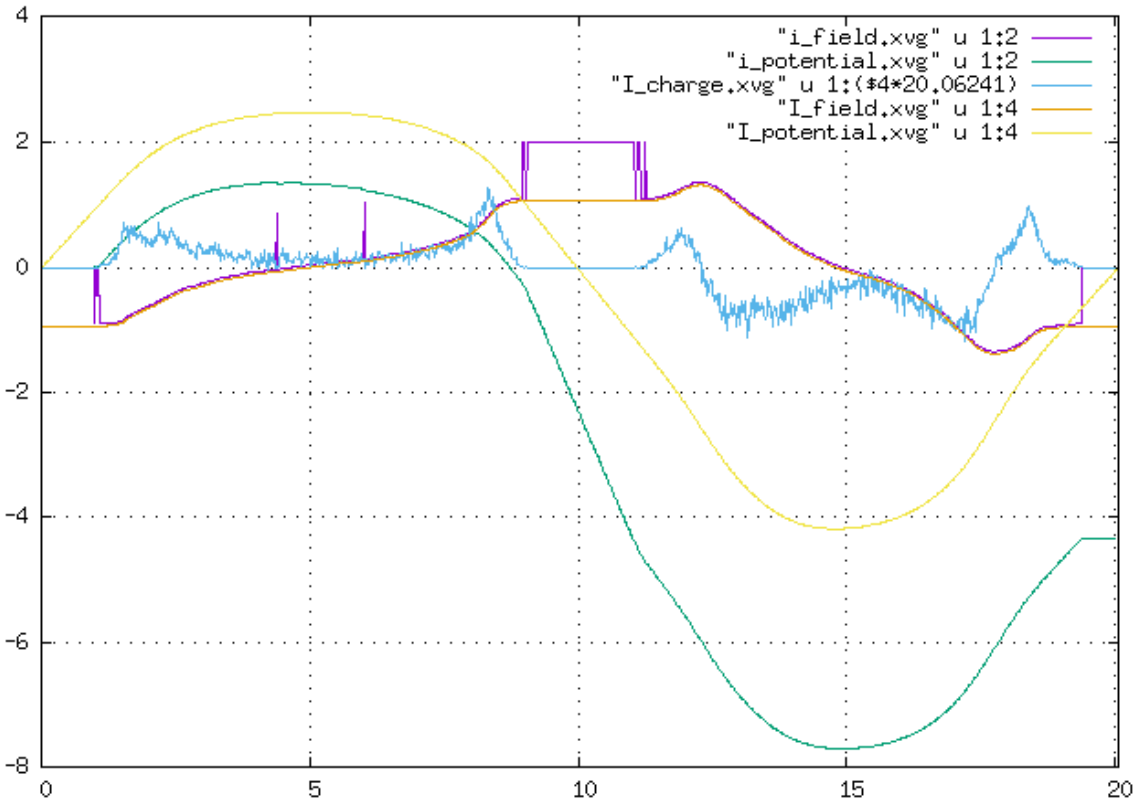


Fig. 20: *Test plot created while trying to diagnose discrepancy of results from GROMACS potential utility and new brute force method on a non-pore system. The errant field values output by the GROMACS utility were traced to a logic error in the code when encountering values that were less than macheps (machine epsilon), which is* $\sim 1.19 \times 10^{-7}$ *for single precision floating point calulations.*

With the electric field calculations validated, several figures were rendered to investigate the improved three-dimensional visualization technique. **Fig. 21 s**hows two panels of the electric field map for the water moiety plotted at two different scales and a choline panel at the same scale as the second water panel. Visualizations are at the lowest resolution, containing 50 vertical layers. The left panel of **Fig. 21** shows the full dynamic range of vector magnitudes for the water moiety only. The middle panel shows the same vectors scaled with respect to the largest magnitude vectors seen among all moieties

analyzed. This comparison allows a fully detailed representation of the water-induced field while also puting the magnitude of the water-induced field into the larger context of all moieties. Since the middle and right panels are plotted at the same scale, the relative effect of each is portrayed visually, as well as their individual contributions to the overall field.

Electric Potential Maps. Electric potential is a scalar quantity and it is measure of potential difference only, since a universal reference of absolute zero electical potential does not exist. Since the simulation volume is periodic, any periodic path (line

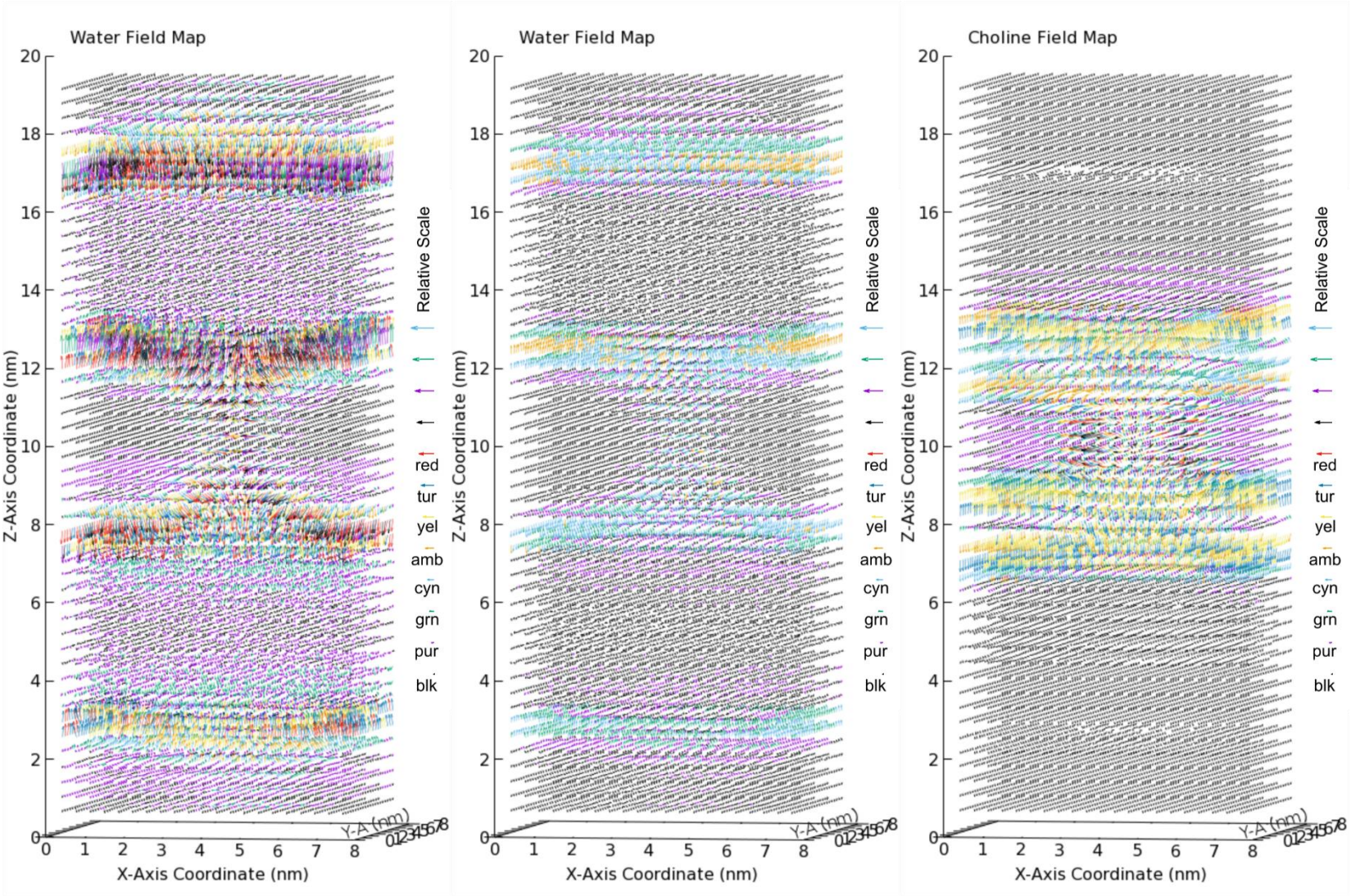


Fig. 21: *Electric field maps of water at different scales (left 2 panels) and water and choline moieties (right 2 panels) at same scale.*

integral) through that volume must return to its starting point with a net potential change of zero. To keep the post-processing math as simple as possible, boundary conditions are enforced to set the potential at the box's edge to zero. Notice in **Figs. 10** and **11** that the potentials all have a value of zero volts at the extents of the z-axis. The new three dimensional potential plots were graphed with vectors and colors. In such a representation, the vectors can be understood as a direction field that isolines cannot cross, they must

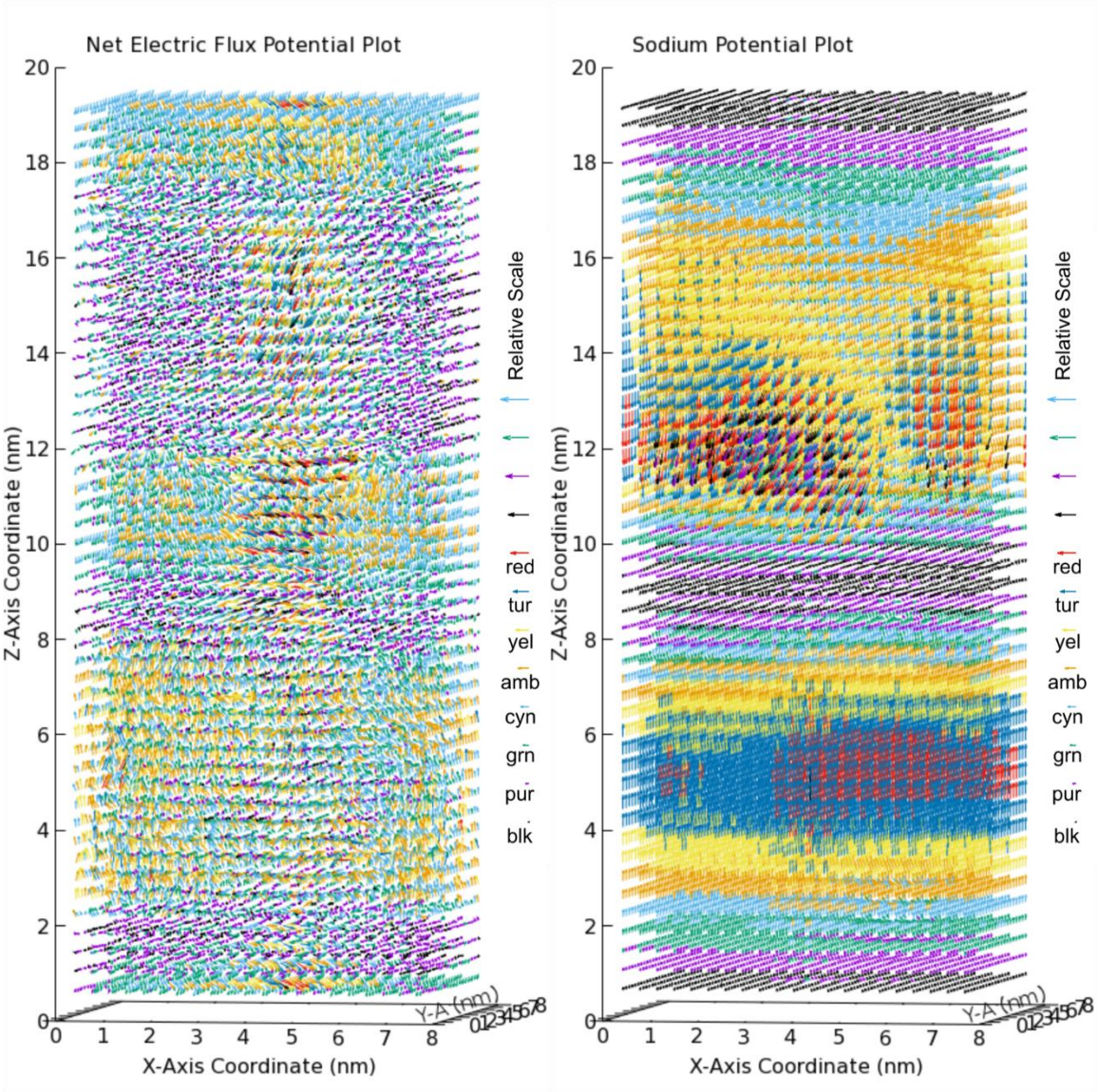


Fig. 22: *Electric potential vector map of the Net Electric Flux and Sodium potential maps. Here the full dynamic range scale for each moiety is used.*

always be parallel; mathematically the dot product of their unit vectors must always be 1. The surfaces that would intersect the points with identical color values would represent isosurfaces. With that understanding we can view the results in **Figs. 22** and **23**.

These plots correctly convey the relative magnitude of the driving forces. Individual contributions of each element of the system are masked when summed together.

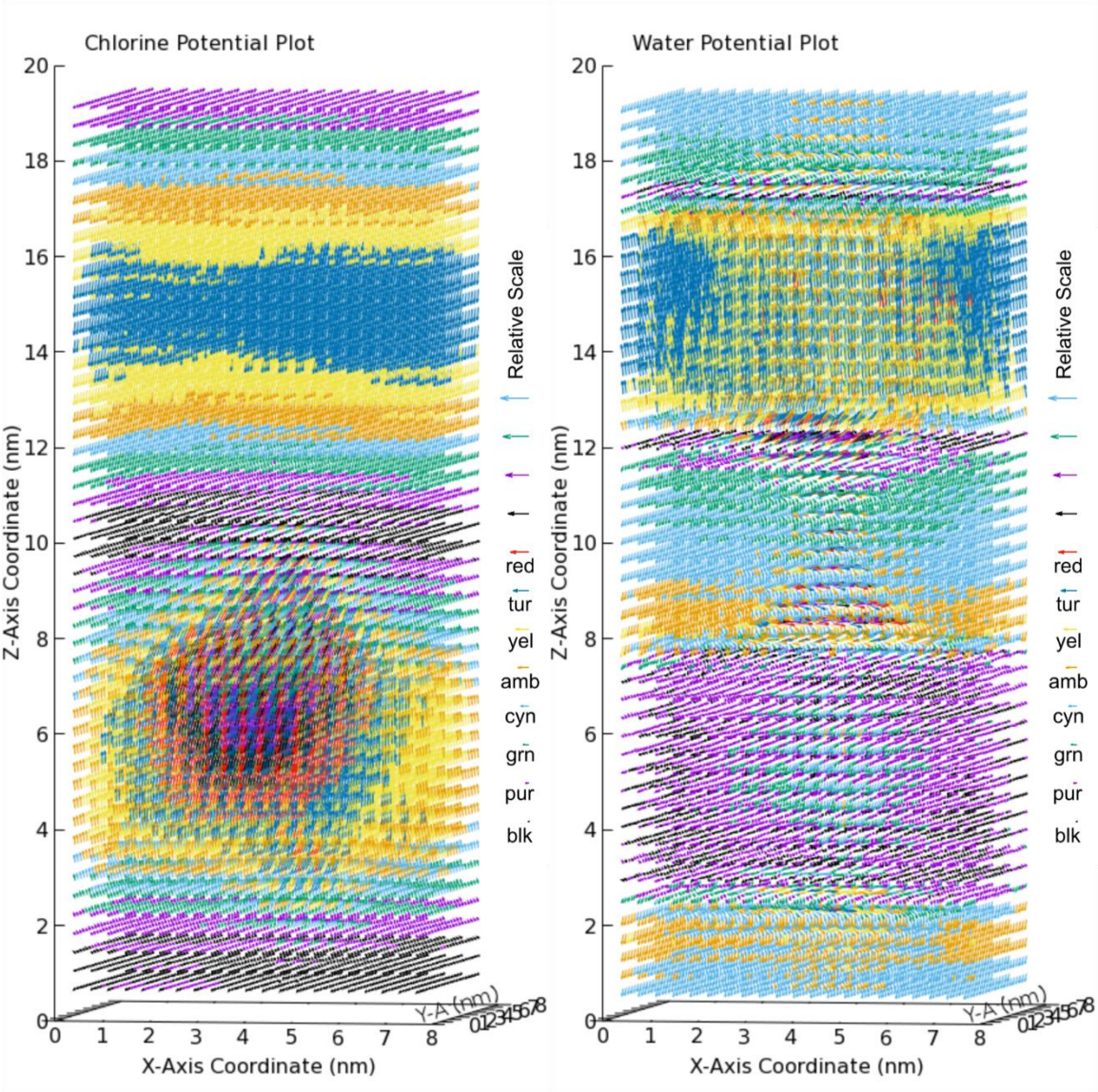


Fig. 23: *Electric potential maps of the chlorine ions and water dipoles.*

An important insight is an appreciation of the magnitude of the potential energy available in cell membranes that could be exploited in the design of biomolecular systems.

Future Directions. Much remains unknown about the intricate interactions of biomolecules and electric fields. New insights will come with the more detailed view provided by the work described here. In the future, we aim to better understand nature's molecular control mechanisms and how she modulates biological function through electric fields. Initially better visualization tools that can simultaneously combine these field and potential maps with the matter of the biomolecules interacting with them will be a top priority. Fortuitously and unexpectedly, another new capability developed from this work that can provide unprecedented physical detail of these biostructures, an imaging technique aptly dubbed "Computational X-Rays".

PART 2 – COMPUTATIONAL X-RAYS

To produce a CompXR, the same voxel space as implemented for electric field calculation is used, but the trajectory data is used to compute number densities. Additionally, with the power of the GROMACS *select* utility selection syntax, the number densities could be calculated from any group of atoms the *select* command syntax could define. Groups are not limited to any conventional definitions like molecules, species, or moieties. We refer to any selection of atoms that we analyze using CompXR as a "molecular sub-species". As described in the Supporting Material of Section 2, three parameters are passed to the electric flux charge calculation routine, *3dchrg4A.awk*, while seven parameters are passed to the other scripts. These additional parameters enable the routines to calculate the indexes of the atoms of the different sub-species (e.g. choline,

water) that warrant special consideration. To create CompXRs, for the charge file one must substitute a file that assigns each atom a value of one instead of the atoms' fractional charge. The same charge integration routine produces a data set that maps the volume space in terms of number densities. By representing the density as the opaqueness of transparent pixels and by allowing opacity to accumulate when pixels overlap, unprecedented images of the biostructure can be generated.

To the degree that MD models accurately capture realistic behavior, CompXRs are accurate representations of biostructures and not simple drawings, cartoons, or artistic renderings. **Fig. 24** contains the first image to come out of this work, and it shows the structure of a pore in a BLM generated through electroporation. The throat of the pore visible in the center is intriguing because it shows an

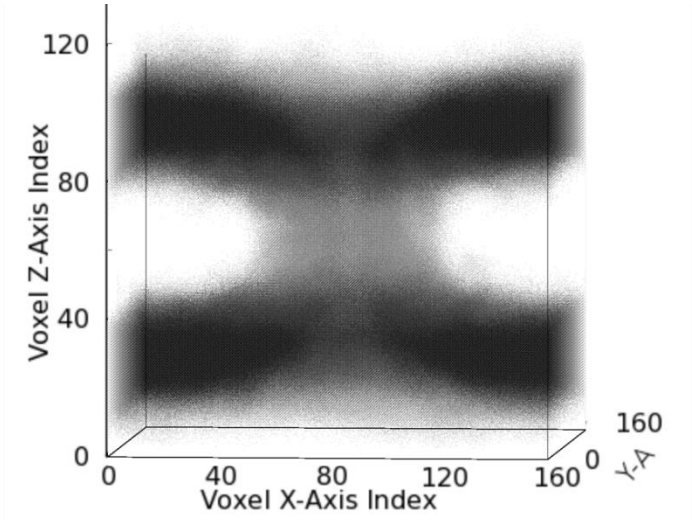


Fig. 24: *First CompXR plotted from number density data, although not without problems. Further validation revealed an aspect ratio problem, and a code error causing "double-exposure" of at least one sub-species in the pore. The central void was actually larger.*

inaccessible-to-direct-observation structure. After validating the new tools, sets of CompXRs were generated for pores with and without SCMTRs for charge imbalances +/- 1 e^- through +/-9 e^- for select moieties and their combinations from the data produced in Section 1. **Figs. 25-28** show two of these containing pores with associated SCMTRs. All the other charge imbalance figures are available on GitHub.

Figs. 25 and **26** contrast the structure of several moiety combinations at the extents of the investigated charge imbalances, +/-1e^- and +/-9 e^-. Contrast the flat upper and lower surfaces of **Fig. 25** with the curvature in **Fig. 26**. **Fig. 25** has a headgroup channel spanning the leaflets, but no pore. Thus, it is at the pre-pore stage of development. Because of the opacity issue previously discussed, these images still cannot convey the full story. Details about variations in number densities throughout the volume are lost once pixels start overlapping. Image D in **Fig. 25** clearly shows a pre-pore attribute. Showing choline and phosphate moieties, barely a thread of the pre-pore appears in the center. Compare Image D in **Fig. 25** to image D in **Fig. 26**. Clearly a cylindrical structure with an "empty" central core now exits, which represents a membrane pore. We can discern that the diameter of the central core at this charge imbalance is still less than 1 nm in diameter.

Figs. 27 and **28** contrast the same pores, but with a focus on SCMTR and water moieties. In the A images, the SCMTR headgroups are red and the SCMTR tails are green. This image represents an averaging over 40 ns of data, and SCMTRs look stationary in this time frame. Coloring portions of the system differently allows better insight to the mobility of different moieties. Lateral diffusion of lipids and membrane-bound entities can now be visualized over long trajectories. Since the CompXR method was developed for post processing, it can be applied to any simulation trajectory from the past or present.

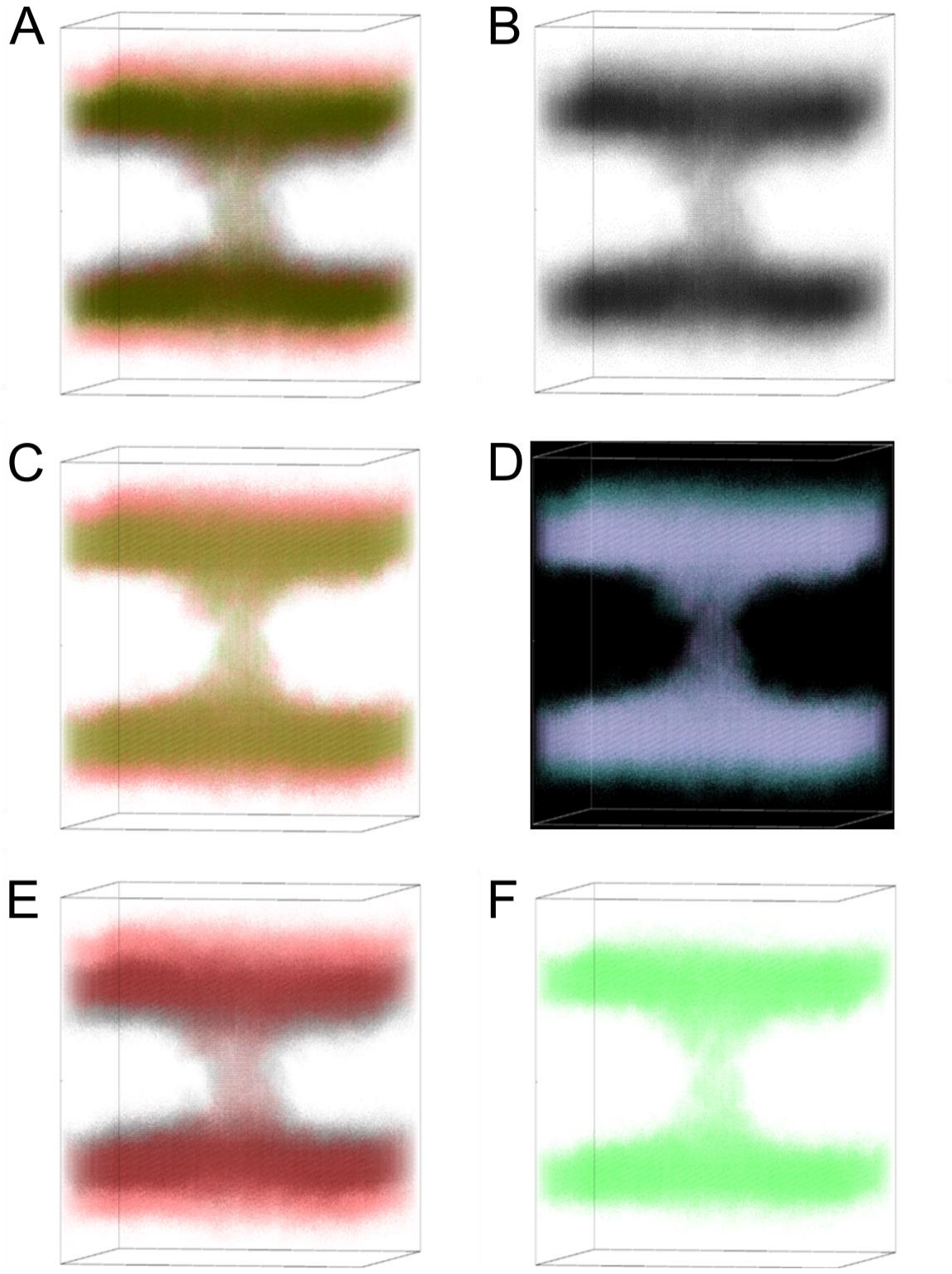


Fig. 25: *CompXRs at charge imbalance +/- 1, SCMTR's are associated with the pore. A-cholines, phosphates, esters; B-cholines, phosphates, esters; C-cholines, phosphates; D-cholines, phosphates; E-cholines, esters; F-phosphates.*
Colors: red-cholines; green-phosphates; black-esters.

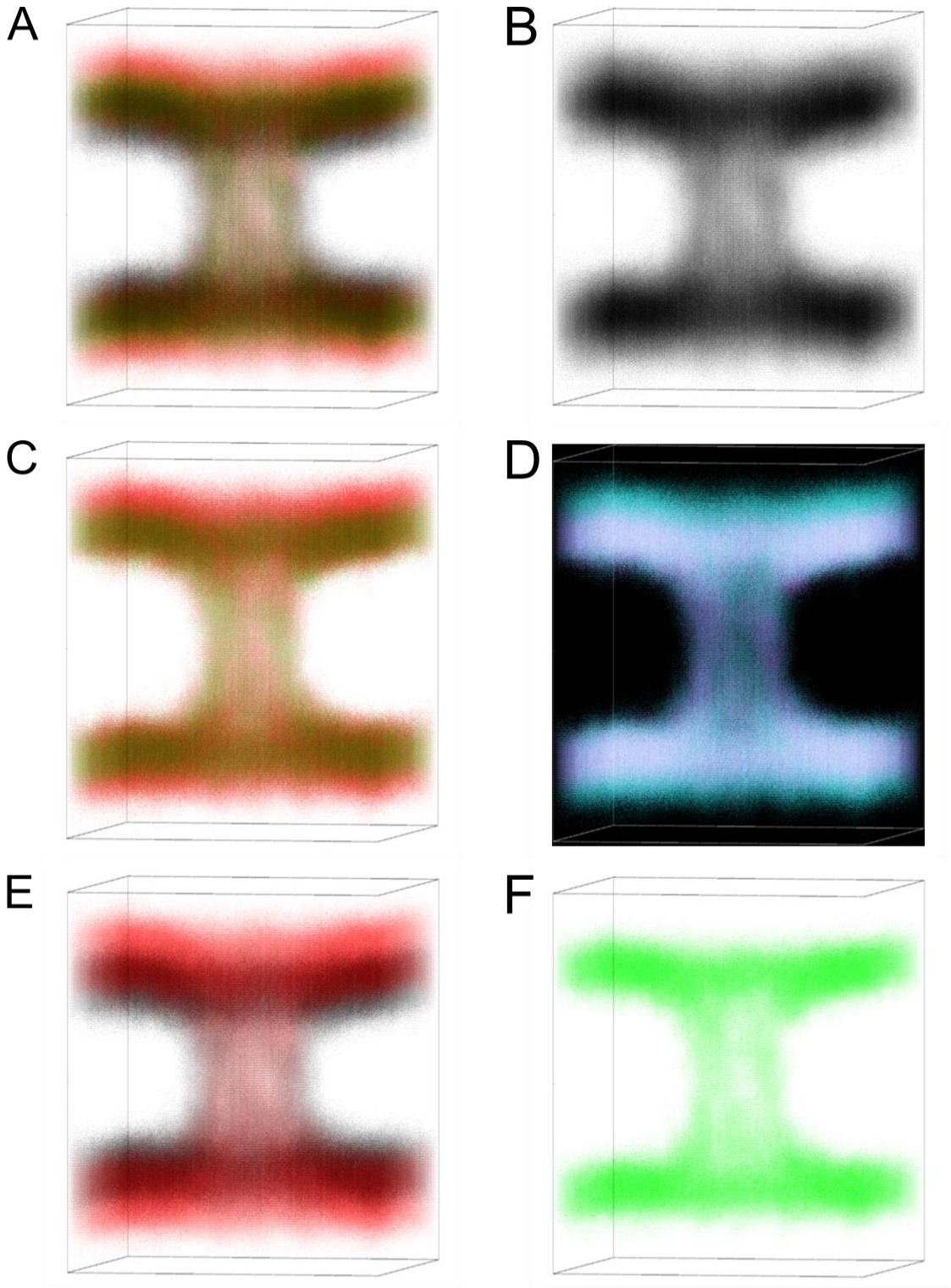


Fig. 26: *CompXRs at charge imbalance +/- 9, SCMTR's are associated with the pore. A-cholines, phosphates, esters; B-cholines, phosphates, esters; C-cholines, phosphates; D-cholines, phosphates; E-cholines, esters; F-phosphates.*
Colors: red-cholines; green-phosphates; black-esters.

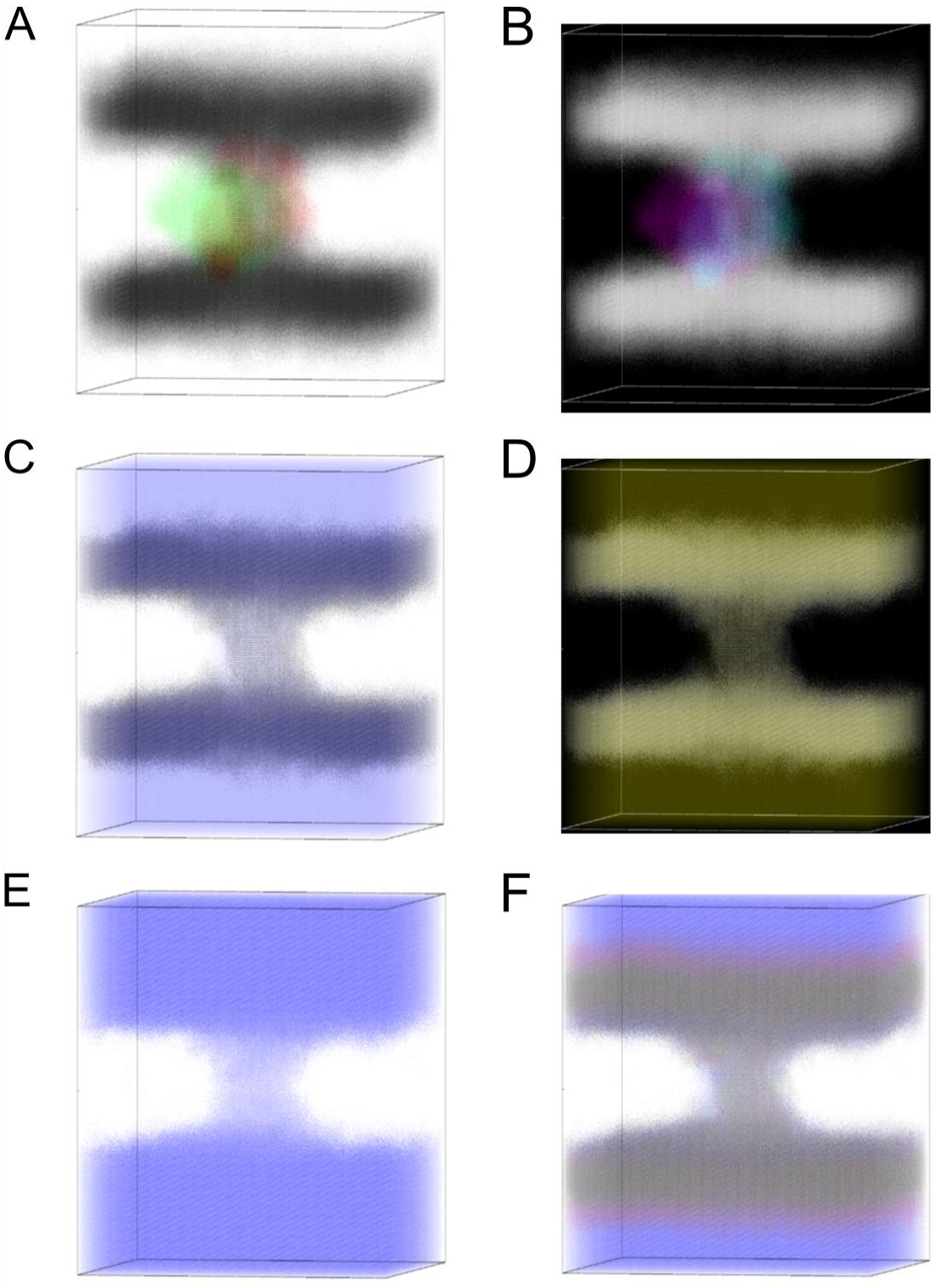


Fig. 27: *CompXRs at charge imbalance +/- 1, SCMTR's are associated with the pore. A-cholines, phosphates, esters, SCMTRs; B-cholines, phosphates, esters, SCMTRs; C-water, esters; D-water, esters; E-water; F-cholines, phosphates, esters, water. New Colors: red-SCMTR head; green-SCMTR tail; blue-water.*

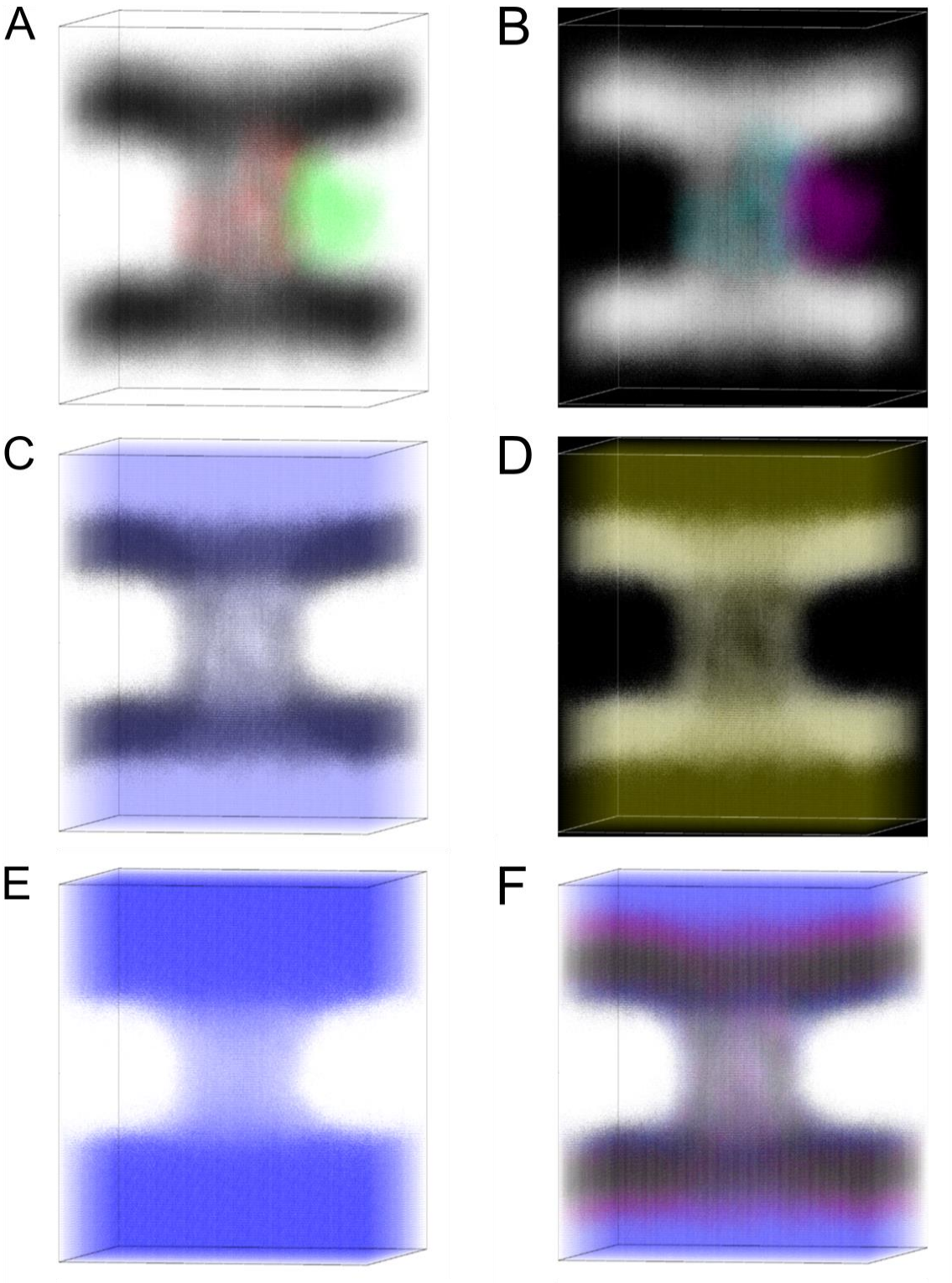


Fig. 28: *CompXRs at charge imbalance +/- 9, SCMTR's are associated with the pore. A-cholines, phosphates, esters, SCMTRs; B-cholines, phosphates, esters, SCMTRs; C-esters, water; D-esters, water; E-water; F-cholines, phosphates, esters, water. New Colors: red-SCMTR head; green-SCMTR tail; blue-water.*

As exciting as the previous images are, *gnuplot* was created for creating plots and graphs, not imaging biostructures. An *AWK* routine was developed to translate the data from a GROMACS format into a set of custom PDB structure files that could include the number density and either charge, field, or potential for every voxel. Because calculating number densities is computationally cheap, CompXRs can easily be produced even at resolutions of 0.5 Å. By rendering in VMD, it is now possible to make use of the calculated number densities to render mole composition isosurfaces. If the density fraction for a volume element for a sub-species like choline is 0.08, i.e.

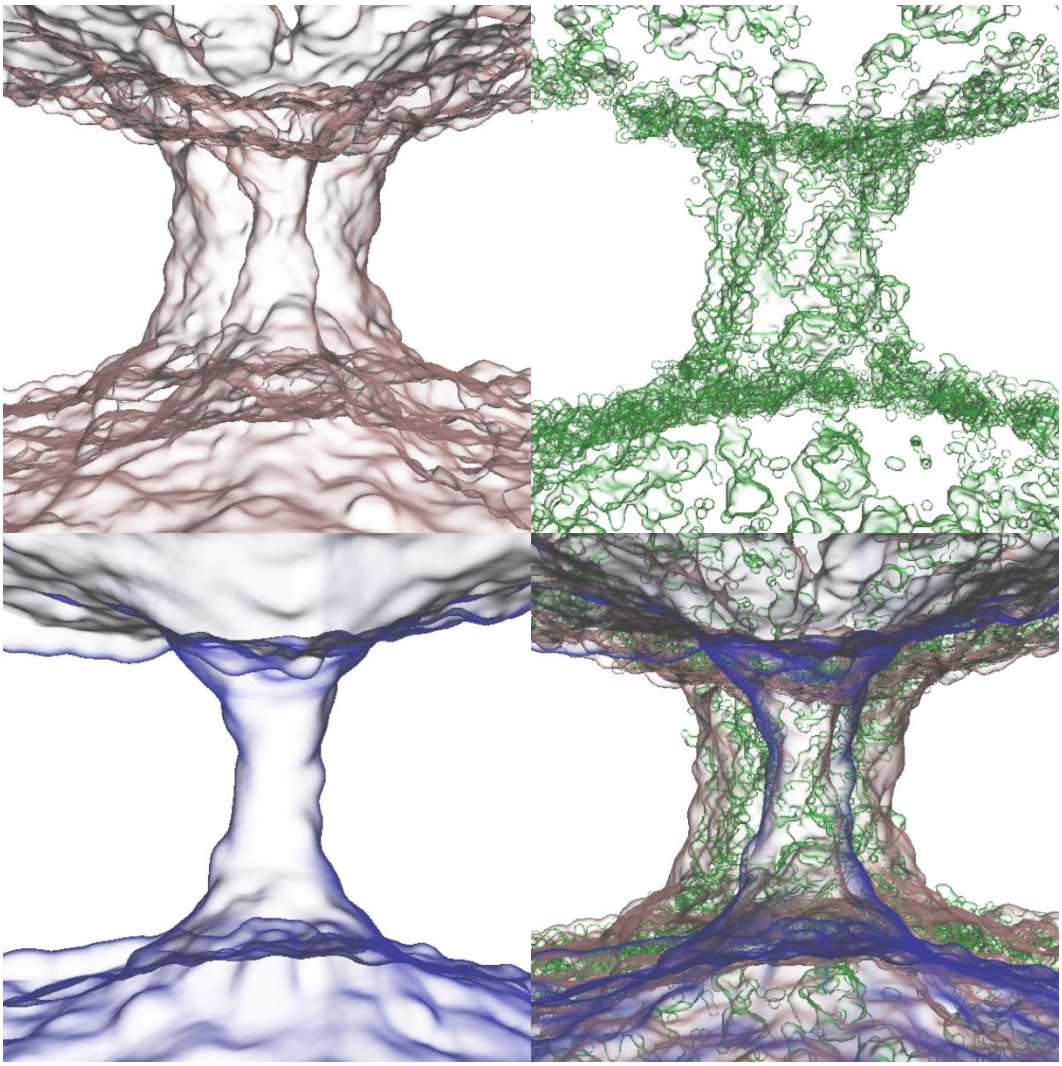


Fig. 29: *Computational X-ray of pore rendered in VMD. Red – Choline, Green – Phosphate, Blue – Water.*

the molar composition of choline at that position is 8%. The first two panels in **Fig. 29** represent the boundaries at which the molar composition of choline and phosphate are 8% or greater. Notice the difference in how the two moieties are distributed. These and other CompXRs illustrate exactly how the zwitterionic elements of DOPC lipids interact. Artist renditions of DOPC membranes that are typically used to illustrate membrane surfaces in papers do not come close to capturing this reality.

The blue surfaces in the third and fourth panels of **Fig. 29** represent water at 75% molar density, illustrating the throat of the pore. By scanning through density values more detailed information of the pore structure, in terms of conformation and composition, can be extracted. For example, it can quickly be demonstrated that the phosphate groups within DOPC lipid form clumps in a narrow band surrounded by a halo of the oppositely charged cholines.

Future Directions. CompXRs will aid the study of biostructure-electric field interactions. We envision creating a new plug-in module for VMD to add controls to fully exploit the CompXR datasets. The more technical details of creating CompXRs are introduced with an overview explanation in the Supporting Material for Section 2. All the relevant code and full documentation for implementing them have been published on GitHub along with many visualizations and animations.

SECTION 3

EFFECTS OF SALT, WATER, AND PROTEIN FORCE FIELDS ON PROTEIN FOLDING THERMODYNAMICS IN MOLECULAR SIMULATIONS

INTRODUCTION

The stability of a functional, folded protein depends upon a complex thermodynamic landscape wherein the protein interacts with itself, neighboring proteins, water, ligands, and other solutes. By modifying a protein's chemical environment, its thermodynamic stability can be greatly affected. For example, buffer solutions that include monoatomic and molecular ions can stabilize a protein's folded conformation, control protein sidechain protonation, and maintain protein solubility. Non-native solvents have

also been shown to affect protein stability. Depending on the nature of protein-solvent interactions, a protein's conformational stability and function can either be enhanced or diminished. Solvents that differ significantly from a protein's natural environment tend to destabilize the native folded conformation, thus impeding or eliminating function. However, non-native solvents will sometimes induce beneficial changes in enzyme activity and selectivity. In one of the most famous examples lipase from *Geotrichum candidum* has demonstrated the ability to selectively catalyze oleic, linoleic, and α-linolenic acid reactions at water-octane interfaces(74) as well as in the ionic liquids 1-butyl-3-methylimidazolium tetrafluoroborate ($[C_4mim][BF_4]$)6 or hexafluorophosphate ($[C_4mim][PF_6]$)12(75) as well as others(76). Other enzymes such as glycoside hydrolases have been demonstrated to maintain high levels of activity in the mixtures of ionic liquids and water(77) with relatively high ionic concentrations (~20% v/v). In fact, enzymes and other proteins are often modified through directed evolution(78, 79) to withstand conditions that would otherwise deactivate the wild-type. Generally, an increase in the enzyme's surface charge imparts higher thermal stability while also increasing tolerance of solvents of high ionic strength(80). It has been observed that organisms which are adapted for conditions of high temperature, high pressure, and chemically heterogeneous, such as those observed near volcanic vents in the ocean floor, tend to have enzymes containing high surface charge(79, 81). In addition to conformational stability, a protein's colloidal stability is affected by the ions contained in the solution. At excessively high salt concentrations, proteins precipitate due to competition between the protein's charged surface residues and salts in binding water molecules(82). Thus, hydrophobic interactions

among neighboring proteins begin to overcome the electrostatic repulsive forces that typically separate them.

Traditionally, the effects of salt species and concentration on protein conformational stability have been predicted by correlative equations and heuristics, which are computationally efficient but often lack the specificity or transferability that would make them intriguing design tools. We hypothesize that ions, and particularly molecular ions (e.g. sulfate, nitrate, phosphate, choline), should be modeled similarly to neutral ligands (e.g. drugs, substrates, carbohydrates) for which specific molecular interaction models (e.g. MD) are often employed. Perhaps it is even more crucial to consider molecular-level interactions when dealing with ions rather than neutral species due to coulombic interactions being much stronger than van der Waals interactions in almost all cases. With increases in computational power, we envision MD becoming a powerful tool not only for the rationalization of experimental results but also for the molecular-level design of new protein-solvent systems that enhance protein function and stability.

While salts mediate the interaction of proteins with water, it is difficult to experimentally observe molecular level interactions of proteins with water or salt. X-ray crystallography is one technique that has proven indispensable for determining the location of tightly bound water molecules, ions, and ligands. However, the crystal environment and cryogenic temperatures required for crystallography do not accurately represent the solvent environment where room-temperature experiments are performed, and biological function is observed. Moreover, crystallization typically occurs only under a narrow range of experimental conditions. Often, crystallographers find few if any solvents in which large, high-quality crystals will grow. These limitations hamper the study of protein-ion

interactions. Nonetheless, structures deposited in the Protein Databank(83) provide initial clues as to the locations of ion binding sites and the strength of the competition between ions and water to occupy a protein's surface.

To gain a better understanding of the effects of protein-water-ion interactions on the conformational stability of proteins, molecular dynamics (MD) simulations can provide a physical interaction model that is not constrained by experimental considerations, assuming an initial experimental protein structure or high-quality homology model is available. Motivated by a desire to understand how non-native (particularly highly ionic) solvents modulate protein conformations and function, Dr. Jaeger and others have studied the effects of ionic liquids on several proteins and enzymes(84, 85). In most cases, the observed behavior from their simulations has been directly compared to experiment to validate models. However, only a few simulations have attempted to theoretically predict protein folding, thermodynamics in ionic liquids and high-salt environments using MD simulations(86). The usefulness of an MD simulation is tied to the quality of the description of interatomic potential energies. A self-consistent set of these interaction potentials is known as a force field (FF). Due to limitations in computational power, most previous studies have used only one FF combination. In Dr. Jaeger's previous work, proteins were most often treated with a variant of the AMBER99SB FF(87), water was modeled with the TIP3P model(88), ions used the standard AMBER parameters, and other ligands or molecular ions used the General AMBER FF (GAFF)(89). We hypothesize that changes in the FF will drastically affect predictions of folding thermodynamics and the stability of specific conformations. While there have been numerous studies about how various protein and water FFs affect protein structure, to the best of our knowledge, there has been no

systematic study of the effects of FF choice on folding thermodynamics with respect to salt species and concentration. With recent advances in computing power and simulation methods, namely parallel bias metadynamics (PBMetaD)(90), a systematic study of these effects has now become feasible.

Metadynamics (MetaD) is an enhanced sampling method used in molecular simulations to intelligently drive the system toward underexplored regions of conformational space. During an MD simulation, a time dependent bias is added to certain low-dimensional descriptors of the system conformation known as a collective variable (CV). The bias takes the form of the sum of Gaussian kernels deposited at specific time intervals, and the bias evolves to push the system away from previously explored states until all relevant energy barriers are overcome. The dimensionality of the Gaussian kernels in traditional MetaD(91, 92) has been equal to the number of the CVs set by the researcher. Due to issues with convergence and computational complexity, MetaD was limited to two to three CVs in practical application. In many cases, these CVs were insufficient to overcome all relevant energy barriers, and convergence continued to be a problem. PBMetaD is an evolution of the MetaD protocol which overcomes this obstacle. Specifically, PBMetaD allows the efficient sampling of many CVs at once by a clever reweighting scheme which allows the deposition of one-dimensional Gaussians. After the simulation has sufficiently explored all CV space, the free energy the system can be estimated as a function of the biasing potential built throughout the MetaD simulation. The full details of the PBMetaD protocol and its implementation are presented in the Supporting Material.

In simplest terms, consider the free energy of a thermodynamic system with m CVs. Collective variables are not strictly independent variables, each is affected to a degree by any conformation change in the system. Likewise, changes to the bias potential of one, affects the CV value of each of the others to various degrees; they are interdependent. (1) Without prior knowledge of these degrees of interdependence, the need for such can be eliminated by noting that the energy contributions of their collective sum must equal the total energy change of the system during the simulation. To exploit this thermodynamic reality, the expression for the history-dependent bias potential $V_G(S,t)$ (eq. 1) for well

$$V_G(S_1,t) = k_B\Delta T_1 log\left(1 + \frac{\omega_1 N(S_1,t)}{k_B\Delta T_1}\right)$$

tempered MetaD(93) is discretized with a probability based weighting factor η for the individual CVs. For m CVs this generates a set of m equations…

$$V_G(S_1,t) = k_B\Delta T_1 log\left(1 + \frac{\omega_1 N(S_1,\eta = (1,0,\dots,0)_{m\ terms},t)}{k_B\Delta T_1}\right)$$

$$V_G(S_2,t) = k_B\Delta T_2 log\left(1 + \frac{\omega_2 N(S_2,\eta = (0,1,\dots,0)_{m\ terms},t)}{k_B\Delta T_2}\right)$$

$$\vdots \qquad (2)$$

$$V_G(S_m,t) = k_B\Delta T_m log\left(1 + \frac{\omega_m N(S_m,\eta = (0,0,\dots,1)_{m\ terms},t)}{k_B\Delta T_m}\right)$$

where $\boldsymbol{S}$ is a function of the microscopic coordinates $\boldsymbol{R}$ of the system $\boldsymbol{S}(\boldsymbol{R}) = (\boldsymbol{S_1}(\boldsymbol{R}),\dots,\boldsymbol{S_m}(\boldsymbol{R}))$ for $\boldsymbol{m}$ collective variables, $\boldsymbol{k_B}$ is the Boltzmann constant, $\boldsymbol{T}$ is the thermodynamic temperature and $\boldsymbol{N}$ is the Monte Carlo based discretizing function, which determines which of its $\boldsymbol{m}$ terms will be non-zero for each simulation step. When term 1 of the discretizing function is 1, term 1 of the bias potential is updated. When term $\boldsymbol{m}$ of

the discretizing function is 1, term $\boldsymbol{m}$ of the bias potential is updated. When a bias potential term is updated (increased) the size of the increase is set by the current deposition rate $\boldsymbol{\omega_t}$…

$$\omega_t = \omega\, exp\left(-\frac{V_G(S,t)}{k_B \Delta T}\right)$$

where ω is the initial bias potential energy deposition rate. However, this math lacks a feedback component to bias it towards convergence because the CVs are affecting the bias potential independently of one another; it models reality poorly.

Fortunately, since the probability of observing any set of microscopic coordinates R of the system is equal to the negative exponential of the system energy divided by $k_B T$, this thermodynamic reality can be exploited. To do so, note the conditional probability $P(\eta|R)$ of observing a certain value of η given a configuration R is simply…

$$P(\eta|R) = \frac{P(R,\eta)}{P(R)} \tag{4}$$

and these probabilities are equal to negative exponentials of known energies, the potential bias of an individual CV and the collective sum of their potential biases. Defining $k_B T$ in terms of β, these probabilities can be written as…

$$\beta = 1/k_B T \tag{5}$$

$$\begin{aligned} P(\eta = (1,0,\dots,0)_{m\ terms}|R) &= \frac{\exp\left(-\beta V_G(S_1,t)\right)}{\exp\left(-\beta V_G(S_1,t)\right) + \cdots + \exp\left(-\beta V_G(S_m,t)\right)} \\ &\vdots \\ P(\eta = (0,0,\dots,1)_{m\ terms}|R) &= \frac{\exp\left(-\beta V_G(S_m,t)\right)}{\exp\left(-\beta V_G(S_1,t)\right) + \cdots + \exp\left(-\beta V_G(S_m,t)\right)} \end{aligned} \tag{6}$$

In the limit $\lim_{t\to\infty} \omega(t) = 0$ (model convergence), probabilities become weighting factors. By weighting the bias energy deposition equation (3) with these probabilities, the independent CV discretizing function N in equation(s) (2) can be replaced with a continuous collective variable η, the sum of the individual CV probabilities. Because the sum of these probabilities is known to be equal to one, the mathematical expressions drop out from the equations. First, equation (3) is expanded to incorporate the m probabilities…

$$\begin{aligned}
\omega_1(t) &= \omega_1\, exp\left(-\frac{V_G(S_1,t)}{k_B\Delta T_1}\right) P(\eta = (1,0,\dots,0)_{m\ terms}|R) \\
\omega_2(t) &= \omega_2\, exp\left(-\frac{V_G(S_2,t)}{k_B\Delta T_2}\right) P(\eta = (0,1,\dots,0)_{m\ terms}|R) \\
&\ \ \vdots \\
\omega_m(t) &= \omega_m\, exp\left(-\frac{V_G(m,t)}{k_B\Delta T_m}\right) P(\eta = (0,0,\dots,1)_{m\ terms}|R)
\end{aligned} \tag{7}$$

Expressing these into the original set of discretized equations completes the history-dependent potential-bias term for PBMetaD…

$$\begin{aligned}
V_G(S_1,t) &= k_B\Delta T_1 log(1+\omega_1(t)) \\
V_G(S_2,t) &= k_B\Delta T_2 log(1+\omega_2(t)) \\
&\ \ \vdots \\
V_G(S_m,t) &= k_B\Delta T_m log(1+\omega_m(t))
\end{aligned} \tag{8}$$

METHODS

Simple monoatomic ions were selected as test cases. The two salts (NaCl and KCl in the AMBER and Joung FFs) were mixed with three protein FFs (AMBER99SB*-ILDN(94), CHARMM27(95), and CHARMM36(51)), and two water models (TIP3P and SPC/E)(88, 96) across four salt concentrations (0 mM, 150 mM, 300 mM, 1000 mM). Two common peptides, (residues 41-56) (GB1: aka Hairpin) of the immunoglobulin binding

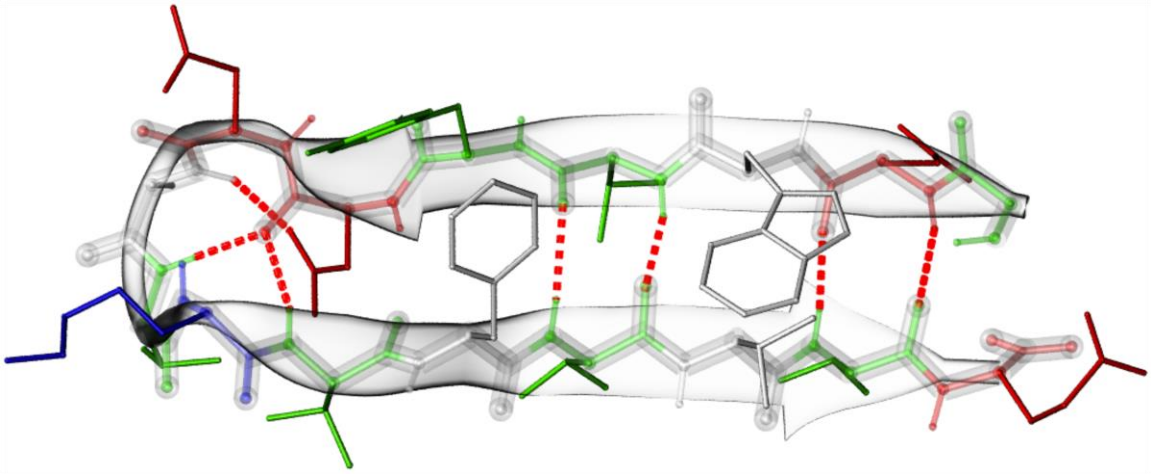
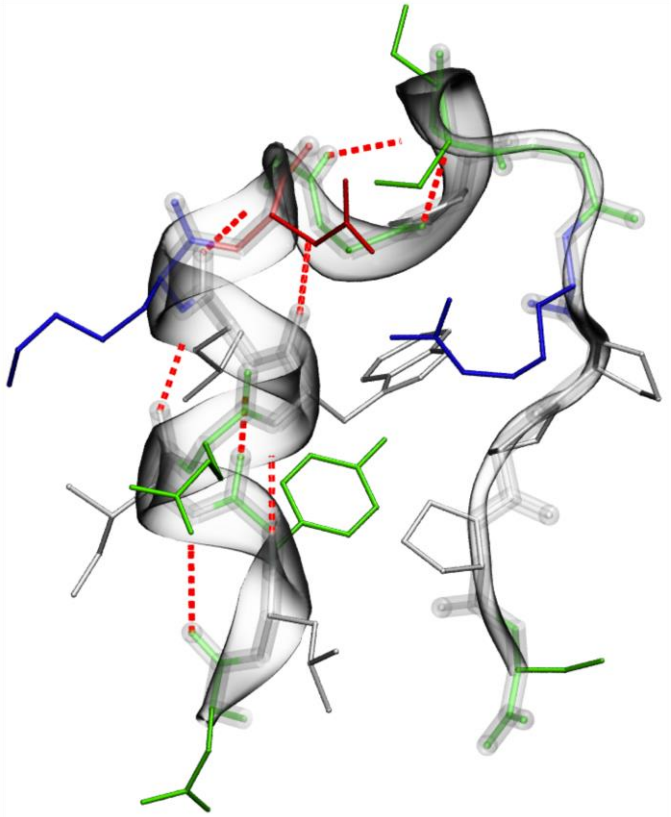


Fig. 30: *GB1 "Hairpin" peptide sequence. It reverses the direction of the peptide backbone.*

domain of the streptococcal protein (GB1)(1GB1.pdb)(97) and the 20-residue tryptophan-cage protein (TRPC)(1L2Y.pdb)(98) were taken from the RCSB Protein Databank(83) . **Figs. 30** and **31** show conformations of the natively folded peptides. For both systems 6039 water molecules were added to a dodecahedron box type and the potential energy of the system was minimized for up to 50,000 steps using the steepest descent gradient method before adding ions. TRPC has a natural charge of +1q and GB1 has a natural charge of -1q at neutral pH. The TRPC simulations were made charge neutral by the addition of one Cl^- ion. Depending on the salt, the Hairpin simulations were made charge neutral by the

addition of one Na^+ or one K^+ ion. Salts (NaCl or KCl) were added with the GROMACS *genion* utility to create simulations with 150mM, 300mM and 1M salinities.



MetaD simulations were facilitated using the PLUMED2(99-101) plugin for GROMACS2018.2(55, 58-62). Four CVs were hypothesized to be key in the characterization of the folding transition. These four CVs were, (a) the number of native hydrogen bond contacts from the protein's native secondary structure, (b) the radius of gyration of the

Fig. 31: *TRPC mini protein. It contains a short section of an α-helix.*

protein alpha carbons, (c) the number of contacts between the distal sidechain carbon in the hydrophobic residues (GLY, ALA, VAL, LEU, ILE, PRO, PHE, TRP), and (d) the number of native alpha carbon contacts. Native contacts were defined by analyzing the alpha carbon positions in the experimental native structures and identifying any non-nearest neighbors within 5.47 Å. To ensure that the CVs involving the quantification of contacts were continuous and differentiable rather than discrete (which is a requirement of MetaD), a sigmoidal function was used to quantify partial contacts (hydrogen bonds, native and hydrophobic) as in Equation (8). For hydrogen bonds, n = 6, m = 12, and r_0 = 0.3, for native contacts, n = 20, m = 40, r_0 = 0.6, for hydrophobic, n = 6, m = 8, r_0 = 0.6. For

$$S_{ij} = \frac{1 - \left(\frac{r_{ij} d_0}{r_0}\right)^n}{1 - \left(\frac{r_{ij} d_0}{r_0}\right)^m} \tag{8}$$

guidance on the choice of the alpha carbon separation for native contacts, scripts for identifying and grouping alpha carbon neighbors, and the full PLUMED parameters used for the simulations see the Supporting Material. **Fig. 32** shows the complete simulation parameter space and CV's thus simulated.

Simulations Parameter Space		
Force Fields	AMBER99sb*-ILDN	
	CHARMM27	CHARMM36
Water Models	TIP3P	SPCE
Ion Models	AMBER	Joung
Peptides	GB1	(1GB1.pdb)
	TRPC	(1L2Y.pdb)
Salts	NaCl	KCl
Molarities	0.0 M, 150mM, 300mM, 1.0 M	
Collective Variables		Hydrogen bonds
	Native contacts	Hydrophobic contacts
	Radius of Gyration	

Fig 32: *Force fields, peptides, salts, concentrations, and CVs explored by simulations.*

To explore the free energy landscape of these collective variables, each simulation was run for a total of 5 μs. With virtual sites for the hydrogen atoms, a 5 fs timestep was used requiring a total of 1×10^9 integration steps. The Verlet cutoff-scheme was used with a van der Waals cut-off of 1.0 nm and a potential switch modifier beginning at 0.8 nm. Electrostatic interactions were calculated explicitly at distances shorter than 1.0 nm and with the particle mesh Ewald summation method at longer distances. One temperature coupling group controlled the simulation temperature with the stochastic velocity rescaling algorithm at a temperature of 310 K. Constraints were implemented on all bonds using the LINCS algorithm. Random initial velocities were assigned to all atoms at the start of each simulation. For a full listing of the MD parameters used in the simulation, see the Supporting Material.

Free energy surfaces were generated from the simulation data. Changes in Helmholtz free energy (since the simulations used the NVT ensemble) associated with peptide folding and unfolding were quantified. Convergence of these simulations was assessed by tracking changes to the free energy surfaces over the last 20% (1 μs) of the simulation. A simulation is considered well-converged by one of two benchmarks; a standard deviation of the folding free energies at 5 μs and the preceding five, 200 ns spaced intervals being no more than 1 kcal/mol, or; the ΔΔG during the final 200 ns is less than 20% of the standard deviation. When a collective variable has converged, the weighted cumulative total in these file instances ceases to change. **Fig. 33** shows how this total converges (ceases to meaningfully change) for a CV free energy profile over the last 24% of a simulation. High free-energy states tend to converge more slowly than low-free energy states, as can be seen on the left side of **Fig. 33**. Individual details of all the simulation convergences with discussion are included in the Supporting Material. It should be noted that the experimental folding free energy of TRPC is -0.67±0.1 kcal/mol(102).

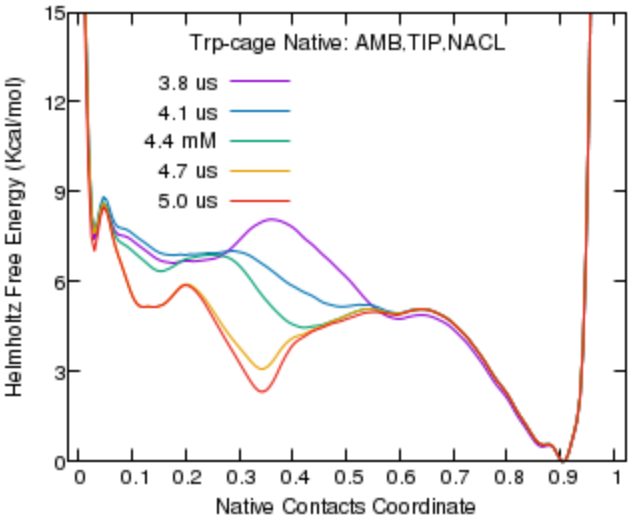


Fig. 33: *Evolution of convergence of CV free energy profile during ~last fifth of simulation. Note that by 3.8 μs, the deepest well has been fully explored.*

Free energy diagrams were normalized so that the lowest reported free energy is always zero. This makes comparing free energy surfaces at different solute concentrations easier. This normalization does not impact conclusions, because relative changes in free energy of folding/unfolding are the important observables. The left panel of **Fig. 36** shows

an example of a free energy surface plot for the TRPC peptide native contacts simulated with the CHARMM36 FF, TIP3P water and NaCl salt modeled with Joung ions, and each trace has a minimum value of zero. With this data, the change in free energy associated with folding/unfolding can be calculated from the Boltzmann factor, $e^{-\frac{A}{RT}}$, which equals the relative probability of the arrangement that has the energy A. At 310 K, RT is about 2.58 kJ/mol. TRPC contained 36 native contacts. The fraction of native contacts (Q) was used as the metric by which a protein is defined as being folded or unfolded. The boundary between unfolded and folded, is thus defined as Q = 0.5, and the Boltzmann weighted average of the free energies above and below this cutoff were compared to find the folding free energy. The ratio of the relative probabilities of these states is the equilibrium constant of the system. So, the change in free energy associated with folding/unfolding is simply $-RTlnK_{eq}$. An *AWK* program was written to automate these calculations (see Supporting Material). These values are shown in **Fig. 36** according to the convention of regarding the unfolded state as the initial state.

The radial distribution function between charged residues and ions were generated using the *rdf* utility in GROMACS for folded and unfolded states. Lysine (LYS) and aspartic acid (ASP) were present in both peptides. TRPC also included arginine (ARG), and GB1 included glutamic acid (GLU). For the negatively charged sidechains, ASP and GLU, interactions with sodium and potassium ions were analyzed. For the positively charged sidechains, LYS and ARG, interactions with chlorine ions were analyzed.

RESULTS AND DISCUSSION

An assessment of AMBER ions was also planned for this study. It has been observed in previous studies and in our early work on this subject that for sodium and potassium and chlorine, AMBER ions do not replicate real solubility behavior, and precipitation occurs at much lower than expected concentrations. AMBER ions were originally parameterized to accurately represent solvation free energies. Yet, there are two adjustable parameters for the Lennard-Jones interactions of ions, namely ε and σ which dictate the depth, width, and location of the attractive well. There are therefore many sets of ε and σ that give accurate solvation energies but inaccurate precipitation behavior. Joung and Cheatham recognized this fact and added additional restraints in their search for more accurate ε and σ values. In their study, they generated ε and σ values for several monovalent ions in several water FFs. In summary, we could not simulate concentrations of 300 mM and above with AMBER ions, and we removed them from planned simulations.

For PBMetaD, convergence was assessed by comparing changes in the folding free energies as indicated by the native contacts CV. Six free energies of each simulation were determined for the final 1 μs of the simulation (4.0 μs to 5.0 μs at 200 ns intervals). For TRPC, 29 of 32 simulations achieved this benchmark, and for GB1 the figures were 27 of 32 simulations as summarized in **Figs. 34** and **35**.

This analysis revealed that three of the GB1 simulations had standard deviations in bins isolated from the group cluster. Two of these also had final $\Delta\Delta A_{folding}$ energies greater

than 20% of their standard deviations, not achieving our defined benchmark.

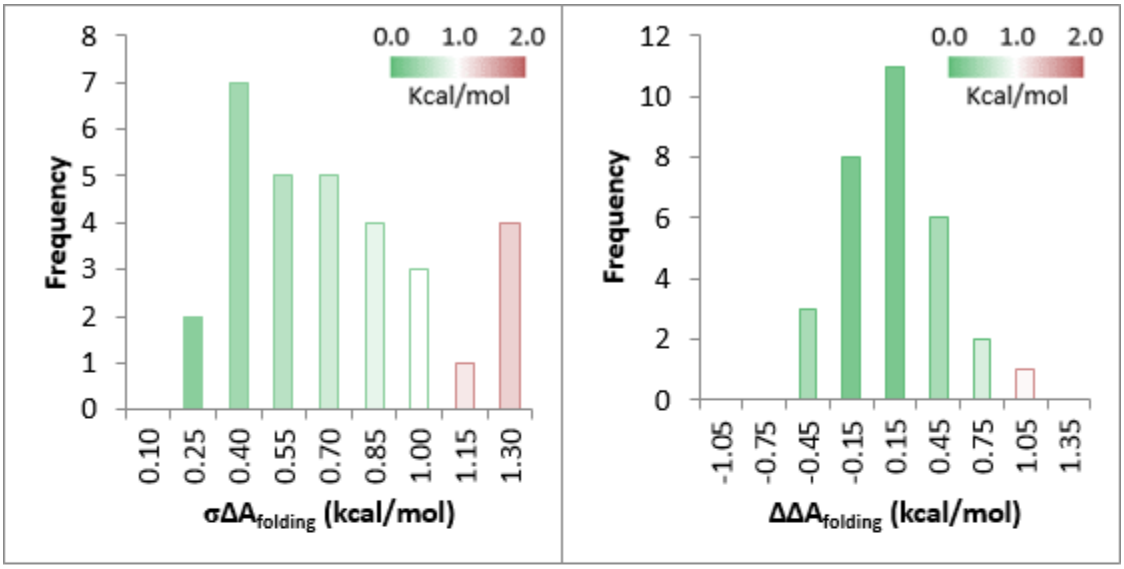


Fig. 34: *Simulation convergence benchmarks for TRPC. Standard deviation is σ. Twenty-nine of the 32 simulations achieved both or at least one of these benchmarks.*

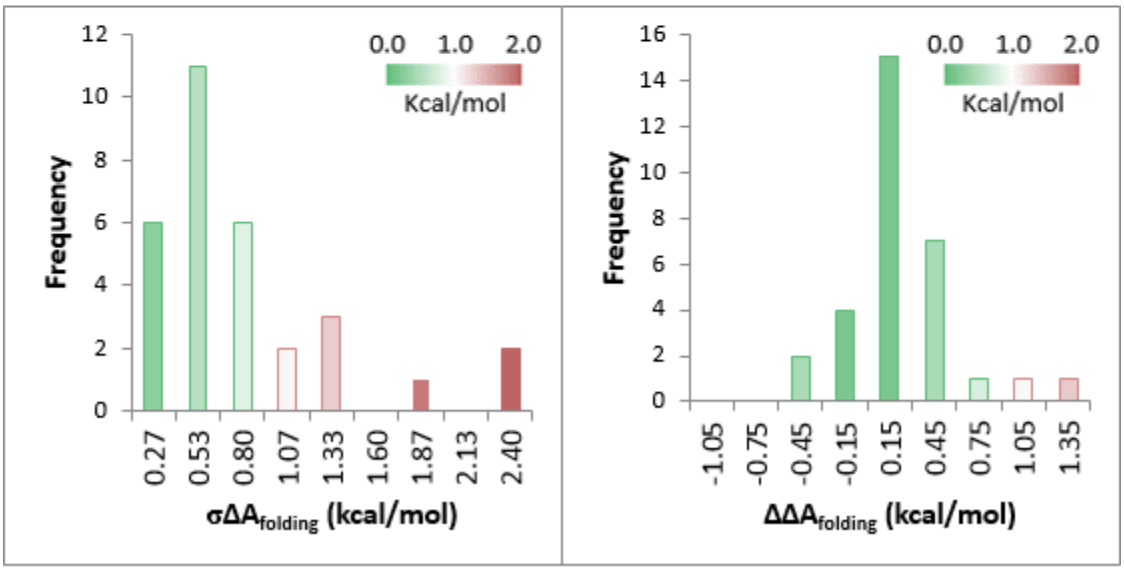


Fig 35. *Simulation convergence benchmarks for GB1. Standard deviation is σ. Twenty-seven of 32 simulation achieve both or at least one of these benchmarks.*

Originally only 2 μs trajectories were collected. After some very preliminary analysis, the convergence of the simulations in general was deemed inadequate. This prompted an extension of all the calculations by another 3 μs to a total of 5 μs.

For each set of FFs and salt type, the free energy profiles for each of the four salt concentrations were plotted together with respect to their normalized CV coordinate. The two panels in **Fig. 36** contrast the substantial differences of results between the CHARMM36 and CHARMM27 FFs. In this example we observe the completely

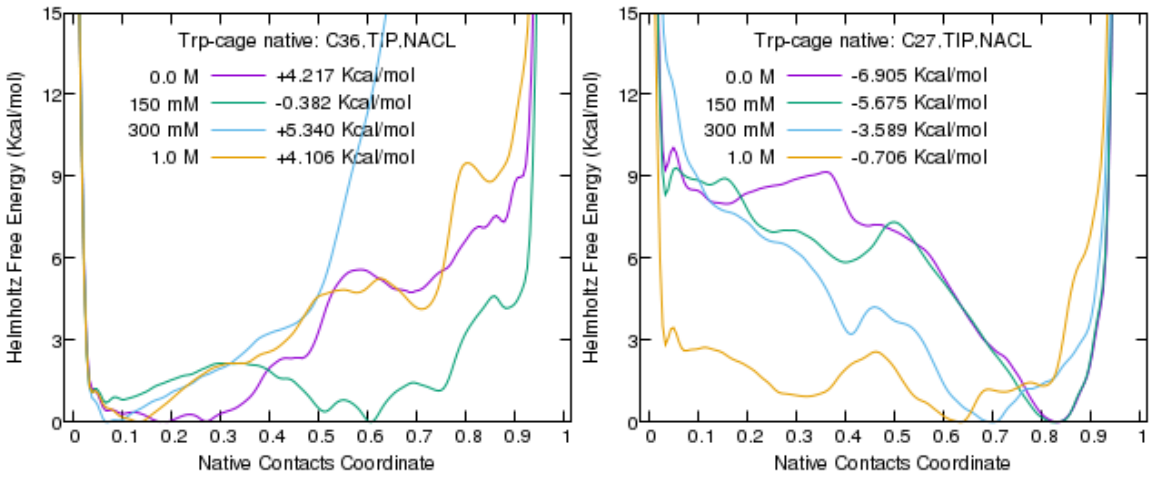
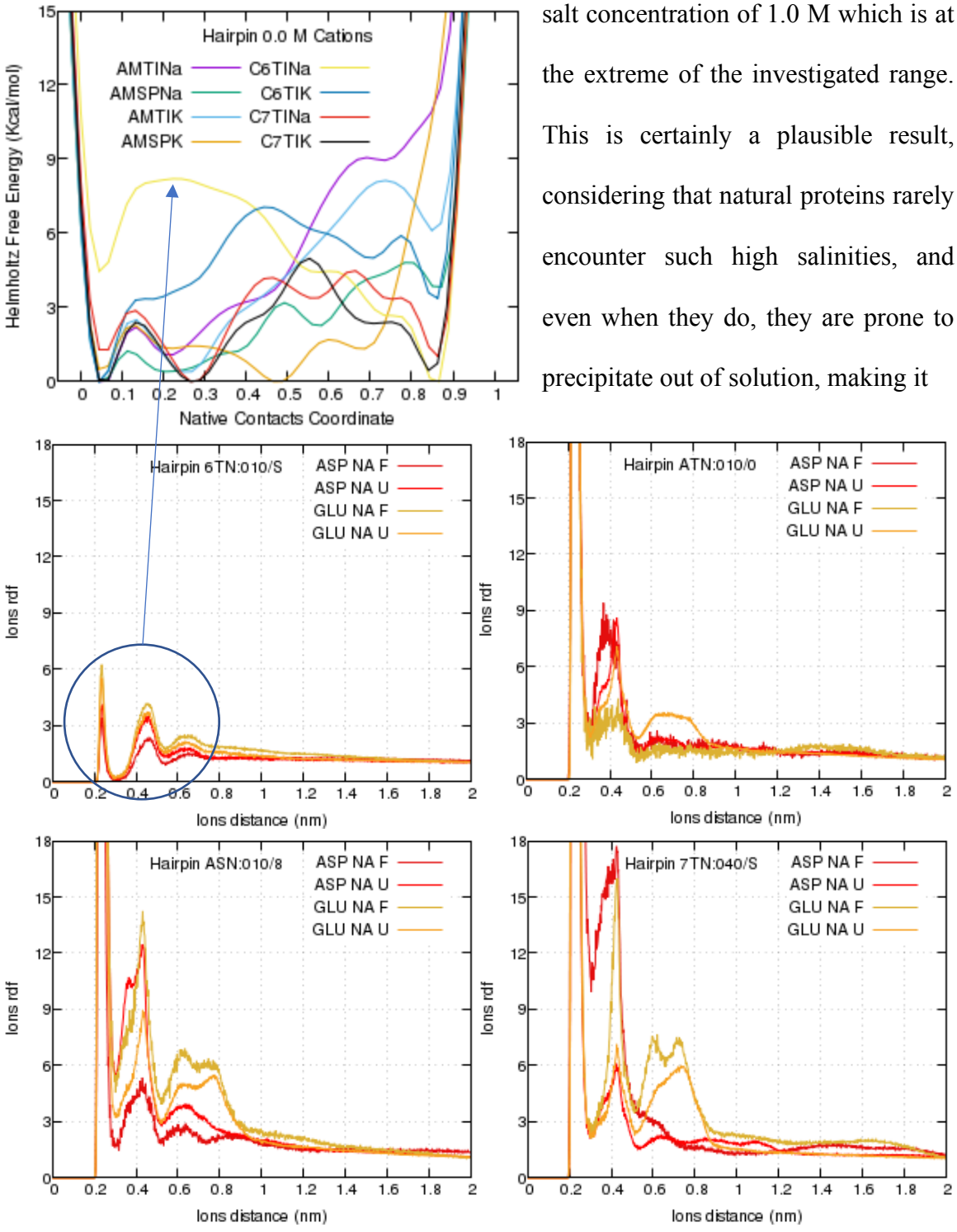


Fig. 36: *Comparison of free energy profiles produced by two different CHARMM force fields.*

opposite skews of the energy profiles modeled by the two different protein FFs, even though the FFs are derived from the same family. One FF is clearly indicating stabilization of unfolded states, and the other indicating the opposite. Also note the outlier character of the 300 mM trace in CHARMM36. Since TRPC is experimentally known to prefer the folded state at physiological salt concentrations, these two plots demonstrate clearly that CHARMM27 would be the preferred FF when evaluating the effects of salinity on peptide conformation stability. The one surface with a different character for CHARMM27 is at a



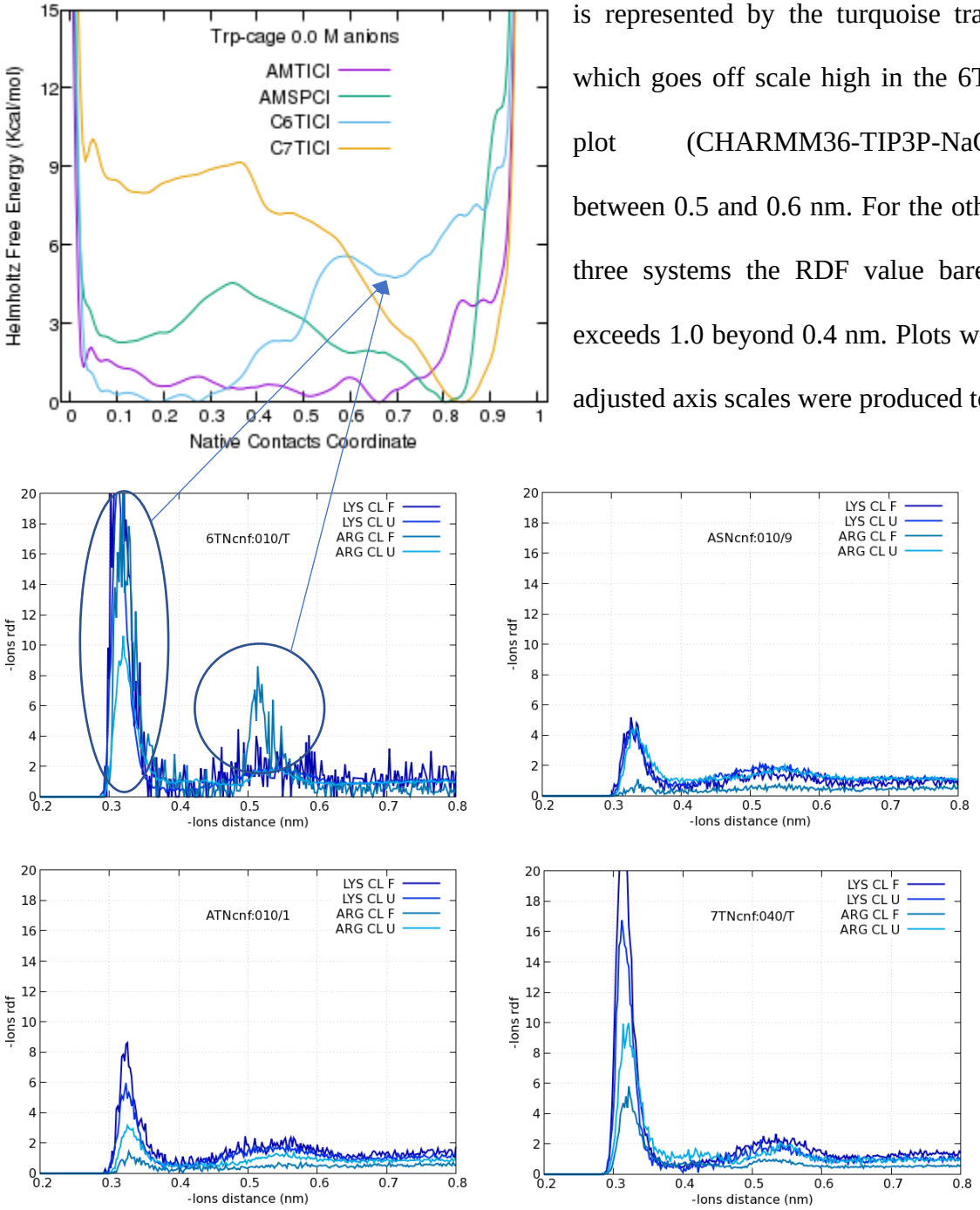
salt concentration of 1.0 M which is at the extreme of the investigated range. This is certainly a plausible result, considering that natural proteins rarely encounter such high salinities, and even when they do, they are prone to precipitate out of solution, making it

Fig. 37: *Comparison of outlying CHARMM36 FES profile with RDF plot which reveals significantly different interaction between the cation and protein as well.*

difficult to experimentally disentangle the effects of salting-out versus unfolding.

With five primary parameters to consider in the free energy plots, **Fig. 36** stood out while other discrepancies are less obvious. For instance, recall GB1 has a net charge of -1q at physiological pH, therefore even at 0.0 M concentration, one cation of sodium or potassium must be present to neutralize charge. By comparing the GB1 native contact free energy profiles at 0.0 M, in seven of the eight systems an energy preference towards the unfolded state is apparent. This preference is observed in **Fig. 37** in which AMBER (AM) CHARMM36 (C6) and CHARM27 (C7) FFs with sodium and potassium ions using TIP3P(TI) and SPC/E (SP) water models. Is this slight energy preference in any way attributable to the presence of the cations? No simulations were performed on GB1 without the presence of cations. However, by examining the radial distribution functions (RDF) of the four systems with a sodium cation, another clue appears. The one outlier of the folding free energy profiles occurs with the CHARMM36 FF, TIP3P water, and Na ion, and the RDF of this system shows a distinct difference of very minor association with GB1's negatively charged sidechains compared to the others.

Notice the outliers in our first two observations have the CHARMM36 FF in common. Interestingly, examining anionic interactions with 0.0 M (i.e. with one charge-neutralizing anion) on TRPC at physiological pH reveals a similar anomaly. For the non-CHARMM36 simulations in **Fig. 38**, all the folding free energies are negative demonstrating a slight preference for the folded state. For the CHARMM36 profile the demonstrating a folding free energy is +4.22 kcal/mol. Once again a distinct difference exists in the RDF plot for this system. From the RDF plot it is apparent that chlorine is involved in a much stronger interaction with arginine in the folded state. This interaction



is represented by the turquoise trace which goes off scale high in the 6TN plot (CHARMM36-TIP3P-NaCl) between 0.5 and 0.6 nm. For the other three systems the RDF value barely exceeds 1.0 beyond 0.4 nm. Plots with adjusted axis scales were produced to

Fig. 38: *Comparison of outlying CHARMM36 FES profile with RDF plot which reveals significantly different interaction between the anion and protein as well.*

make the trace colors more distinguishable to highlight the difference produced by CHARMM36.

CONCLUSIONS

Based upon our findings, we can recommend the following good practices for simulations that probe protein-ion interactions and their effects on protein conformational stability:

1. Use AMBER99SB*-ildn and CHARMM27 FFs rather than CHARMM36 which tends to over-stabilize the disordered states of the protein.

2. Use Joung ions rather than native AMBER ions, because AMBER ions tend to aggregate at concentrations higher than 150mM.

3. Use the TIP3P water model before using SPC/E, while the latter is not an entirely unreasonable choice.

4. PBMetaD is an effective way to simultaneously bias many CVs, and it is useful for quickly assessing the folding stability of a protein for which there are many hidden energy barriers.

With this information, we have developed plans to model the stability of small therapeutic proteins in various solution formulations. Since protein unfolding often precedes aggregation and precipitation, the stabilization of a protein's native folded structure is important to maintaining the quality of a protein solution. This future research will aim to find new chemical additives (and mixtures thereof) to stabilize the long-term storage of steroids, hormones, and enzymes such that they have a longer shelf life for hospitals, patients, and researchers.

SECTION 4

NUCLEOTIDE DEPENDENT STRUCTURAL TRANSITIONS AND DYNAMICS IN NITROGENASE ENZYMES

INTRODUCTION

Nitrogenase is the enzyme that allows bacteria and archaea to convert atmospheric dinitrogen into biologically available ammonia. Before the advent of industrial processes (i.e. Haber-Bosch) to produce ammonia-based fertilizers, nearly all nitrogen entering the biological nitrogen cycle originated in a nitrogenase enzyme. Even today, it is estimated that nearly half of nitrogen fixation in the world occurs by natural rather than synthetic means. The industrial production of ammonia via the Haber-Bosch process consumes about 3-5% of natural gas produced annually worldwide(103). Breaking the triple bond within dinitrogen is energetically expensive, and the process takes place at extraordinarily high

temperatures and pressures (773 K and 200 bar) in the presence of hydrogen gas over an iron catalyst. However, iron catalysts do not sufficiently accelerate reaction kinetics. Therefore, a synthetic catalyst to reduce the activation energy of nitrogen fixation has been widely sought after for decades. Recently, with the advent of improved experimental and theoretical methods to study enzymes at the electronic, atomic, and molecular levels, there has been a surge of research into nitrogenase as a model for a synthetic catalyst.

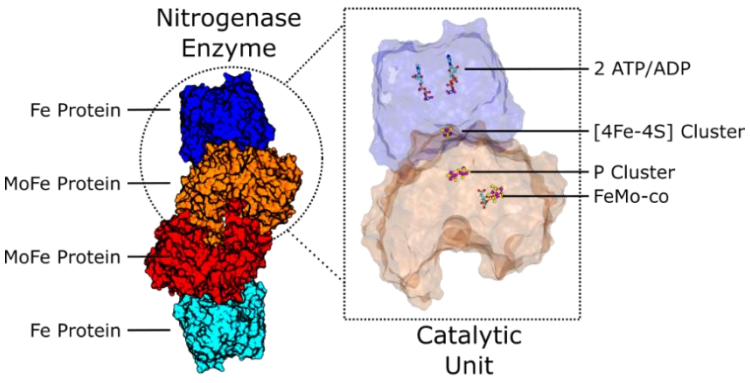


Fig. 39: *Nitrogenase complex and location of organometallic CFs and nucleotides in each of the two catalytic units. Graphic by: VJ.*

The nitrogenase complex consists of two components as can be seen in **Fig. 39**. First, is the heterotetrametric MoFe protein which contains a specialized FeMo cofactor (CF), as in **Fig. 40**, where dinitrogen is reduced to ammonia according to the reaction: $\mathbf{N_2 + 8\ H^+ + 8\ e^- + 16\ MgATP \rightarrow 2\ NH_3 + H_2 + 16\ MgADP + 16\ P_i}$, where P_i is a phosphate ion. In the reaction, eight electrons are pumped to the central catalytic CF through the second main component, the Fe protein, also known as the

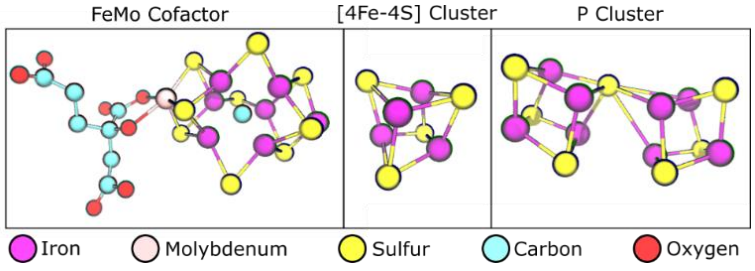


Fig. 40: *Structure and composition of the three organometallic CFs present in nitrogenase. Graphic by: VJ.*

dinitrogenase reductase. An individual Fe protein is a homodimer with two binding pockets for the nucleotides adenosine diphosphate (ADP) and adenosine triphosphate (ATP). Energy and electrons are pumped from ATP to the dinitrogen substrate through eight cycles of association and dissociation of the Fe protein and the MoFe protein. The cycle is shown in **Fig. 41**. With each cycle, (I) a Fe protein loaded with two ATP molecules recognizes MoFe protein, (II) the Fe and MoFe proteins associate based upon electrostatic interactions and hydrogen bonding, (III) a conformational shift is induced by a change in the electronic structure of the 4Fe-4S and/or P clusters, (IV) ATP molecules within the Fe protein are cleaved to form ADP, thereafter energy and electrons are released to change the electronic states of the 4Fe-4S and P clusters as electrons are delivered to the substrate, (V) the Fe protein changes conformation to signal dissociation, (VI) the Fe protein dissociates, and (VII) ADP molecules are swapped for new ATP molecules.

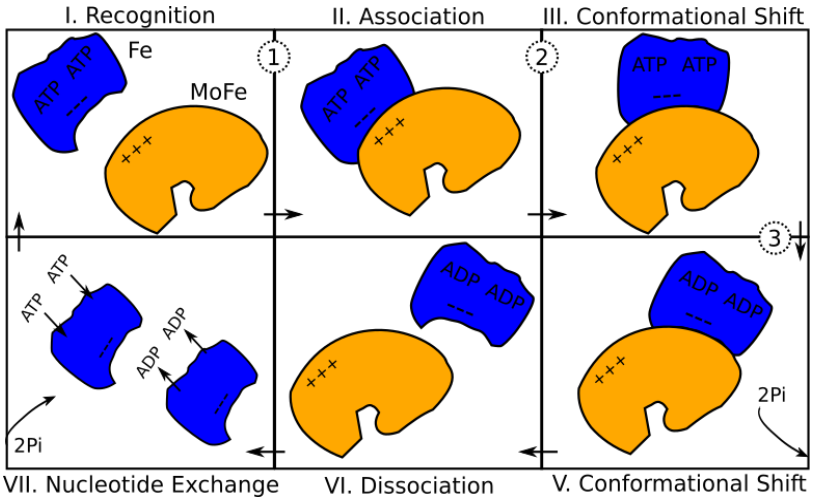


Fig. 41: *Sequence of docking, hydrolysis and undocking of the dinitrogenase reductase co-enzyme. Four cycles are required to reduce one dinitrogen molecule. Graphic by VJ.*

Much of the computational research into nitrogenase and its function has focused on the catalytic domain of the MoFe protein(104-106).While we see great benefit in a better understanding of the electronic structure and of the FeMo CF and its association with dinitrogen, ammonia, and various intermediates, there has been little focus on the enzyme

as whole, with a few notable exceptions(107, 108). The structure and dynamics of an enzyme underlies its function. Enzymes are well-tuned molecular machines, designed for accelerating chemical reactions. Because chemical reactions take place within a complex energy landscape, and because chemical intermediates are typically stabilized under a narrow set of circumstances, an enzyme's function is tied intricately to its three-dimensional structure. This is evident in the nitrogenase enzyme, in which researchers have found surprisingly few mutations that allow for the maintenance of the enzyme's function(109). The dynamics of a protein operate separately from the catalytic function. Dynamics are hypothesized to help enzymes attach to substrates, to move them into position near the catalytic domain, and to signal changes in binding or electronic states across distances much larger than are possible from pure van der Waals or electrostatic interactions(110). The phenomenon of communication within a protein or protein complex via protein dynamics is best illustrated by concerted allosteric mutations(111).

Enzymes and proteins have been observed to contain two or more mutations that have co-evolved to maintain or improve performance. When any one of these mutations is reverted to the wild type, function is lost, even though the physical distance between the mutations is relatively large and even when the structure of the protein remains largely intact. Based upon these observations of allosteric regulation of enzyme function via coevolution, we hypothesize that ATP, ADP, and different electronic states of the organometallic clusters(112) within nitrogenase are playing a similar role through long-range signaling induced by changes in protein dynamics.

Molecular dynamics (MD) simulations can be used to probe the dynamics of proteins and changes in dynamics based upon changes in chemical structures. However,

because nitrogenase contains metals, common MD interaction models (i.e. FFs) are insufficient by themselves to simulate this problem. While other researchers have also simulated nitrogenase using MD by creating FFs for the metal-containing portions of the enzyme(113), we sought to independently develop a model to verify the accuracy of previous models with respect to electrostatic interactions. We have therefore developed FFs for all the organometallic clusters and the CF using a modified AMBER protocol and inserted the new models into the existing MD framework. Protein complexes the size of nitrogenase can be simulated relatively easily on the microsecond timescale, requiring about a month of computational effort for a single complex on a single computing node.

Since the full nitrogenase cycle takes much longer than microseconds to complete, certain specialized techniques can be used to probe the enzymes fundamental modes of motion, namely principal component analysis (PCA) and functional mode analysis (FMA)(114). PCA is a method by which a large set of data can be projected onto a smaller set of fundamental vectors, and much of the underlying information can be retained. By performing PCA on the trajectory produced by an MD simulation, the slow modes of motion, typically containing most of the variance, are revealed. These slow modes of motion are hypothesized to be related to those that allow allosteric transfer of information across long distances across the protein structure. FMA is a method related to PCA by which a new vector is constructed as a linear combination of PCA modes. This new vector is one that is maximally correlated to some other observable in the MD simulation. For example, collective motions of the Fe protein can be related to the binding pose of an ATP or ADP molecule to reveal the ways in which protein dynamics change upon nucleotide switching.

Using our new FFs for organometallic moieties, we gathered tens of microseconds of MD trajectories representing various functional states of nitrogenase, and analysis has revealed fundamental differences in the structure and dynamics of the protein based upon nucleotide species and position. Ultimately, we believe these new insights will spawn new hypotheses about nitrogenase function and help researchers to unravel the mystery of nitrogenase's function.

METHODS

For this work, structure (4WZA) from *Azotobacter vinelandii* was selected from the Protein Databank(115), rendered in **Fig. 42.** Consisting of 3,102 amino acids, the

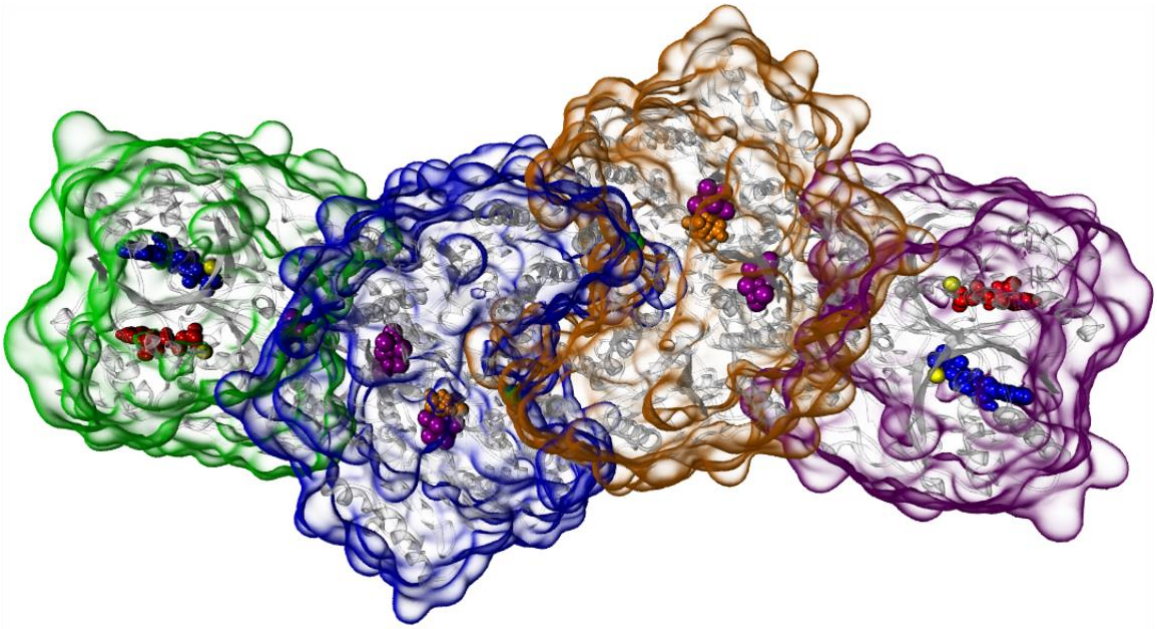


Fig. 42: *Structure of the crystalized nitrogenase. The ATP analog, (blue spheres), does not hydrolyze, stops the association-dissociation cycle of the reductase co-enzyme permitting the enzyme complex to be crystallized whole (reductases still bound) with the binding pockets populated.*

enzyme also contains two sets of the three unique CFs (purple), containing iron and molybdenum. The 4Fe-4S cluster is partially obscured in **Fig. 42** since it is positioned at the protein interface in the center of the reductase binding surface (i.e. at the green/blue

and orange/purple protein-protein interfaces). In general, the CFs are attached to residues by non-bonded interactions (e.g. hydrogen bonding, van der Waals, and electrostatic interactions) with the sole exception of one covalent bond between a cysteine sulfur atom and an iron atom of the Fe-Mo CF. The Fe-Mo CF has a second peculiarity, being chelated to a molecule of acidified homocitric acid (orange) which itself is attached to the residues solely by non-bonded interactions. Homocitric acid has three carboxylic acid and one alcohol functional groups along a four-carbon alkyl chain backbone. The acid group and the alcohol group bonded to the chiral carbon are deprotonated and form a chelating structure that associates with the molybdenum in the FeMo CF. Opposite this chelate, a deprotonated nitrogen of the protein's histidine residue forms a third ligand bond to the FeMo molybdenum atom.

To crystallize the full functional nitrogenase complex, ADP and a nonhydrolyzable ATP analog are added to the solution to fill the dinitrogenase reductase binding pocket. The ATP analog, phosphomethylphosphonic acid adenylate ester (ACP), cannot be hydrolyzed by the enzyme, but it is structurally similar enough to ATP to induce the changes in the structure of the reductase enzyme to allow association of the nitrogenase enzyme complex. However, because ACP does not cleave to release energy into the enzyme complex, dinitrogen reduction halts, facilitating precipitation of the complex in this form for X-ray crystallography. In **Fig. 42**, both the ADP (red) and ACP (blue) molecules are associated with a magnesium^{+2} ion (yellow). A final peculiarity to the complex is the presence of two iron atoms (green) at the binding interface between the two MoFe enzymes close to the homocitrate molecule. The function of these atoms is unknown, but they are included in our models, because they could be structurally important.

The AMBER99sb*-ILDN FF(116) was to be used to model protein interactions, TIP3P(88) was used for water, and Joung ions(117) were used for salts. However, the organometallic complexes and cofactors within nitrogenase would need to be parameterized. Challenges to building a simulation model fall into two main categories, namely: (1) determining the charge distributions in the metal-containing CFs and (2) building topology files for CFs and the other molecules not present natively in any FF. Considering each separately, we begin with the CC aspects, which will deal with charge distributions in the CFs.

Computation of Organometallic Cofactor Charges. CC seeks to approximate solutions for Schrödinger's equation. Since X-ray crystallography does not provide atom-centered point charges, as is required by standard MD simulations, the charges must be calculated approximately from quantum mechanics (QM) through Schrödinger's equation. QM methods scale exponentially as the number of electrons involved, and different orbital types require different algorithms. As nuclei become larger, core electron velocities approach relativistic speeds, and relativistic effects become non-trivial. Briefly, to address these hurdles and balance the fidelity of results with their computational cost, computational chemists split the problem into two parts. The first is known as the level of theory, and the second is known as basis sets. Models are used to account for orbital shapes and interactions among electrons in them. We selected a level of theory that generally produces the most accuracy per computational cycle, which is density functional theory (DFT)(118-120). DFT considers regions of varying electron density, rather than the more exact *ab initio* approach of calculating the track of each electron individually. The common

B3LYP functional(121, 122) was selected for its ability to handle all elements present in nitrogenase CFs.

For any problem, the appropriate or compatible basis set to choose depends on two factors, the types of orbitals occupied, and the degree of fidelity required. Basis sets have a wider selection than level of theory. The Basis Set Exchange(123), a web portal for basis sets, currently lists 589 basis sets in its database. Briefly, the two families used in this work are the Pople and Dunning basis sets. Pople basis sets are generally used for molecules comprised of smaller atoms (i.e. the first three rows of the periodic table). Although parameterization was attempted through 4^{th} row elements, where d-orbitals begin occupancy, Dunning basis sets give better results for the same computational cost. Iron is a 4^{th} row element, but molybdenum is a 5^{th} row element, where core electron interactions must account for their relativistic velocities. Effective core potentials (ECP)(124) are a method for dealing with these relativistic effects.

A simple approach to dealing with systems including atoms from hydrogen to molybdenum, is to select the smallest basis set that can handle all the atoms in the system. In fact, for smaller (i.e. common organic) molecules it is so typical to do so that this is the default behavior of the software we use to assemble CC problems, Gaussview. Gaussview is generally able to determine appropriate settings for the QM program we use, Gaussian(125). However, for systems with atoms from hydrogen through molybdenum, like the CFs within nitrogenase, one calculation could easily take months.

Gaussian and Gaussview have the capability to define specific basis sets for specific elements in a simulation, in fact this can be specified per atom if needed. Modeling the

organometallic CF charges calls for all the fidelity that can be mustered. By specifying the basis sets on a per atom basis, a wall time of 106 days for the FeMo CF was reduced to 8 days. For carbon and sulfur atoms a Pople 6-31G++(d,p)(126) basis set was used. The same basis set was used for elements in rows 1-3 in the other molecules (ADP, ATP, homocitrate) as well. The appropriate basis set for the iron and molybdenum atoms was determined to be a Dunning aug-cc-pVDZ basis set(127) with ECP for molybdenum. The Gaussview cc-pVDZ (double-zeta) basis sets to not include parameters up to iron, so the orbital parameters and ECP parameters for iron and molybdenum had to be downloaded from the Basis Set Exchange. For calculating atom electrostatic potentials (ESP), using the correct van der Waals radii (VDR) for the atoms is critical. The elements in molecules requiring QM calculations were H, C, N, O, P, S, Fe, and Mo. For these VDR of 1.2, 1.7, 1.55, 1.52, 1.8, 1.8, 2.44 and 2.45 Å were used. Quadratic convergence for the self-consistent field method was required to achieve convergence of the more complex systems. All other simulation parameters used were the Gaussview defaults.

Preparation of MD Simulation Files. In the GROMACS standards, the topology files hold the details of molecular connectivity, atomic charges, bonded parameters, and Lennard-Jones parameters. Although ADP and ATP are technically nucleotides, they are not components of RNA or DNA and are not known to the FF. Therefore, seven molecules or residues required the creation of topologies so they could be included in the MD model.

Adenosine diphosphate	ADP
Adenosine triphosphate analog	ACP
Homocitric acid	HCA
4Fe-4s cofactor	SF4
P-cluster cofactor	CLF
Fe-Mo cofactor	ICS

Fig. 43: *Protein data bank residue codes for the non-protein molecules present in nitrogenase. Although the ACP molecule is altered to become an ATP molecule, the native simulation files retain the ACP code throughout.*

Creating specialized simulation files is a multi-step process, dependent upon which parts are natively supported by existing FFs, and which parts must be calculated elsewhere. From the original structure file (.pdb) from the Protein Data Bank, molecules requiring specialized topologies were assigned three-letter residue codes as listed **Fig. 43**, and these codes will be used throughout this section.

In **Fig. 44**, ACP (blue spheres) sits in a reductase binding pocket. The phosphate side of the molecule is closest to the Fe4-S4 CF. The substituted atom in ACP is the linking

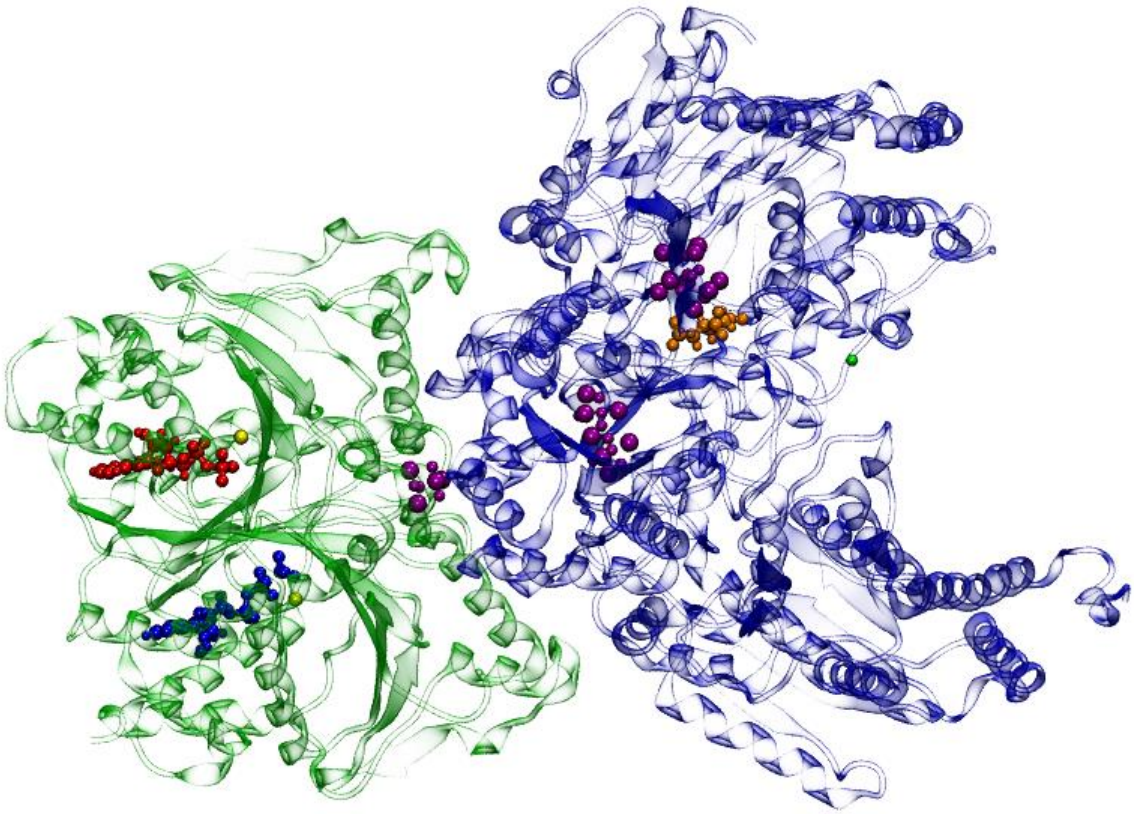


Fig. 44: *Location of the non-protein molecules, atoms, and ions in one catalytic unit. Green: reductase "Fe" protein and Fe atom; Blue: nitrogenase FeMo protein and ADP. Red: ACP "ATP analog"; yellow: magnesium^{+2} ion; purple: organometallic cofactors, l-r 4Fe-4S, P-cluster, FeMo; orange: homocitric acid.*

oxygen between the two distal phosphates, which has been replaced by a carbon atom. This was changed back to an oxygen atom in the .pdb structure file for both molecules. From

the new .pdb file, individual .pdb files were extracted for the six simple organic molecules and a representative for each CF type. The structure deposited in the Protein Data Bank does not include hydrogens. For the six non-metal molecules, these missing hydrogens must be added. To add these hydrogens with the correct coordinates, each individual file was read into Gaussview wherein missing hydrogens can be automatically generated in the proper location. The six molecules were then written back out as .pdb files with the hydrogen atoms. For the metal-containing P-cluster and FeMo CF, Gaussview created additional bonds based on atom proximities. These were removed so that the bond structure again match those in **Fig. 40**. Note, that for clarity, the bonds of the central carbon atom in the FeMo CF are not shown. It has six bonds, one to each of the proximate iron atoms as shown in **Fig. 45**. The organometallic CFs did not require a transfer back to the .pdb format, because they do not require the addition of hydrogen atoms. The organic molecules, now containing hydrogen atoms, were placed back in the main .pdb file.

Atom Charge Calculations. Electrostatic potential (ESP) calculations needed to be performed on the organic molecules to derive atom-centered point charges. The instructions for electronic structure calculations for each of the hydrogenated ADP, ATP and homocitrate molecules were written to a Gaussian input file (.com). Gaussview was also used to generate structures in a new .mol (MDL MOL) format, which tracks not only atomic positions but connectivity and point charges. For the organic molecules, the .mol file can be output from the structure used to specify the QM calculations. For the metal CFs, no hydrogens are added, but ESP calculations are performed nearly identically. No .pdb output file is required, but before saving the structure of the CF as a .mol file, each requires specific manual substitutions. For the 4Fe-4S CF the iron atoms are changed to

carbon. For the P-cluster, the iron atoms are changed to sulfur. For the FeMo CF, the central carbon atom is changed to phosphorous, the six central irons to carbon, the top iron and its three bonded sulfurs to nitrogen as well as the molybdenum and its three bonded sulfurs to nitrogen. The changes in the identity of these atoms are important to dupe programs that will be used later in the development of the FFs for these CFs. Because standard FFs used within biomolecular simulations recognize bonding structures of common organic elements but not metals, programs associated with these FFs will typically fail. The strategy we employed was to use existing programs to create skeleton files that could later be modified to fit our specific, unique needs.

For the FeMo CF, it is still not clear after 50 years of research the exact location where the dinitrogen molecule is initially bound. For this reason and because the homocitrate molecule chelates to the molybdenum atom, itself positioned solely by ligand bonds, QM calculations were performed for the Fe-Mo CF with and without the presence of the homocitrate molecule to evaluate its influence on the CF charges.

Creating Atom Position Files. A unique covalent bond from a cysteine sulfur atom to the FeMo CF was earlier mentioned. A modified cystine residue, named CYX, was used to remove hydrogen from the end of the sidechain, leaving the sulfur open to bond with iron. In the 4WYX .pdb file, these residues have the native cysteine code CYS, which must be manually changed to CYX as well. A file named residuetypes.dat defines the residue class each residue will belong to. GROMACS natively recognizes five types, proteins, DNA, RNA, water, and ions. For the purposes of nitrogenase, entries were added here as proteins for ADP, ATP, and HCA.

To test the effects that bound nucleotides have on the structure and dynamics of the nitrogenase complex, additional models were generated based upon the structure described above. There were four additional structures to generate. One with the nucleotides removed, one with two ADP molecules in each reductase enzyme, one with two ATP molecules in each enzyme and one with the position of the original ADP and ATP molecules swapped. The precise geometry of the two binding pockets in the crystal structure are not identical, nor are the bound ADP and ATP molecules. To address this difference for the system with ATP molecules (i.e. crystallographic ADP must be changed to ATP), the ADP molecule had a third distal phosphate added to match the existing ATP geometry. The coordinates of the new ATP molecule were however not compatible with the old ADP binding pocket. Materials Studio was used to detect close contacts and to find positions and orientations to satisfactorily fit new ATP molecules into the old ADP pockets.

Building GAFF Topology Files. Normally, a protein simulation in GROMACS begins in the utility *pdb2gmx*, which builds the GROMACS topology files from an experimental structure, which in our case is stored as a .pdb file. The *pdb2gmx* utility also typically modifies the structure to contain missing hydrogen atoms and to introduce disulfide bonds. Each of the four proteins in the nitrogenase complex are comprised of two amino acid chains, each of which are describe structurally by a separate topology modifier file with a .itp extension. The first eight topology files generated by the *pdb2gmx* utility, as well as those for the two lone iron atoms and the four magnesium ions are natively supported. However .itp files must be generated for any molecules that are not natively present in the FF or explicitly added by the user. Fortunately, the AMBER family of FFs

benefit from the General AMBER FF (GAFF), which was developed to model most organic molecules with standard bond orders. GAFF, through a set of programs and scripts to be described below, was used to create FFs and topologies for simulation components unknown to the native AMBER99SB*-ildn FF. Remember that iron and molybdenum were substituted for other elements in earlier files. Because GAFF does not contain these elements, the false atoms were used to dupe existing programs. Files were earlier converted to the .mol format to accommodate the method by which GAFF creates topology files, a program called Antechamber(128). Open Babel(129) was used to convert the .mol files from Gaussview to the .mol2 format used by Antechamber.

Antechamber is a suite of 14 auxiliary programs for molecular mechanics studies, the use of which has been streamlined through a, open-source script known as *acpype.py(130).* An example of the *Bash* invocation for the homocitrate molecule is:

./acpype.py -i HCA.mol2 -c user -f

The -c option of user deselects the atom charge calculations and the -f option forces the script to not use existing files but instead rerun all Antechamber functions from scratch. The native GROMACS elements generated are the structure (.gro), FF and topology modifier (.itp), and topology (.top) files. The molecules' structures were visually checked in VMD. The .itp files were checked for the presence of dummy atoms, which would indicate a deficiency in the input .mol2 file that could not be resolved. Each .itp file produced by Antechamber starts with a listing of the atom types present in that topology file. An atom type may only be defined once in the stack of topology files, and they must all be defined before subsequent .itp file statements reference them. Consequently, each unique atom type occurrence was added to the atom types section in the unifying

AMBER99sb*-ILDN ffnonbonded.itp file. For this system 29 additional GAFF atom types were needed. For the CFs, the duped atom types in the atoms section of the .itp files were changed back to the real atom types. Atom names in the Antechamber-generated .itp files were modified to match those in the original .pdb file. The atom charges resulting from the QM calculations in Gaussian are used to populate the charge values (see **Fig. S22**) for each atom, and the masses must be corrected for any atoms substituted in the Antechamber steps. All the bonds lengths involving substituted atoms must be corrected. All the CF structures have symmetries. Bond lengths were calculated by averaging the distances in the original .pdb file for each of the symmetric bond types. For the 4Fe-4S CF all the bond lengths were set to 0.219 nm. See **Fig. 45** for these values for the other two CFs. The determination of

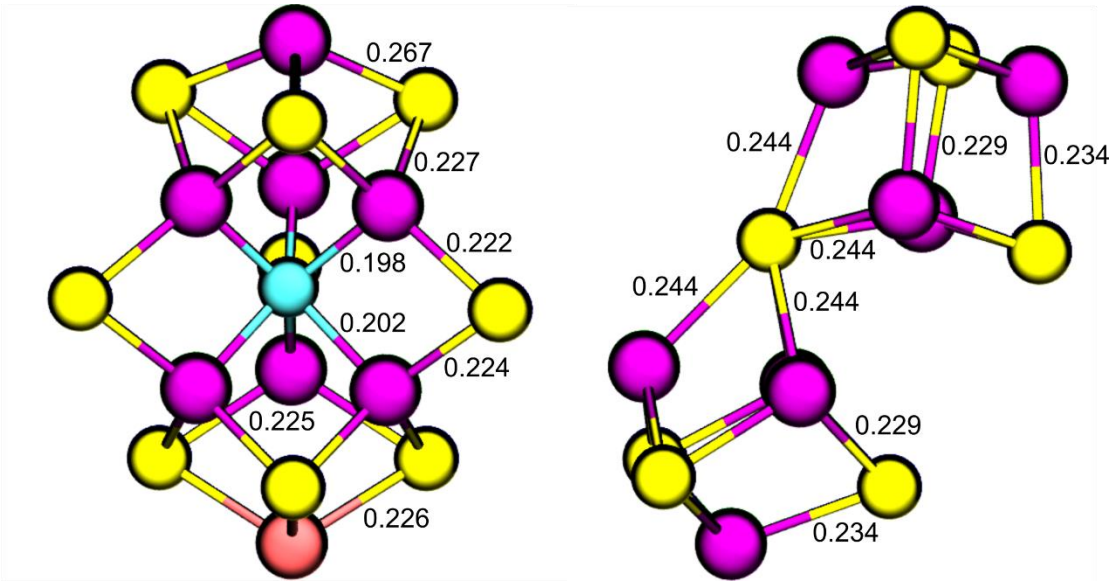


Fig. 45: *Bond lengths of symmetric bonds in 4Fe-4Mo CF (left) and P-Cluster CF (right) bond lengths are shown in units of nm. Atom colors are same as in* ***Fig. 40****.*

the correct bond energy elastic modulus for the metal CFs (mostly Fe-S bonds) was challenging. No references were found for the specific CFs present in nitrogenase. The most relevant reference found(131) listed bond energies for Fe-S bonds in two

configurations. Their "bridging" structure was virtually identical to the equatorial sulfur bonds in the FeMo CF and reported a modulus of 1.44 millidynes/Å, which is equivalent to 8.672×10^4 kJ/(mol-nm^2). With the difficulty finding experimental data for these very esoteric bonds, we decided to use this value for all the metal-containing CF bonds, including the ligand bonds.

When GROMACS creates an .itp file, by default it inserts conditionally expressed statements to apply positions restraints, often needed for the initial energy minimization step. These will not be present in the .itp files created by ANTECHAMBER and must be added. For reference, these are the lines added to the .itp file for the FeMo CF:

```
#ifdef POSRES
#include "../+/nitrogenase/posre_Protein_chain_A3.itp"
#endif
```

Assembling Main Topology File. When the previous tasks have been completed, the preparation of the main topology file (topol.top), which is produced by the *pdb2gmx* utility, can begin. In the chain topologies section, the file names must be changed to match the file names output by *Antechamber*. If the molecule type name was changed in the .itp file, the names in the molecules section must be changed to match. We defined ligand bonds that hold the CFs in position. These parameters are placed at the very end of topol.top. Ligand bonds used harmonic bond potentials with a modulus matching the organometallic bonds. With these modifications, our model systems were processed through standard GROMACS preparation steps to add approximately 81,000 water molecules with utility *solvate* and to add sodium and chloride ions at a concentration of 150mM with utility *genion*. At physiological pH, nitrogenase a net negative charge of -62q or 62e$^-$. An additional 62 sodium ions are added in the *genion* step to neutralize this charge.

The smallest systems we built had rectangular (orthogonal) parallelepiped periodic boundaries with a 12 Å buffer between any part of the protein complex and simulation extents. In total, the entire systems contain over 300,000 atoms.

Steepest descent energy minimization was conducted for up to 50,000 steps with position restraints on all protein atoms including CFs. The energy minimized systems were used to build 1 μs production simulations in the NPT ensemble. Virtual sites were generated to increase the integration time step to 4 fs. Pressure was controlled at 1 bar using the Berendsen barostat, and temperature was set to 300 K using a stochastic velocity scaling thermostat. Non-bonded interactions were cut off at 1.0 nm. Electrostatics interactions were calculated explicitly at distances below 1.0 nm and using the particle mesh Ewald summation method at longer distances. Additional parameters are presented in the *grompp.mdp* file listed in the Supporting Material.

RESULTS AND DISCUSSION

The processes used to establish the MD model were repeated and documented from scratch while scripting as much of them as could be to eliminate manual errors. A total of five distinct systems were built, all with net organometallic cofactor charges of zero. The distinction in between these systems was the molecule present or not in the reductase binding pockets. 1) Original 1L2Y.pdb (with ACP changed to ATP) 2) Both binding pockets empty. 3) ATP in both binding pockets. 4) ADP in both binding pockets. 5) Original with ADP and ATP in swapped positions.

The root mean squared deviation(132) (RMSD) was calculated on the trajectory of each model system to determine if the simulation models were stable. The RMSD analysis

compares the experimental crystal structure to the structure is assumes as it relaxes into its functional state. For proteins of this size, a structure with an alpha carbon RMSD less than 0.4 nm is generally considered reasonably stable. Because the protein migrated over the periodic boundary, and because the protein consisted of several chains, RMSD analysis was performed independently on each of the protein chains, of which there are two for each of the four nitrogenase complex proteins. The nitrogenase chains are 477 and 522 residues long, identified as the chain pairs AB and CD in the experimental .pdb file. The reductase chains are 276 and 271 residues long identified as chain pairs EF and GH. RMSD analysis (**Fig. 46**) showed that all the models were stable, indicating no serious errors in the simulation model, or in the preparation of the GAFF components of the system, and reasonable simulation fidelity of the system with this FF. The RMSD of each system is displayed in the five panels of **Fig. 46** and identified by the nucleotide codes (ADP, ATP, EMPTY) specifying which are present in the reductase binding pockets. Variations in the response of each chain in the five systems is hypothesized to be attributable to allosteric adjustments for nucleotide occupancies.

Root mean square fluctuations (RMSF) of the alpha carbons versus the average trajectory structure were calculated. RMSF quantifies the relative motion of each residue referenced to an average structure. This helps identify which parts of each complex are most mobile or stationary, which are important clues for deciphering complex dynamics of the protein. The nitrogenase protein made of chains AB is physically docked with the reductase protein, made of chains EF. Because of their physical proximity in the complex, plotting should place them side by side. The same reason applies to the two nitrogenase proteins and the second nitrogenase and second reductase protein, therefore the most

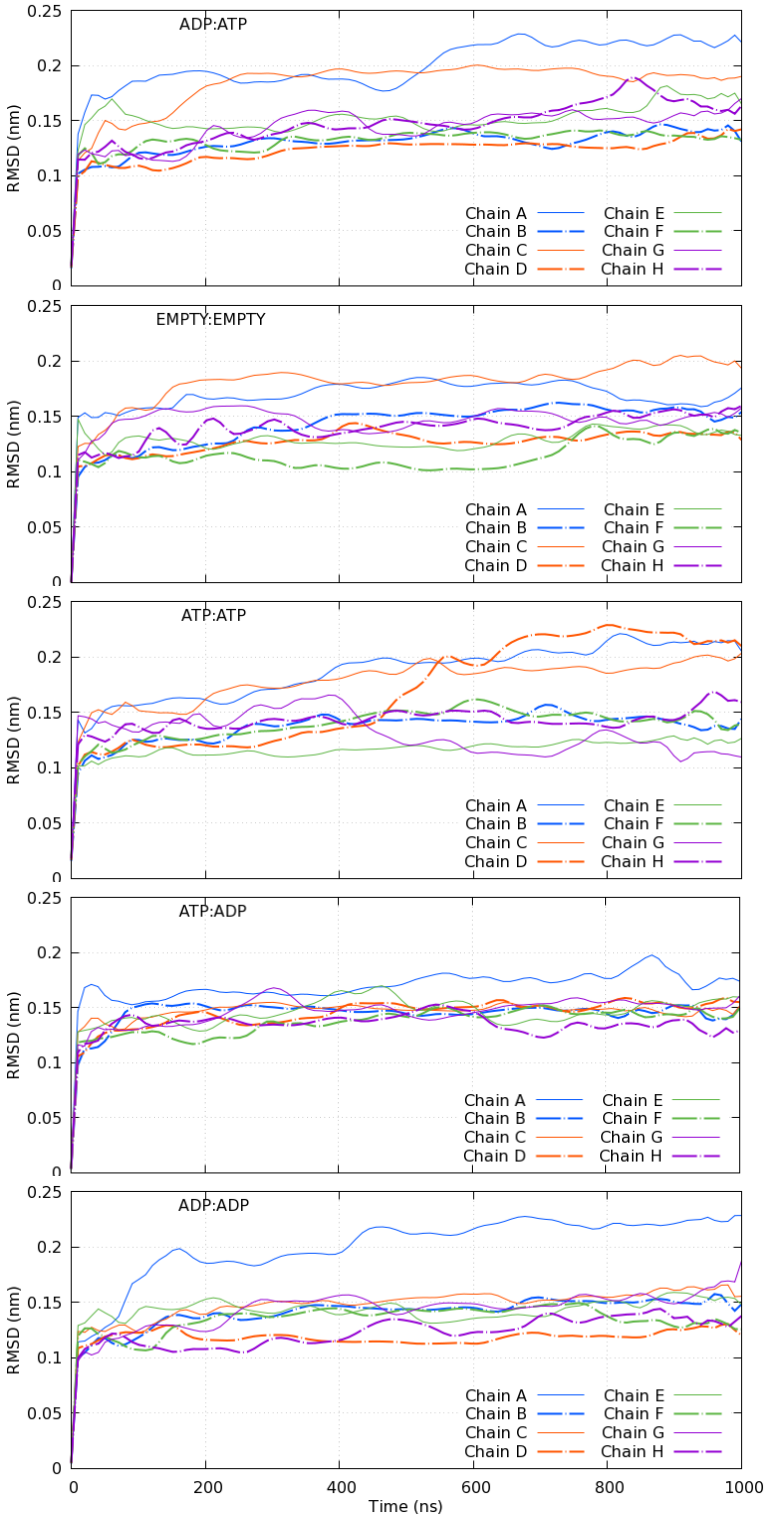
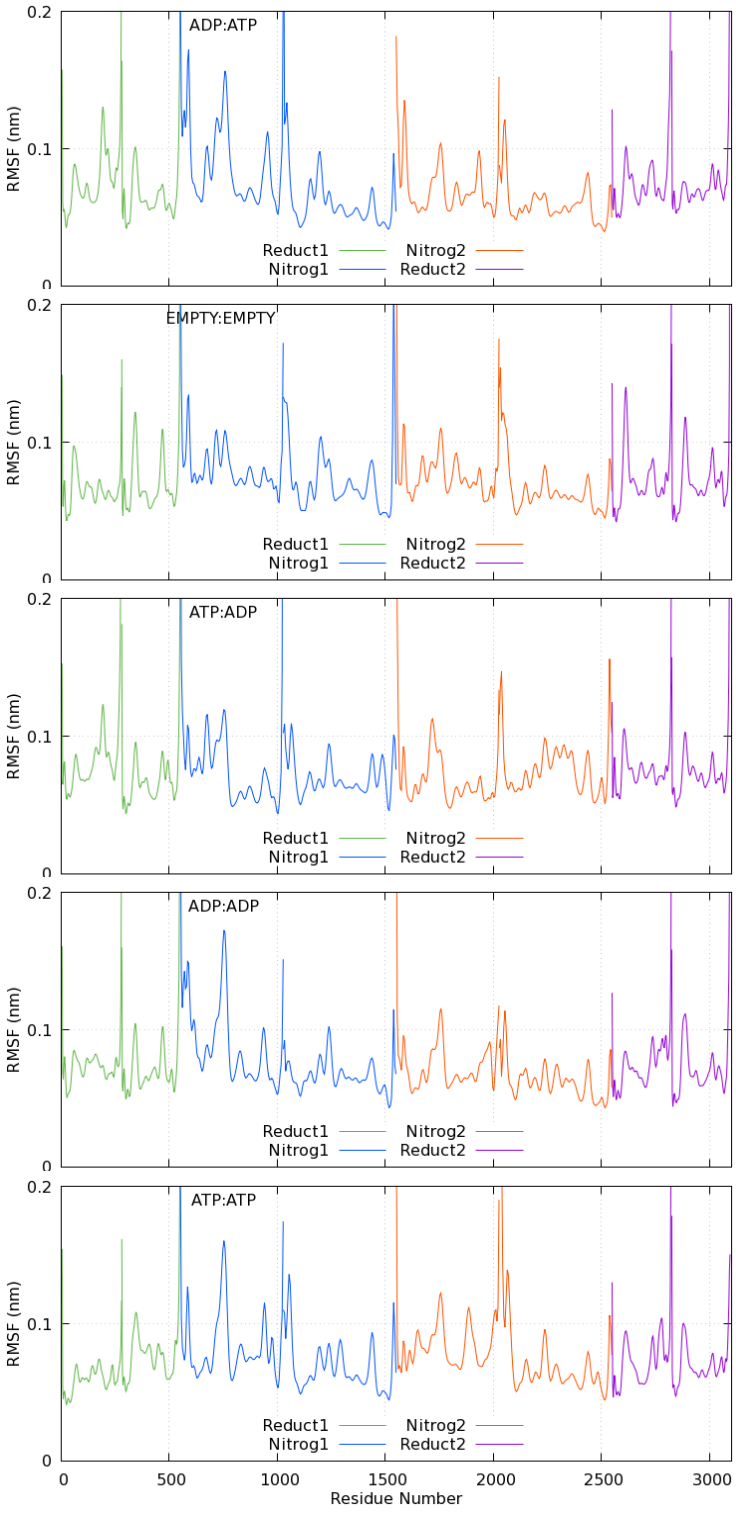


Fig. 46: *RMSD plots. The RMSD plots show the relaxation of atom positions from the initial crystal structure in the first 20 ns. The molecules subsequently relax very slowly but they do not unfold.*

appropriate sequence in the plots are Reductase 1, Nitrogenase 1, Nitrogenase 2 and Reductase 2 as shown in **Fig. 47** Panels 1-5.

A primary goal of this work is to develop an all-atom model with which to investigate the role nucleotide species and the resulting conformations play in driving nitrogenase function. We, therefore, analyzed the differences in equilibrium structures induced by the presence of various bound nucleotides. The images are created by taking the last frame of each of the five systems at the end of 1 μs. The α-helices in the



complexes are analyzed by Bendix(133), a routine that color codes the severity of the bends. The resulting plots help visualize variations in complex conformation due to the changes in the occupancy of the binding pockets. These images reveal significant differences among the five model systems. To help with comparison, two pages of three images are shown. The second panel of each page is the reference conformation of the complex, the one with empty binding pockets. **Fig. 48** compares this to the original ADP-ATP system from the protein data bank in

Fig. 47: *RMSF plots. The RMSF plots show the magnitudes of motions along the protein backbone. Peaks reveal where the protein backbone is moving freely. The motion at the ends of the protein chains cause blips at the ends and centers of each enzyme section.*

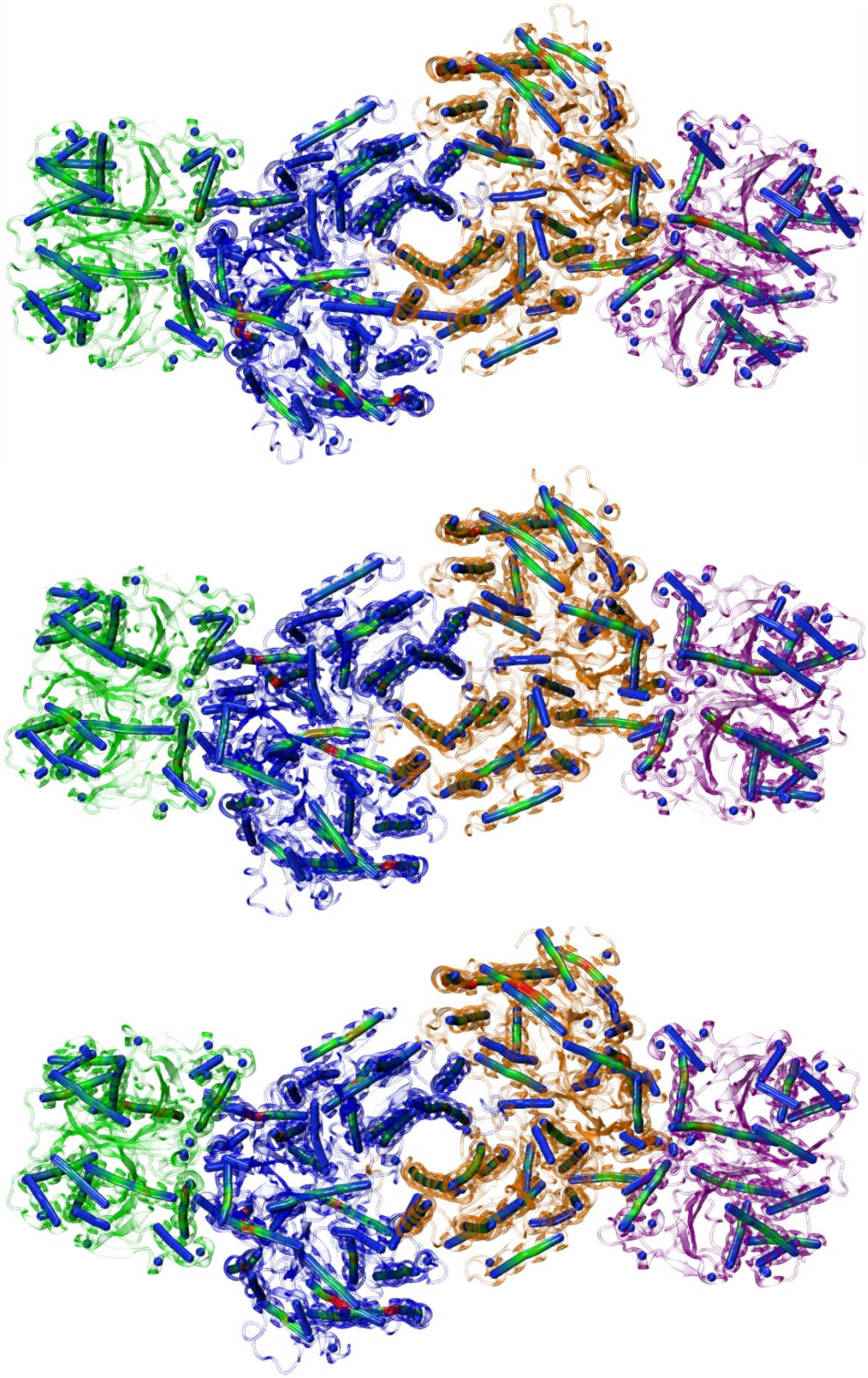


Fig. 48: *Bendix plots of ADP-ATP, EMPTY and ATP-ADP binding pocket occupancy.*

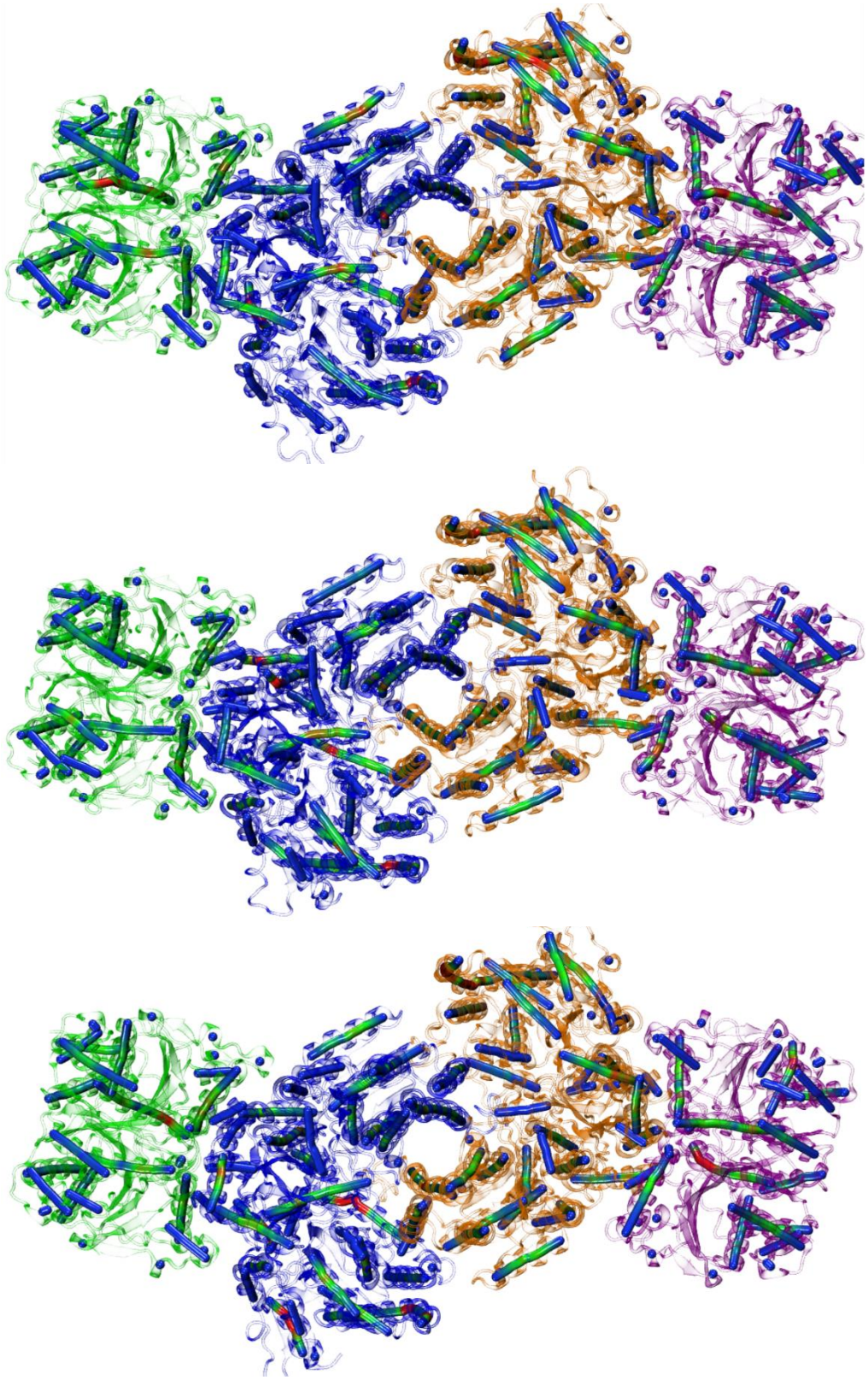


Fig. 49: *Bendix plots of ADP-ADP, EMPTY and ATP-ATP binding pocket occupancy.*

the top panel and the ATP-ADP system in the bottom panel. **Fig. 49** allows the comparison of the reference to the ADP-ADP system in the top panel and the ATP-ATP system in the bottom panel. All five systems display unique conformations that differ from one another to a surprising degree.

CONCLUSIONS

This work demonstrates that microsecond-scale simulations can provide crucial information about fundamental structural and dynamical changes in an enzyme complex, even when the enzyme functions over much longer timescales. The methods developed for organometallic complexes show promise for future simulations of other proteins and enzymes with inorganic ligands, which until this point have been rarely explored using MD simulations due to difficulties in deriving interaction models. Due to the complexity of the nitrogenase cycle, there are a multitude of future simulations that will be performed on this system. There are several known nitrogenase structures that probe the effects of nucleotides on the protein-protein interface. The incorporation of these additional models will be used to strengthen our observations. Other parts of the enzyme will be modified to probe additional legs of the nitrogenase cycle, including the charge state of clusters and cofactors, the presence of dinitrogen and intermediates during the transition to ammonia, and the structure and dynamics of the unbound dinitrogenase enzyme. The mechanistic insight provided by these simulations will likely assist in the development of an improved nitrogenase enzyme or the engineering of a bio-inspired nitrogen reducing catalyst.

SUPPORTING MATERIAL

Supporting Material – Section 1

Electrostatics

Electric charges exert forces at a distance on surrounding charges via a first order tensor called the electric field, often represented by electric field lines whose density through a plane normal to their direction is equivalent to the strength of the electric field. Electric flux is the average density of these field lines through a defined surface, finite as well as infinitesimal, without which the strength and direction of non-uniform electric fields cannot be described.

Since the external electric field the membranes are exposed to in these simulations is generated solely by the transmembrane charge imbalance, each membrane is also exposed to a finite quantity of electric flux. When the electric flux is distributed uniformly over the surface of the membrane, the electric field is uniform, and we can simply and accurately say the membrane is exposed to a specific transmembrane potential, being the product of the uniform electric field to which the membrane is exposed and the thickness of the membrane.

Once the membrane structure interacts with this external field, having overcome the formation energy barrier, and develops persistent non-uniform structure, non-uniform electric fields develop, meaning they now vary in intensity and/or direction as a function of position. However, since the only source of the original external electric field was charge imbalance, and the CompEL method restores that charge imbalance every time an ion crosses through a pore, the membrane remains exposed to a finite, unchanging amount of

electric flux. These conditions thereby enable us to investigate and study the equilibrium conformation of membrane pores with electric flux.

On all graphs, Charge Imbalance on the x-axis is equivalent to and scales identically with Electric Flux.

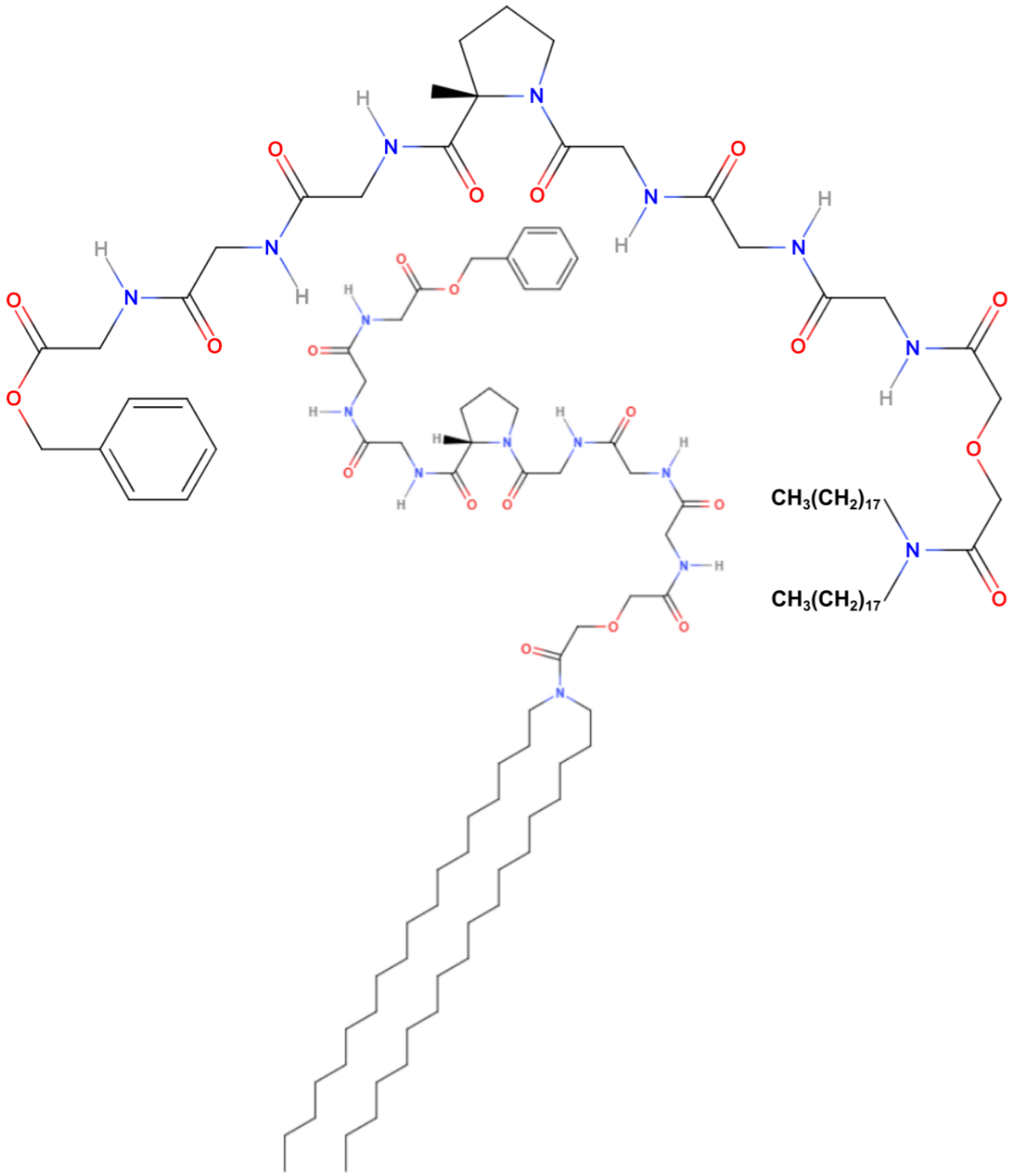


Fig. S1: *SCMTR Molecule: SCMTR molecules are designed to be lipid-like with tails like glycerol derived lipids linked to a heptapeptide section placed between the tails and various headgroup types, a phenyl fragment in this case. Variations of SCMTRs include different tail lengths, tail saturation, tail cis-trans geometry, amino acid sequences and headgroup types. The GLY-GLY-GLY-PRO-GLY-GLY-GLY ($(C_{18}H_{37})_2NCOCH_2OCH_2CO$-GGGPGGG-$OCH_2Ph$) sequence shown here is generally considered the most active.*

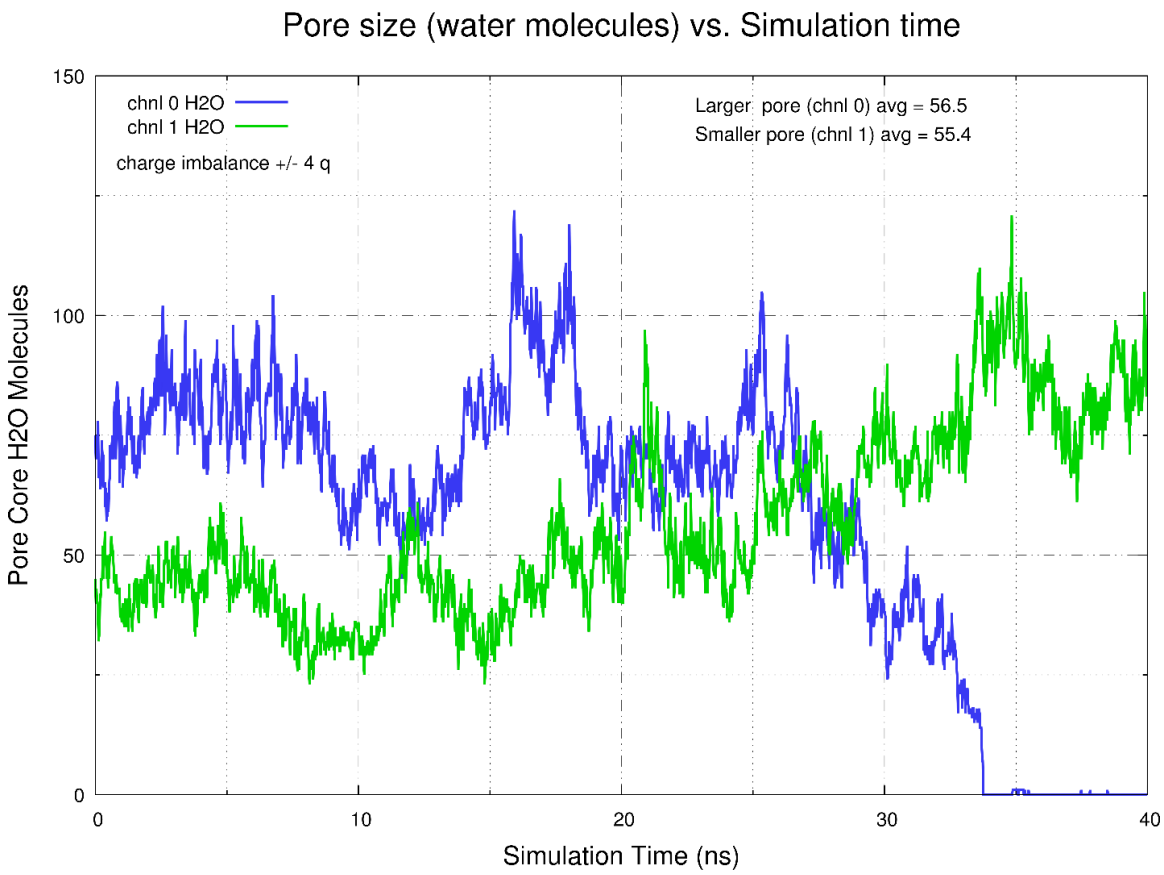


Fig. S2: *History of pore core water molecules in upper and lower membranes during one 40 ns simulation. The method of measuring these values on a frame by frame basis is shown and explained below.*

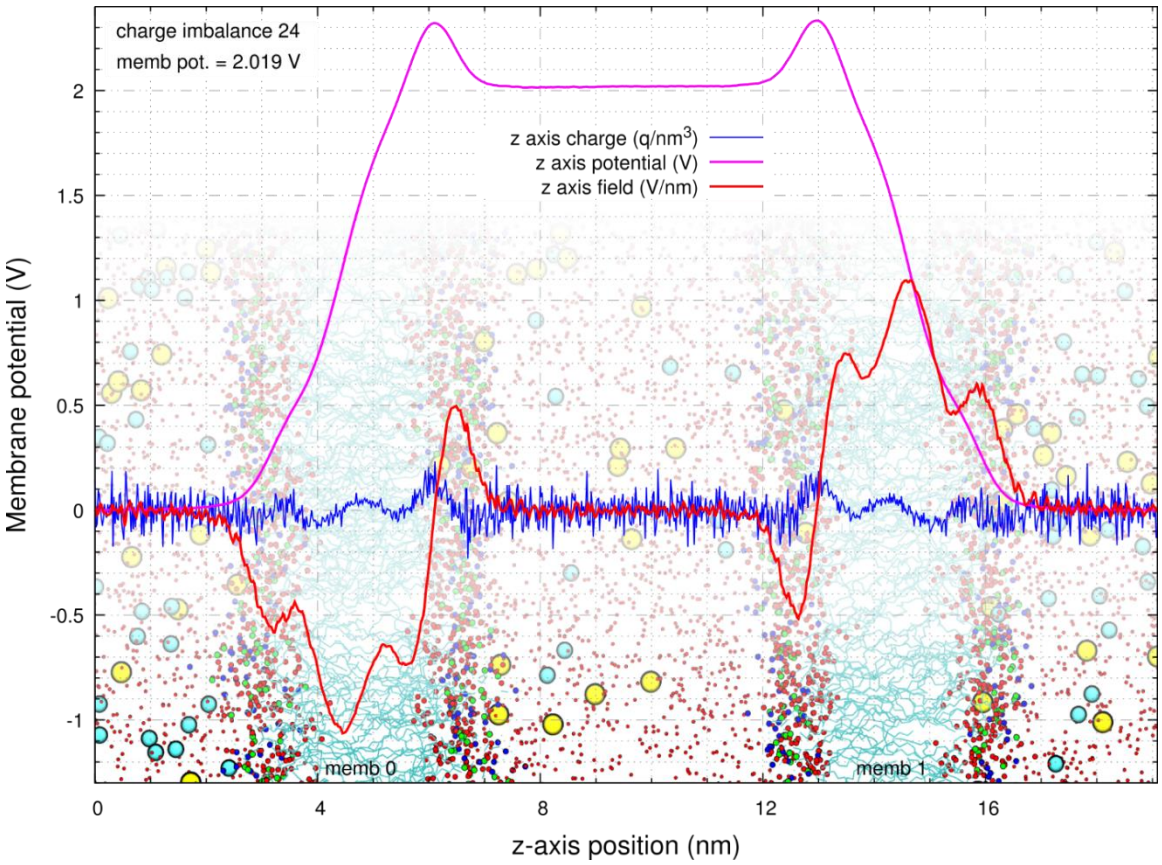


Fig. S3: *Details of electrical parameters for pre-pore stage of large (4X) system. To equate these details to the membrane structures, a background image of an accurately scaled frame from the trajectory used for the GROMACS potential analysis was added. Since the simulation conformation at this stage is uniform over the x-y plane, the resulting analysis data is valid. Lipid headgroup atoms are same colors as* ***Fig. 2****. Water oxygens are rendered as small red dots.*

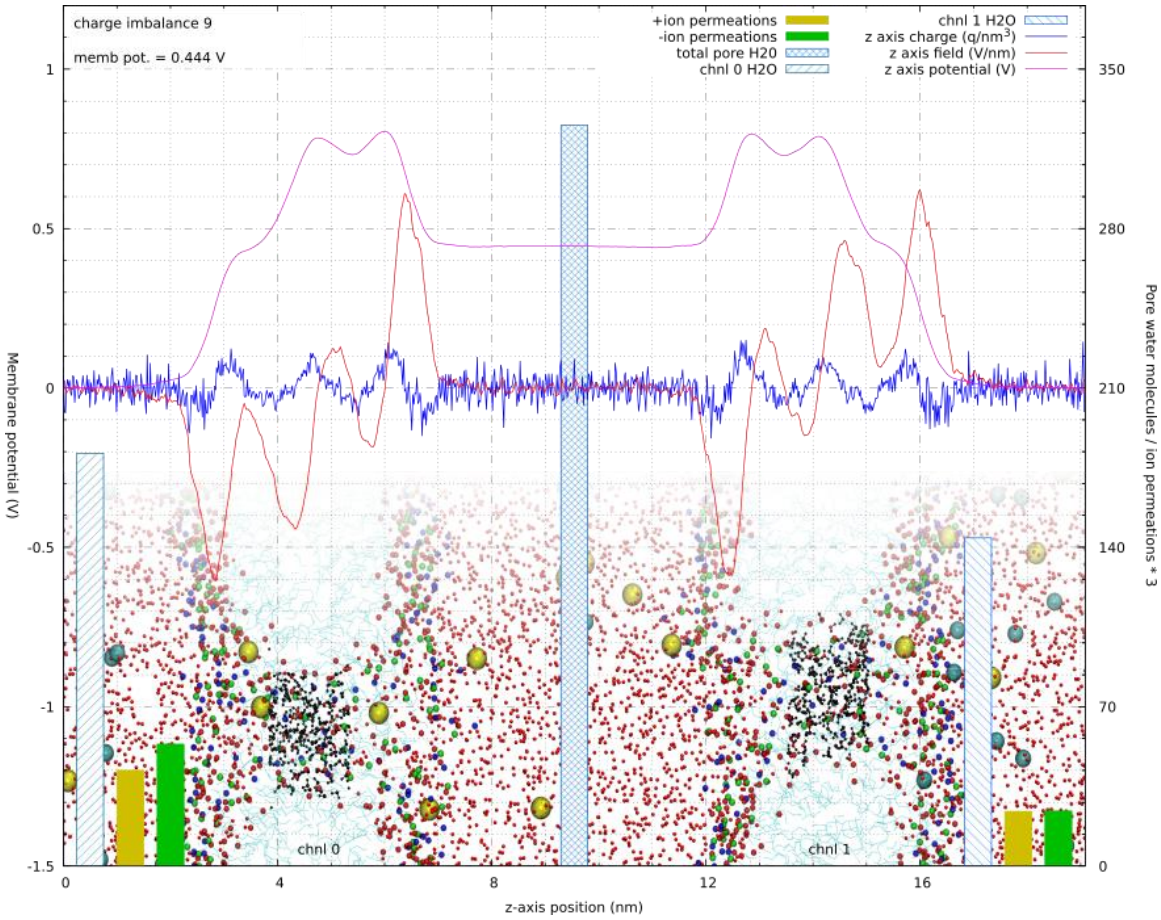


Fig. S4: *Results of GROMACS potential analysis for a 40 ns production run combined with pore size and permeation event details (see legend). Core water molecules are rendered as black CPK. A high-fidelity pore development model will require an analysis tool to elucidate the true details of the three-dimensional non-uniform electric field in this conformation.*

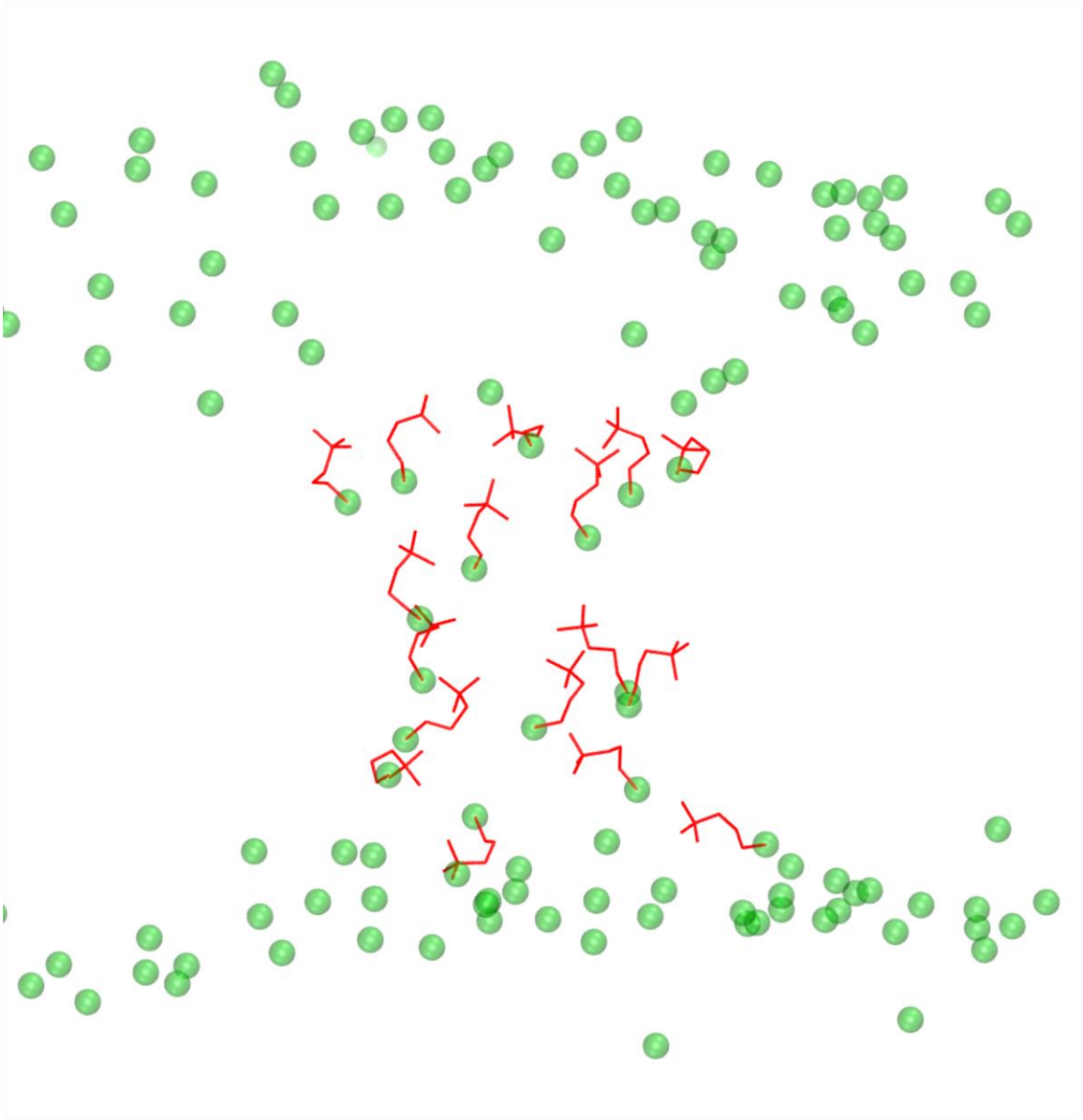


Fig. S5: *Vertical alignment of choline moieties extending into center of pore from the phosphorous atoms of the DOPC headgroups in the direction of the concentrated electric flux, i.e. higher electric field. Orographic projection orthogonal to z-axis. Choline moieties are shown if the phosphorous atom is in the "core" of the pore, i.e. +/- 0.75 nm from center of membrane along the z-axis.*

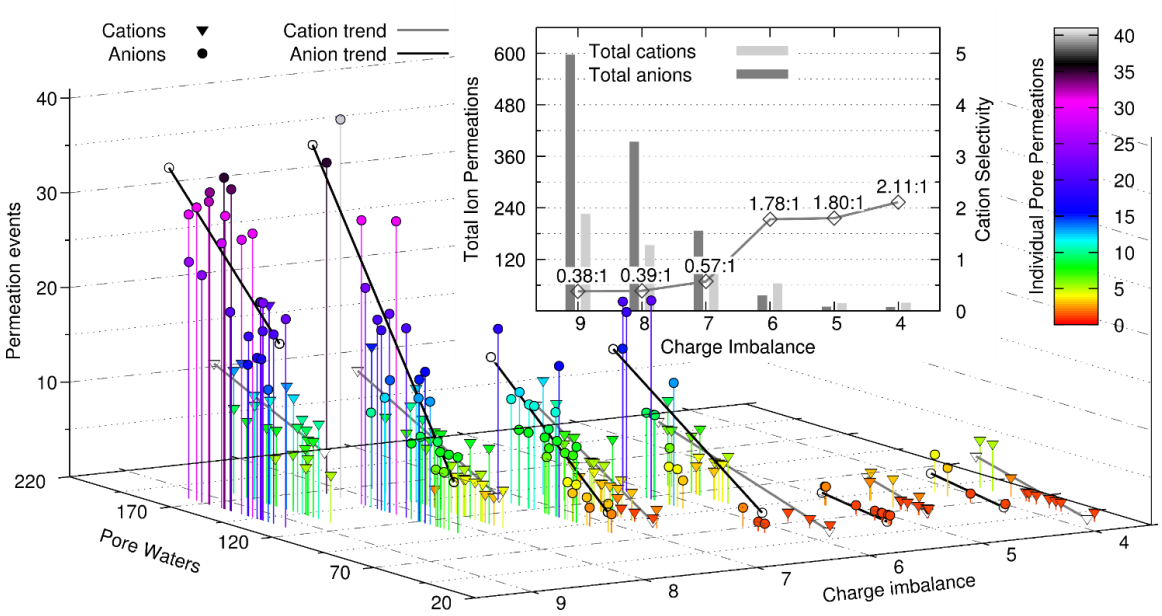


Fig. S6: *Ion permeation rates vs. pore size and charge imbalance for pores containing SCMTRs. Data about each charge imbalance (integer values) are split left and right to separate permeation of anions and cations. Note relatively constant trend of cation diffusion vs. pore size compared to steadily increasing trend of anion diffusion.*

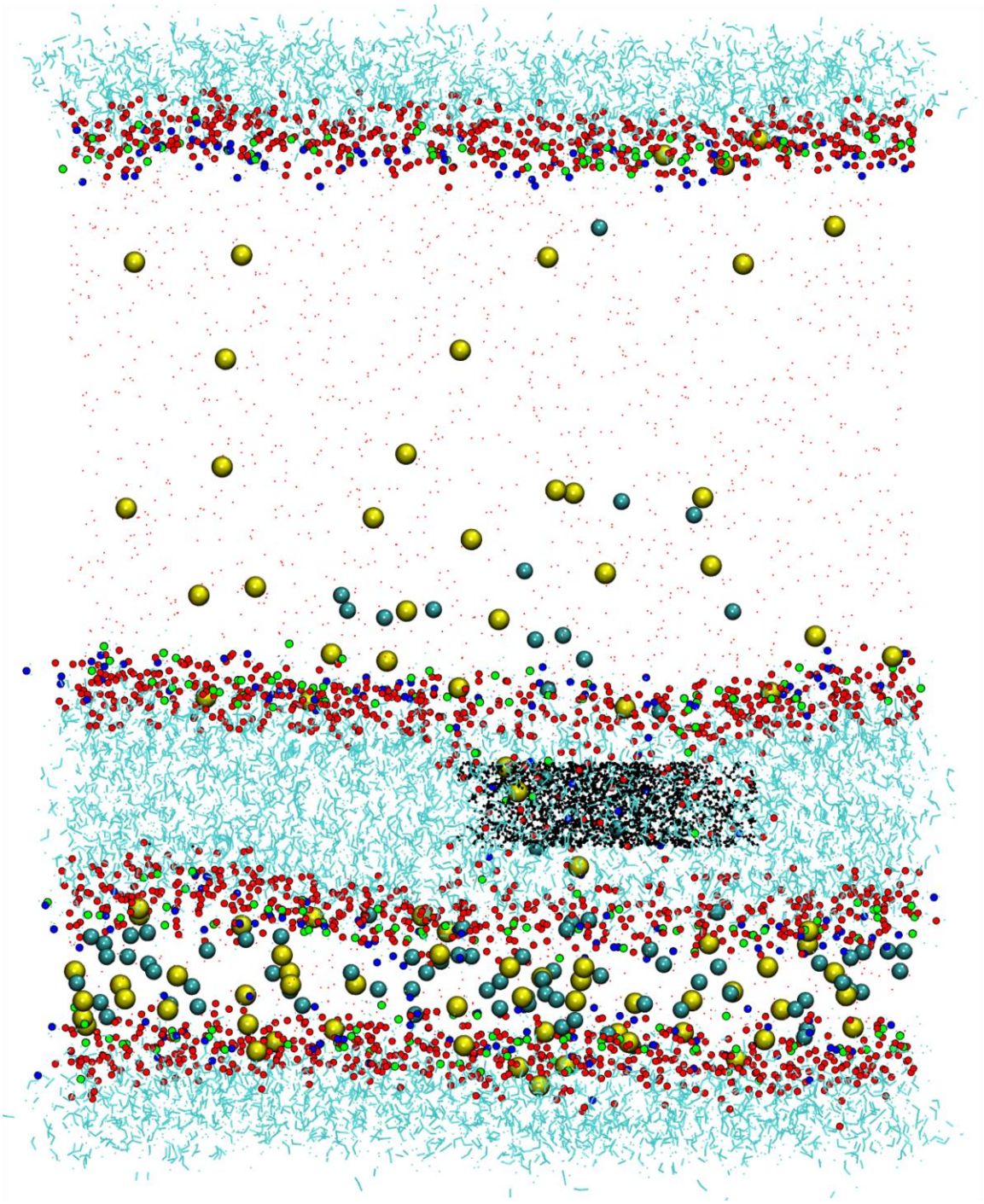


Fig. S7: *Large system after 125 ns. Note distribution of chlorine ions (cyan) in upper bath. This is the extreme of the asymmetric development of the system. Disparity of data between this system and the smaller is inescapable until the larger system can be simulated in a fashion to mitigate or eradicate this asymmetric development. Similar size pores in each membrane may be sufficient.*

Production run .MDP file including CompEL parameters

```
integrator              = md
dt                      = 0.002
nsteps                  = 20000000              ; 40 ns
nstlog                  = 1000
nstxout                 = 0
nstxout-compressed      = 1000
nstvout                 = 0
nstfout                 = 0
nstcalcenergy           = 100
nstenergy               = 1000
;
cutoff-scheme           = Verlet
nstlist                 = 20
rlist                   = 1.2
coulombtype             = pme
rcoulomb                = 1.2
vdwtype                 = Cut-off
vdw-modifier            = Force-switch
rvdw_switch             = 1.0
rvdw                    = 1.2
;
tcoupl                  = Nose-Hoover
tc_grps                 = tmem  bmem    SOL_ION
tau_t                   = 1.0    1.0    1.0
ref_t                   = 303.15 303.15 303.15
;
pcoupl                  = Parrinello-Rahman
pcoupltype              = semiisotropic
tau_p                   = 5.0
compressibility         = 4.5e-5  4.5e-5
ref_p                   = 1.0     1.0
;
constraints             = h-bonds
constraint_algorithm    = LINCS
continuation            = no
gen_vel                 = yes
gen_temp                = 303.15
gen_seed                = -1
;
nstcomm                 = 100
comm_mode               = linear
comm_grps               = upper   lower
;
;;;;;;;;   CompEL Section   ;;;;;;;;
```

```
;
swapcoords = Z


swap-frequency = 100
split-group0 = tmem
split-group1 = bmem
massw-split0 = no
massw-split1 = no
solvent-group = TIP3
coupl-steps = 10
iontypes = 2
iontype0-name = SOD
iontype0-in-A  =  -1     ;  requested  number  of  Na  ions  in
compartment A
iontype0-in-B  =  -1     ;  requested  number  of  Na  ions  in
compartment B
iontype1-name = CLA
iontype1-in-A = -1   ; -1 means: use the number of ions
iontype1-in-B = -1   ;           as found at time step 0
bulk-offsetA = 0.0
bulk-offsetB = 0.0
cyl0-r    = 5.0
cyl0-up   = 0.75
cyl0-down = 0.75
cyl1-r    = 5.0
cyl1-up   = 0.75
cyl1-down = 0.75
threshold = 1
```

Center of mass motion removal

The double membrane box geometry introduces one artifact completely unrelated to the CompEL algorithm having to do with center of mass motion removal. When three center of mass motion removal groups were selected (one for each membrane and one for the solvent and ions), GROMACS seemed to compensate for motion in one membrane in the x-y plane by imparting equal and opposite motion to the other membrane, possibly creating un-natural shear at the membrane-water boundary. To mitigate this effect, the top membrane was grouped with all the solvent and ions to produce the first motion removal group leaving only the bottom membrane to become the second group.

Production run history plots

In post processing, pore core water plots were produced for all 144 production runs, permitting a quick visual check for any post processing anomalies. For each charge imbalance, a representative set of the plots for the two membrane pores (channels 0 & 1) are shown below. (**Figs. S8-S13**)

The core water plot (purple) shows the water molecule count in the central 1.5 nm of the pore. The throat (abbr. thrt & t) water plot (evergreen) shows the water molecule count in the central 3.0 nm of the pore. Note that the throat water molecules are plotted at half scale. To accurately read the water molecules value represented by the evergreen trace, the value from the left-hand scale must be doubled.

The key will also include the number of ion permeations if there are any. Ion permeation events times are indicated by the vertical red and green lines at the bottom of the graph. The green lines (which are plotted after the red lines) are plotted at half height so that an underlying red line will not be totally obscured. Multiple red or green lines have occasionally obscured another of the same color, consequently, a few plots will have a larger number of ions shown in the key than the corresponding vertical lines.

Below the key are the simulation charge imbalance followed by the average pore and throat water molecule counts. Following this are two sets of numbers, first for the average pore and throat Cation permeation times and second set is the same pair for Anions. These represent the average, total time each ion spends in the throat and pore regions for each permeation during the simulation. The units are ns/permeation.

Of the first four plots, notice the two where the purple trace decays at a point to a value of zero. These illustrate the data from simulations where a membrane pore has totally disintegrated.

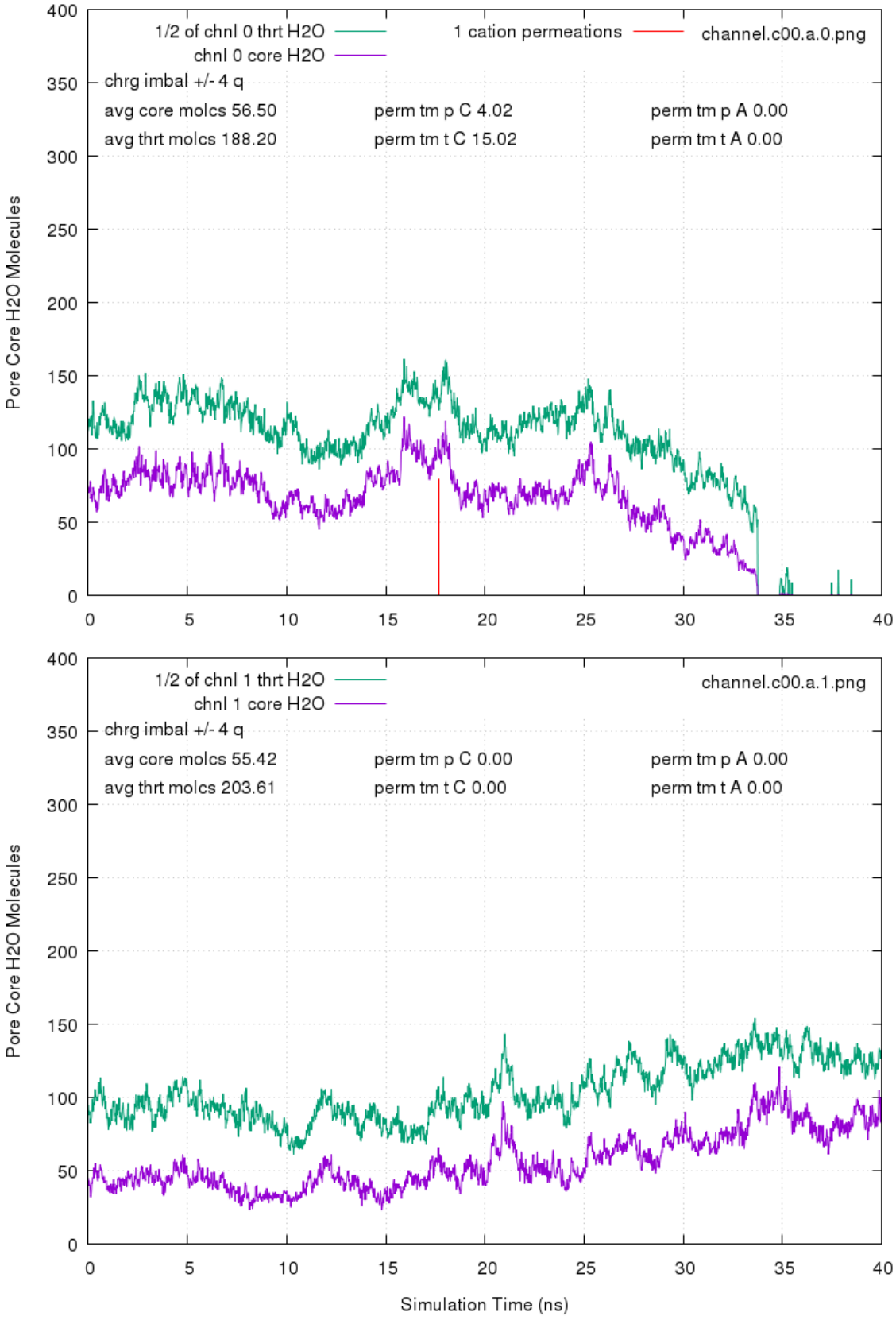


Fig. S8: *Plots for production run history with +/- 4 e^- charge imbalance.*

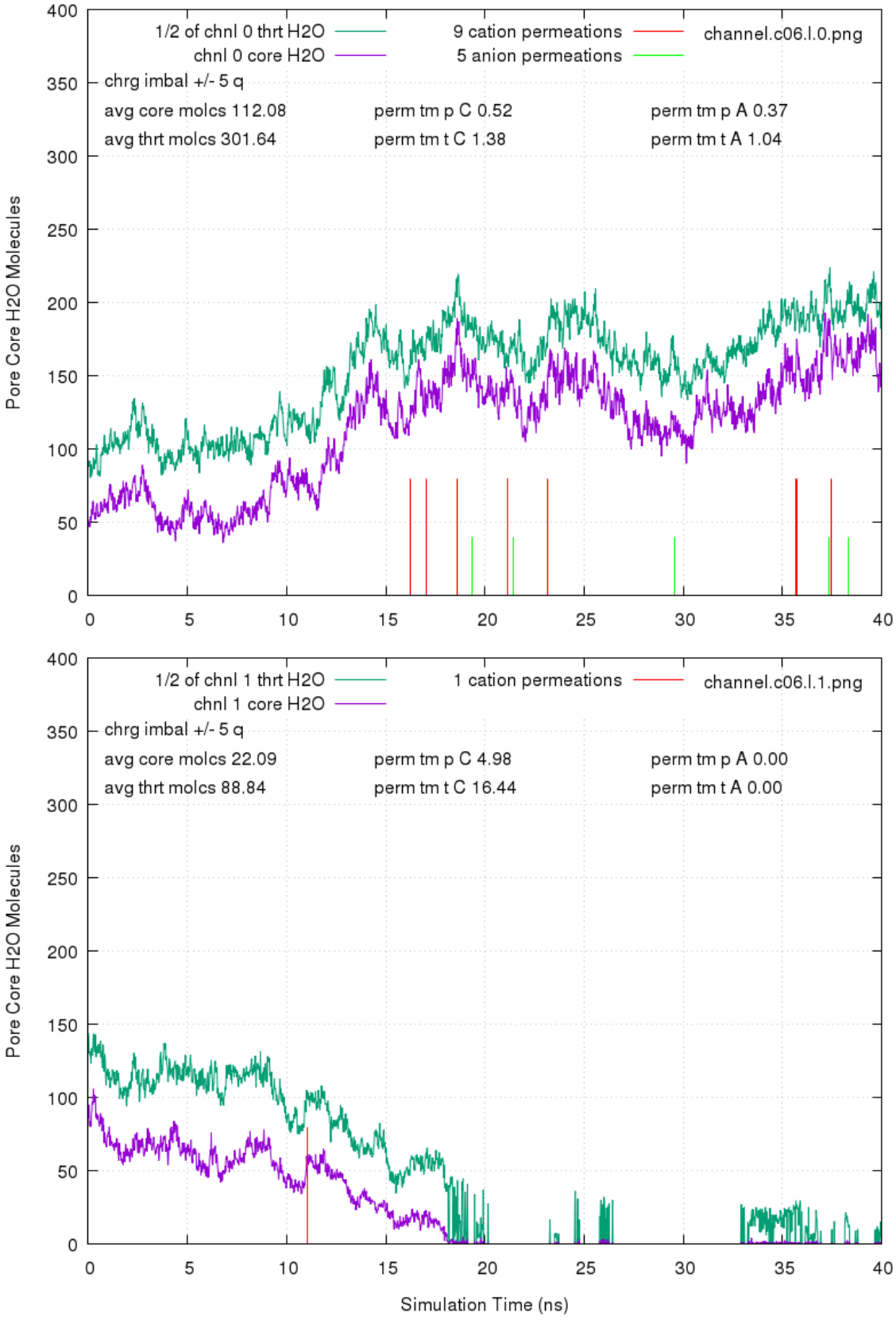


Fig. S9: *Plots for production run history with +/- 5 e^- charge imbalance.*

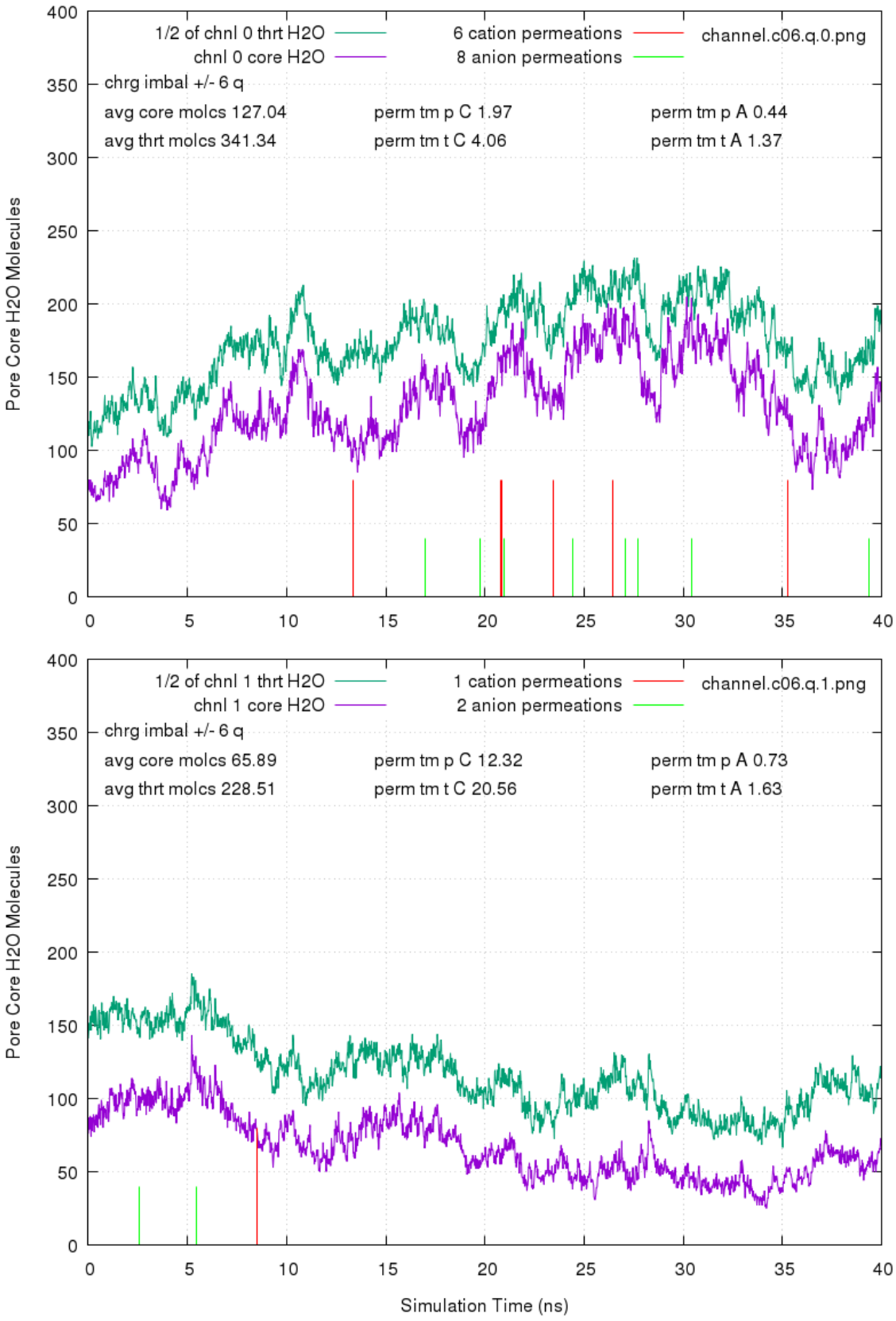


Fig. S10: *Plots for production run history with +/- 6 e^- charge imbalance.*

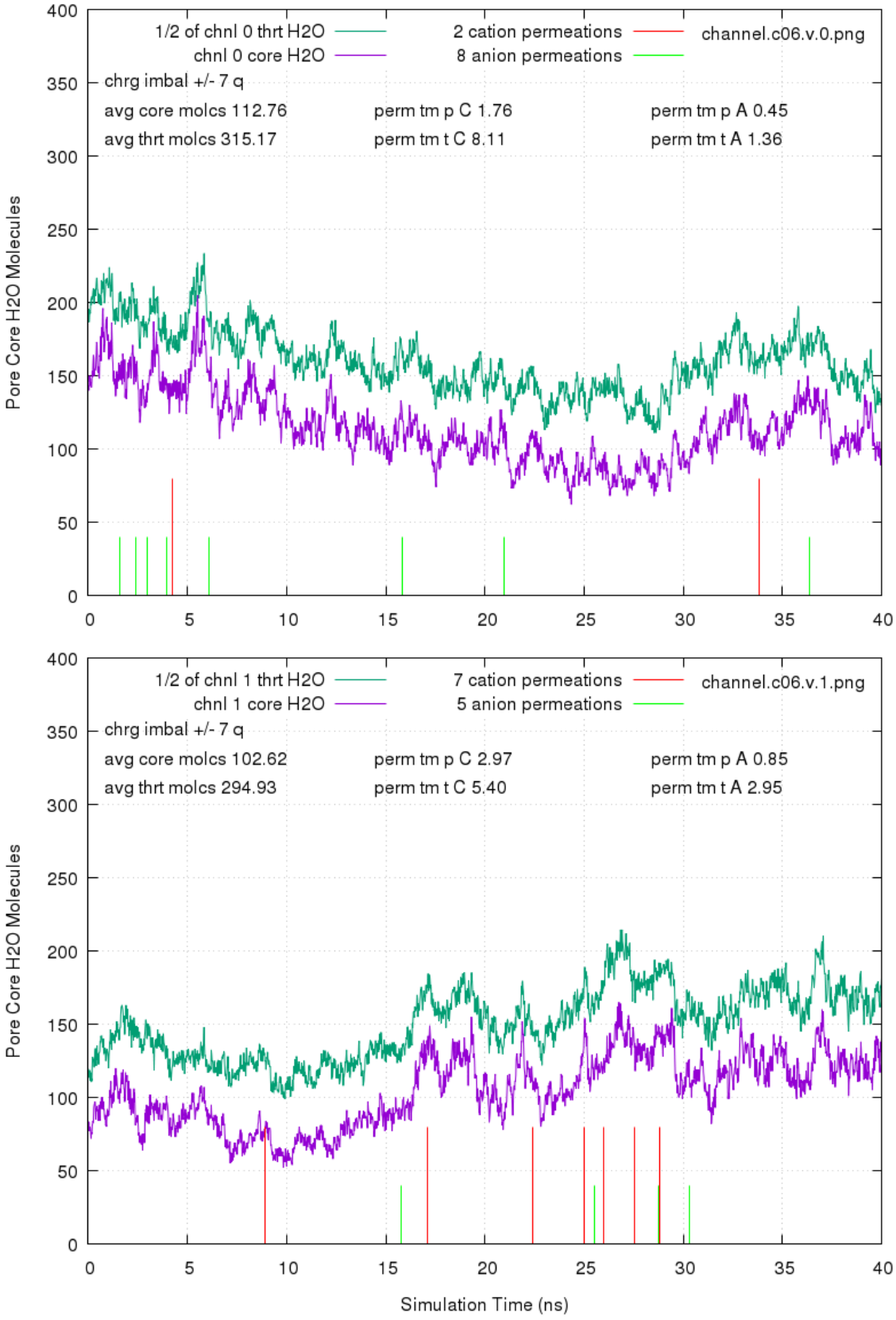


Fig. S11: *Plots for production run history with +/- 7 e^- charge imbalance.*

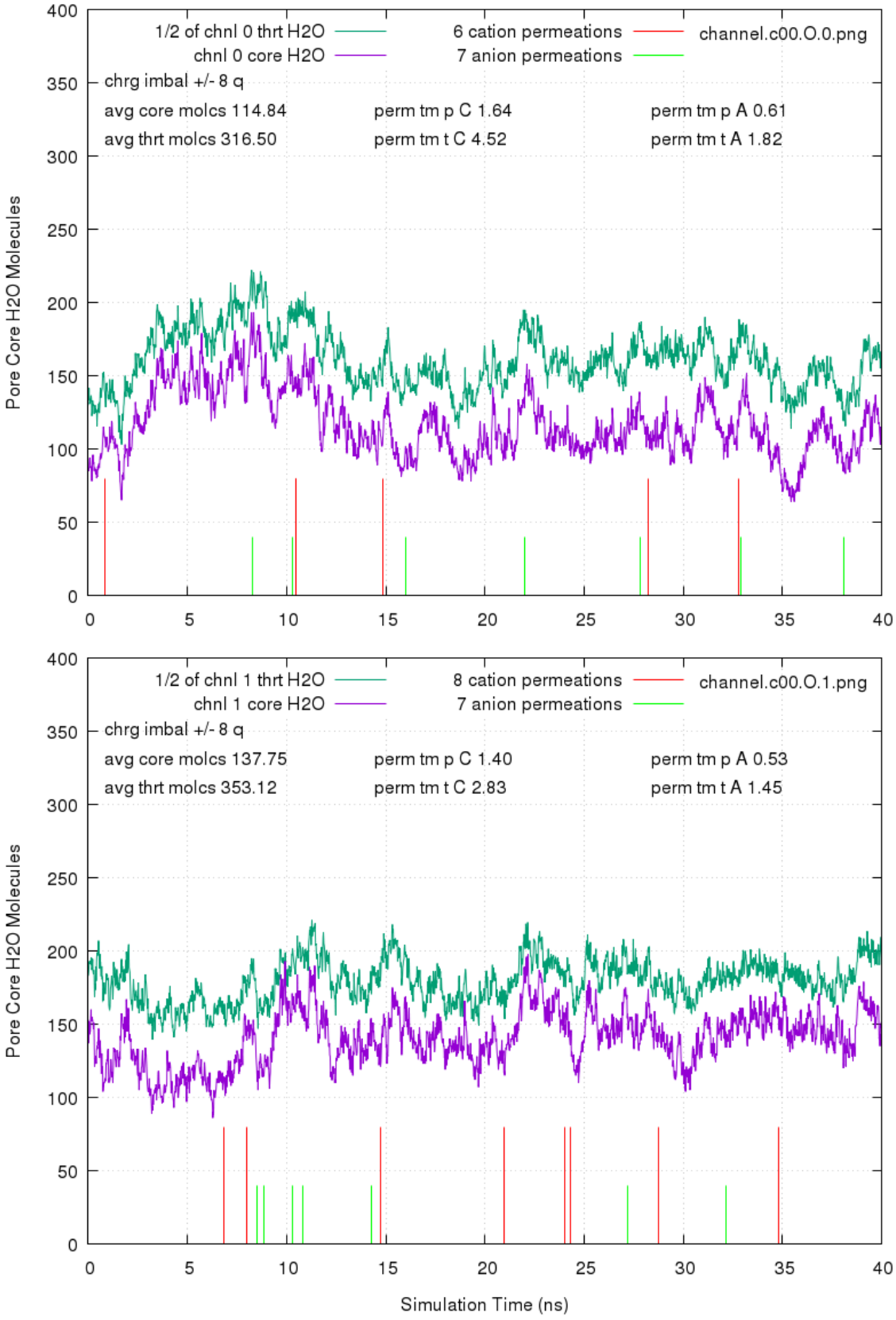


Fig. S12: *Plots for production run history with +/- 8 e^- charge imbalance.*

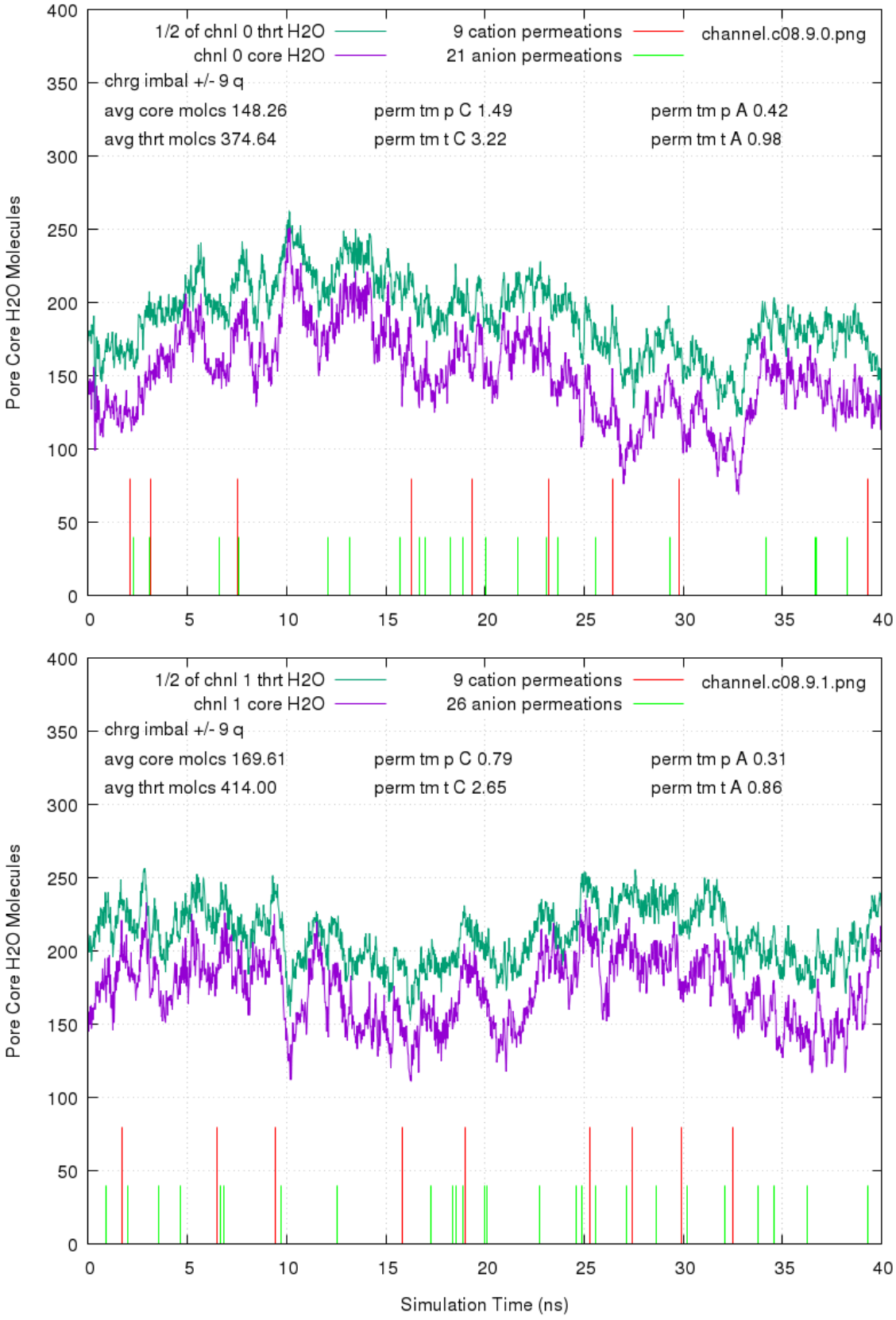


Fig. S13: *Plots for production run history with +/- 9 e^- charge imbalance.*

Post processing issues

If pores drift across periodic boundaries during a simulation, determining the coordinates of the exact center of the pore in every frame becomes problematic. There are instances where the GROMACS post processing utilities are unable to do so and will return a value of NaN (not a number). This must be corrected before any reliable counting of pore water molecules can be conducted. The GROMACS *trjconv* utility has a periodic boundary processing option -cluster that may be able to handle every situation but takes considerably more trouble to implement and time to perform.

For the problem of drifting pores, a much easier and more efficient method of correcting this problem was developed and at its state of development handles completely automatically about 98% of the problem trajectories. Simple manual tweaking of the translation values fixes the other 2%. It requires the use of three scripts over four steps, all of which have been automated. Visual inspection of the simulation trajectory history plot (which is performed anyway), will reveal those trajectories that need manual tweaking.

Links for GitHub access to two of the scripts are listed below (the third just plots the data). The first is named *select.* It counts water molecules and dumps trajectory coordinates. The second is named *shift* which uses data produced by a *select* routine to performs the necessary translations to keep pores from crossing periodic boundaries. To determine if a top membrane pore needs adjustment, the steps are:

1) Use script *select* to tabulate the pore center coordinates throughout the simulation into a file called twatc.xvg. The command is: source select twatc

2) Use script *shift* to translate the trajectory, if necessary, creating a file called trajout0.xtc. The source trajectory file for this operation is called trajout.xtc. If the translation is necessary and file trajout0.xtc is created, all subsequent post processing operations for channel 0 data will preferentially select trajout0.xtc over file trajout.xtc. The original file, trajout.xtc may not need any translation to extract data for channel 1 and must be retained. (Since data extraction is an activity unique to each pore, the same processes are applied to channel 1, and if necessary, will produce a preferentially selected in subsequent steps file trajout1.xtc.)

3) Use script *select* to tabulate the water molecules in the top (channel 0) pore on a frame by frame basis into file twat.xvg. The command is: source select twat. If a pore still comes too close to the periodic boundaries, some of the values in file twat.xvg will be NaN.

4) Plot the water molecules in the pore over the duration of the simulation. If plotted in *gnuplot* with a line, the NaN values will cause the line to jump from the last good value down to the y-axis zero value until the next real number occurs in the file. These are immediately obvious on visual inspection and comprise the roughly 2% of the trajectories that need some manual tweaking. The manual tweaking process is described later.

Listing of scripts

The first script *select* has measuring and counting routines. They are selected by the first and only parameter passed to the script. The parameter choices are present in the script as the case options. For adjusting the channel 0 trajectory frames, only the twatc and twat options are needed. The scripts are written in *Bash* (bourne again shell).

Script: ***https://github.com/JPatrickBrian/Redstone-Engineering/blob/master/select..***

The second script examines file twatc.xvg (and bwatc.xvg for channel 1) to determine if/by how much a trajectory file needs to be translated to keep a pore off the periodic boundaries. It does so by creating histograms of the x and y coordinates of the center of the pore throughout the simulation and attempts to position the center of the most significant gaps in the histogram coordinates on the simulation periodic boundaries.

Script: ***https://github.com/JPatrickBrian/Redstone-Engineering/blob/master/shift.***

Manually tweaking a trajectory translation

If a pore water molecule count file (twat.xvg) still contains nan values after being translated, it requires manual tweaking. This occurs when the pore migrates over the entire x or y dimension (or nearly so) during the simulation. The image below (**Fig. S14**) shows an example of how this migration of the pore center over the periodic boundaries compromises the counting of pore water molecules. Note the data dropout after 33 ns is due to pore collapse and not conflict with periodic boundaries.

The two following commands produce the histogram data for the center of the pore over the *x* and *y* dimensions. Following that is the histogram data from the first command (*x*-axis). Ideally, the smallest counts (left column) at the top and bottom of the list are consistent with a unimodal distribution (this example satisfies that). The other beneficial distribution of the data is nearly equal size gaps between 0 and the first coordinate (right column) and between the last coordinate and the maximum axis dimension (this example is NOT like that since the maximum axis coordinate is about 7.3). If this example had a problem, the fix would likely be to translate the frames ~ (-2.0 nm) along the x axis. Most

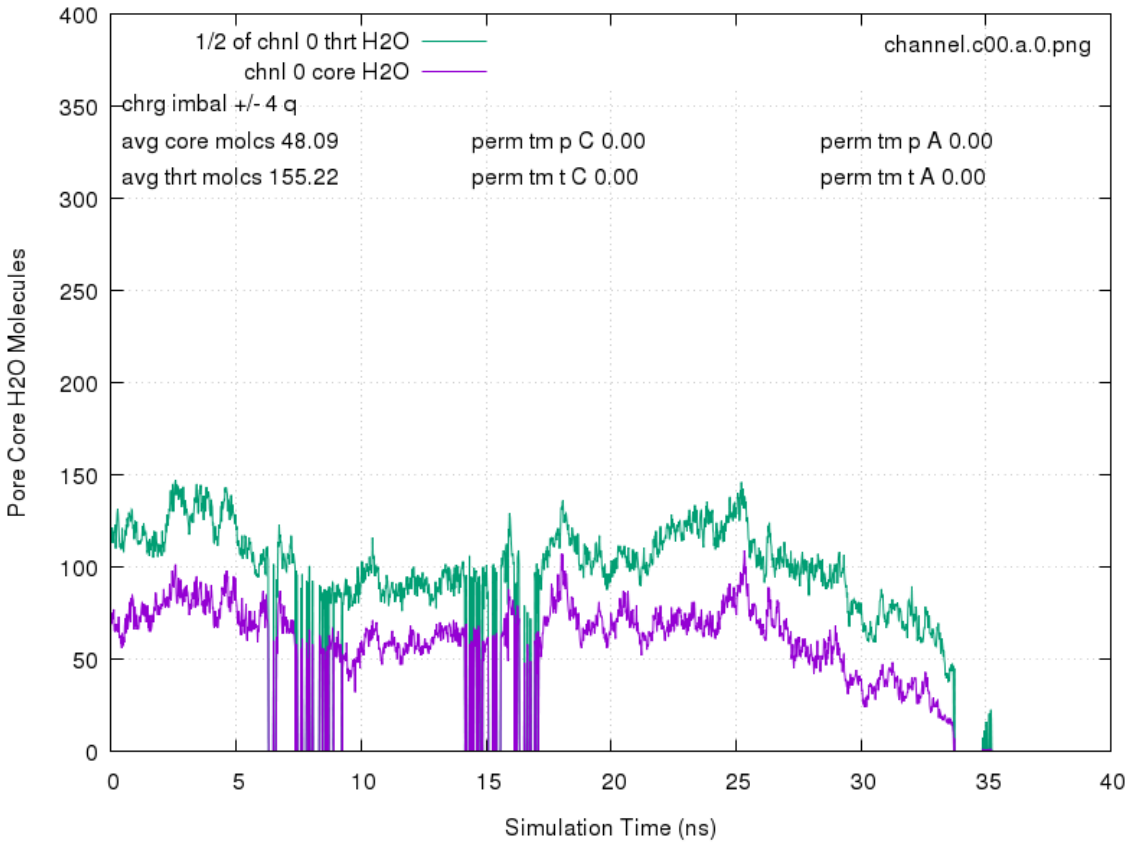


Fig. S14: *Signature pattern of water molecule counts when the geometric center of a pore in a frame of data has an indeterminate (NaN) value.*

commonly, a trajectory translation only needs to be tweaked just a few ångströms. Occasionally, the histogram can appear to have two gaps and the solution is to translate the frames for the other gap to line up with the periodic boundaries.

```
FR=`sed -n '/     0.000/{=;q}' twatc.xvg`;tail -n +${FR} twatc.xvg  | awk '{printf
"%1.1f\n", $2}' | sort -k1,1n | uniq -c | grep -v nan
FR=`sed -n '/     0.000/{=;q}' twatc.xvg`;tail -n +${FR} twatc.xvg  | awk '{printf
"%1.1f\n", $3}' | sort -k1,1n | uniq -c | grep -v nan
```

```
      1 4.0
      9 4.1
     14 4.2
     43 4.3
    107 4.4
     99 4.5
     90 4.6
     62 4.7
     95 4.8
    107 4.9
```

```
150 5.0
124 5.1
 73 5.2
 52 5.3
 64 5.4
 69 5.5
 33 5.6
 43 5.7
 25 5.8
 28 5.9
 31 6.0
 48 6.1
 77 6.2
104 6.3
 94 6.4
 99 6.5
 82 6.6
 67 6.7
 59 6.8
 46 6.9
  6 7.0
```

Complex query explanation

Below is a copy of the query used to count the water molecules present in the core of a membrane pore on a frame by frame basis despite bulk motions of the system caused by other factors. Note that a pore must not drift across a periodic boundary in a trajectory submitted to this query. Trajectories where that is not the case must have other preprocessing to correct that before processing with this step. Only the -select command string is explained as the other parameters are sufficiently covered in the standard GROMACS documentation. Although ostensibly the syntax of the select string is also explained, without an example of a non-trivial application of its capabilities, realizing its full power may require a proficiency with structured query language statements uncommon beyond the domain of database experts.

```
gmx select -f traj.trr -s topol.tpr -os bwater.xvg -rmpbc -pbc -select
'
lhgrps = resname "DOPC" and (name "P.*" or name "N.*" or name "O." or name
"O[12].");
zm     = z of cog of lhgrps;
wcog   = cog of resname TIP3 and ( (z+.45) > zm and (z-.45) < zm );
xw     = x of wcog;
yw     = y of wcog;
```

```
zw     = z of wcog;
radius = ((x-xw)^2+(y-yw)^2)^0.5;
lcog   = cog of lhgrps and ( (z+3) > zw and (z-3) < zw ) and radius < 1.5;
zc     = z of lcog;
dyn_mol_cog of resname TIP3 and ( (z+0.75) > zc and (z-0.75) < zc ) and radius <
1.5
'
```

The last parameter of the select command is the -select flag selection command string and is enclosed in single quotes to prevent the operating system shell from interpreting and processing any text between the quotes. Explaining the statement line by line:

```
lhgrps = resname "DOPC" and (name "P.*" or name "N.*" or
name "O." or name "O[12].");
```

`lhgrps` stands for *lipid head groups*. Here it exists as what GROMACS calls a *variable* and in this syntax, it is equated to an *expression.*

Variables are just names used to help us remember what they represent, because when used later, they become a shorthand method of writing the much longer and more complicated *expression.* Every line beginning with an "=" sign (the first 9 of the selection string) are simply equating *variables* to *expressions*. Each *expression* is a short and relatively simple use of an element of the selection syntax described at

http://manual.gromacs.org/documentation/2018/onlinehelp/selections.html.

In fact, this entire string only executes one single element of the selection syntax, which is specified on the very last line. By using *variables* with functionally descriptive monikers, a much simpler written statement causes the invocation of a much more complex fully detailed statement that precisely articulates every condition and limitation the candidate water molecules must satisfy to be included in the result. The problem non-

database experts encounter is that the fully detailed statement is so seemingly complicated as to appear to be unreadable. Even database experts find it daunting and would use variables as well because it increases readability, maintainability and statement writing efficiency.

If this seems arbitrarily and unnecessarily complex, it is because structured query language statements, all types including select, must be executed on databases one at a time in manner that would be analogous to a run-on sentence, paragraph and chapter if need because there can be one and only one period at the end of the instruction. They cannot be "split" into two sentences. If a statement hasn't finished detailing every aspect of the selection criteria into one sentence, it must continue until it has. Technically, this requirement comes from the necessity for databases to maintain something called "transactional integrity" which is beyond the scope of this explanation. And our trajectory file which our data comes from is not a database.

Nevertheless, it is very common for select commands to access and merge data from files that are not relational database files with data from files that are. To maintain functional compatibility between databases created by different companies, structured query language statements must conform to a standard known as ODBC. ODBC stands for open database connectivity. Whether the GROMACS software has been developed with some future interactivity in mind or not, and whether it is ODBC compliant or not is not clear to us. But making statement mistakes when trying to fashion complex queries give zero feedback about the nature or location of the problem other than "syntax error". To deal with this at the user end, it means a way is needed to understandably create a complex

selection criterion, in a fashion that can be simply read and understood. One method is to use *variables*.

The select element syntax allows the selection of atoms, molecules and residues using three different types of *expressions,* numerical, atom and position. Think of it like this: Correctly written *expressions* define a specific subset of *atoms,* sets or subsets of specific *atoms* are located at a specific set of *positions* and each *position* in that set is comprised of three x,y and z axis coordinate *numerical* values. Note there are expressions that can define sets or subsets of atoms and numerical values as well. By appropriate use of applicable properties of each of these types of sets, or subsets of them, as suits the purpose, a variety of data element types can be extracted from a trajectory file.

In the first selection string line, *variable* `lhgrps` is equated to an *atom_expression*. Henceforth, `lhgrps` becomes a *variable* representing a specific set of atoms. Which ones? For starters, any atom whose parent residue name is "DOPC" but with an additional stipulation. The additional stipulation is added with the "and" and the statement between the following parentheses. Between the "()", the new criteria are added that relates to each atom's name, but by using a scripting convention called regular expressions, used here because they allow the creation of "a complex selection criterion, in a fashion that can be simply read and understood". With regular expressions, that can only happen after learning the conventions and syntax for writing and implementing them. The reality is that without a decent working knowledge of regular expressions, the capacity to implement the FULL power of dynamic select statements will remain substantially out of reach. The scope of regular expression capabilities is enormous, is fully documented elsewhere, and is vastly

beyond the scope of this explanation. However, the statement used in this script will be explained.

```
(name "P.*" or name "N.*"
```

Selects any atom whose name starts with the letter P or N, irrespective of the total name length.

```
or name "O."
```

Selects any atom whose name starts with the letter O and has a total name length of exactly 2 characters.

```
or name "O[12].");
```

Selects any atom whose name starts with the letter O, whose second name character is the numeral 1 or 2 and has a total name length of exactly 3 characters. The semicolon at the end of the line denotes that the variable *expression* ends here.

Given a fully labelled drawing of a DOPC molecule and the explanation above, one has the information to be able to determine the number of atoms in the set represented by the variable `lhgrps` for a single molecule.

```
zm = z of cog of lhgrps;
```

This is a *numerical* expression. The set of atoms represented by `lhgrps` has a center of geometry (as opposed to its center of mass). The variable `zm` represents the numerical value of the z-axis coordinate of this center of geometry.

```
wcog = cog of resname TIP3 and ((z+.45)>zm and (z-
.45)<zm);
```

This is a *position* expression. `TIP3` water molecules whose z-axis center of geometry coordinate value is within +/- 0.45 nm of the value of `zm` are a subset of all TIP3P water molecules. The variable `wcog` represents the *position* of the center of geometry of this subset of atoms.

```
xw      = x of wcog;
yw      = y of wcog;
zw      = z of wcog;
```

These are *numerical* expressions. The variables `xw, yw` and `zw` are given the values of the x,y and z axis coordinates of the *position* represented by `wcog.`

```
radius = ((x-xw)^2+(y-yw)^2)^0.5;
```

This is a *numerical* expression and should be self-explanatory. The interesting thing is that the *expression* is not linked to any specific *position* or sets of *positions* without which a logical source for the values of x and y will be undefined. This linking can be established in the statements where the *expression* `radius` is invoked to act upon a *position* or set of *positions*. This capability is elegant and powerful, and the full implications of this ability require some rumination to fully appreciate.

```
lcog = cog of lhgrps and ((z+3)>zw and (z-3)<zw) and
radius < 1.5;
```

This is a *position* expression. The set of atoms represented by `lhgrps` whose z-axis coordinate value is within +/- 3 nm of the value of variable `zw` and are closer horizontally to the x,y coordinates of position `wcog` than 2 nm are a subset of `lhgrps.` `lcog` represents the position of the center of geometry of this subset. Due to its placement

in the *expression*, the variable `radius` gets the x and y coordinates it needs from the *positions* of the *atoms* in the set represented by `lhgrps`.

```
zc = z of lcog;
```

Numerical expression.

Now all the variables needed have been defined to be able to write the statement in a simplified fashion to extract the data sought, specifically the number of water molecules in this frame that satisfy the selection criteria. So far, the actions of the variable expressions have been explained, but not the geometrical significance of these expressions in relation to the biostructure in which it will count a very specific subset of the water molecules.

Initially the z coordinate of the center of the membrane of interest is needed. `lhgrps` defines the set of atoms used to do this and `zm` gets this from the collective center of geometry of this set. `wcog` gets the center of geometry of water molecules present in a thin horizontal slice centered on `zm`. In normal cases, this should be very close to the center of the pore, particularly on the x,y plane. Because larger membranes exhibit waves and undulations, pores move up and down with these undulations. To move with these undulations, we need the z coordinate of the center of geometry of the lipid headgroups in the immediate vicinity of the pore. This will allow the query to keep track of the pore's center in the z direction as it moves up and down with the waves and undulations. Variables `lcog` and `zc` achieve this aim.

With this foundation laid, the GROMACS *select* statement can be executed. In the statement options, option flag `-os` is specified which writes a file recording the number of entities that satisfy the selection criteria in each frame, in this instance the number of water

molecules in the core of the pore. The statement that brings this about is the final line of the select string.

```
dyn_mol_cog of resname TIP3 and ((z+0.75)>zc and (z-
0.75)<zc) and radius < 1.5
```

This is a position expression. The `dyn_mol_cog` specifies to select center of geometry positions (`_cog`), for molecules (`_mol`), on a frame by frame basis (`dyn`). The number of those positions that satisfy the selection criteria will be recorded into the file specified after the `-os` option flag. The selection criteria are all water molecules (`resname TIP3`) whose elevations are within +/- 0.75 nm of the z coordinate of the center of the membrane in the immediate vicinity of the pore and whose horizontal distance from that x,y center is less than 1.5 nm.

The presence of the `dyn` modifier at the beginning of the statement causes all the dependent variables it references to also be evaluated on a frame by frame basis. The result is a completely accurate count of the water molecules that satisfy the criteria, unaffected by the bulk motions of the system.

It is one thing to read a description of how and why a statement achieves its stated purpose. It is another to acquire enough of the skill to be able to write the needed statements to do novel data extraction from other trajectories to address the next research analysis problem that crops up. In our experience, that has only developed through a commitment to learn the necessary skills. Hopefully this example has demonstrated the value of the available tools and enough insight to make others' attempts at learning them successful.

Supporting Material – Section 2

These scripts and processes were developed over a two-year period in support of my doctoral research. Their development was an evolutionary process. None of this work was imagined or planned in the beginning. Unexpected results were encountered, and processes were developed to better measure and verify the initial indications. This process is nowhere close to being complete. There is much more work to do to better understand what has been discovered, and to bring more utility to the work that has been completed so far. It is beyond the scope of this document to explain everything that is taking place. The purpose of this document is to outline the sequence of steps required to implement these tools on simulation trajectories to produce the resulting analysis. The first step to understanding this work would be to carefully read the **complex query explanation** in the Section 1 Supporting Material (pg. 137). This explains the syntax used in the scripts select. and shift. These enable a very accurate count of water molecules in a membrane pore on a frame by frame basis. They also prepare the trajectories for the following processing.

For the detailed explanation of steps 11-21 in the flowchart in **Fig. 9**, a listing of a master script showing the sequence of commands follows with interspersed comments (highlighted blue) as needed. All the routines are written in *Bash*, calling standard Unix or GNU utilities. Note that many of the *AWK* routines have multiple index arrays, so you must use *AWK* version 3 or above to run them. **Fig. S15** is a screen capture of the beginning of my GitHub repository where a full listing of every routine can be examined or downloaded. Its web address is:

https://github.com/JpatrickBrian/Redstone-Engineering

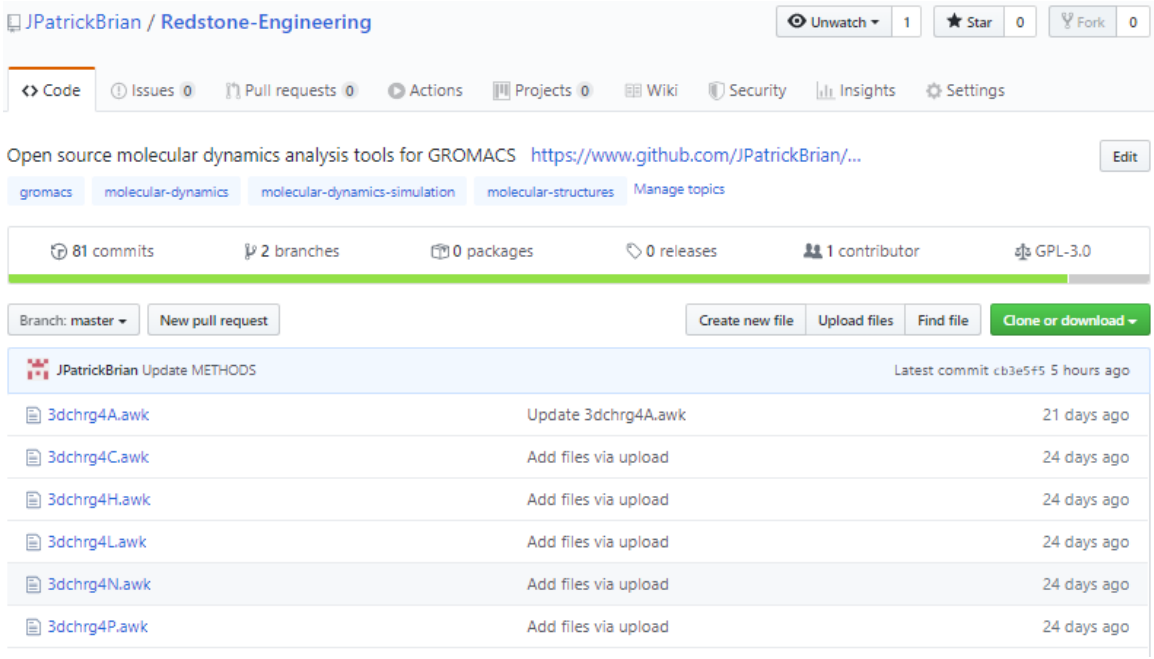


Fig. S15: *Screen capture of top of home page of my GitHub repository for making code freely available to the research community at large.*

These following steps run the initial simulation, then do a quick (cheap computationally) analysis to create some plots that can be reviewed to make sure everything looks OK. For simplicity and completeness every command including the generation of the initial trajectory files is shown.

```
# runs simulation
gmx mdrun -cpi
# makes molecules whole with respect to periodic boundaries
gmx trjconv -f traj_comp.xtc -o trajout.xtc -pbc mol -skip 10 <
          <(echo -e "System\n")
# crudely, but quickly counts water molecules in top membrane
gmx select -f trajout.xtc -n index.ndx -s topol.tpr -os
          waters.xvg -select 'dyn_mol_com of resname TIP3
          and ((z+.75) > (z of com of (group DOPC and name P))
          and (z-.75) < (z of com of (group DOPC and name P)))'
# crudely, but quickly counts water molecules in bottom membrane
gmx select -f trajout.xtc -n index.ndx -s topol.tpr -os
          watersb.xvg -select 'dyn_mol_com of resname TIP3
          and ((z+.75) > (z of com of (group D2PC and name P))
          and (z-.75) < (z of com of (group D2PC and name P)))'
# tabulates totals of ion crossing events for each pore
source ../jobs/cross.
# plots the initial results, validation of no extreme problems
```

```
gnuplot -c ../jobs/cross
#
# The next steps make very accurate counts of membrane pore water
molecules.
source ../jobs/select. twatc
source ../jobs/select. bwatc
source ../jobs/shift. run
source ../jobs/count.
```

Now for each membrane pore (each of two membranes in the sim have one pore), a trajectory is produced in which the center of the pore is translated to the exact center of the simulation box. From these trajectories, a set of coordinates for every atom in every frame of the simulation is created (coord.xvg). This file is used by script *3dchrg4A.awk* to produce the composite electric charge map of the simulation volume elements. Finally, this file is processed by *Bfintegr8.awk* to produce the electric field and potential maps. In this example, channel 0 (top) is being processed. #!NOTE: Since my work has focused at this point on the analysis of only one of the two pores in any single simulation, the file names would conflict if both pores were analyzed simultaneously.

```
# centers a pore in the exact center of the simulation box,
one parameter is passed, 0 - center top pore, 1 - center
bottom pore
source ../jobs/center. 0
```

This is a rather complex routine, so we will list its details. The comments (following #) in blue are not contained in the listings but added here for clarity of the dissertation.

```
#!/bin/bash -l
#================================================
# center.    ### script for centering pores   ###
#            ### in trajectories              ###
# parameters
# $1         channel to center
#   options: 0 - channel 0 in the top membrane
#            1 - channel 1 in the bottom membrane
#________________________________________________
```

```
case ${1} in
# this section handles the top channel
 0)
# trajout.xtc is the original trajectory file, if no
shifting was required to count the water molecules, we can
use that file. If shifting was required for the top
channel, that will exist as trajout0.xtc. Pick accordingly
   if [ -s trajout0.xtc ]; then
     tfyle="trajout0.xtc"
   else
     tfyle="trajout.xtc"
   fi
# if the box dimensions file doesn't exist, create it
   if ! ( [ -a boxcntr.xvg ] ); then
     source ../jobs/boxdims.
   fi
# load the box center dimensions
   declare -a boxcntr
   boxcntr=($(cat boxcntr.xvg))
# we must make sure the pore center coordinates file
matches the most recent version of the trajectory file
   if [ "${tfyle}" == "trajout0.xtc" -a ${tfyle} -nt
"twatc.xvg" ]; then
     source ../jobs/roll. twatc.xvg
     source ../jobs/select. twatc
   fi
# we need to regenerate the pore center coordinates, do so
   if ! ( [ -s twatc.xvg ] ); then source ../jobs/select.
twatc; fi
# calculate the translation vectors for every frame
   s ../jobs/trans. ${boxcntr[0]} ${boxcntr[1]}
${boxcntr[2]} twatc.xvg
# translate the frames
   gmx trjconv -f ${tfyle} -o /local/scratch/temp.gro -pbc
mol -sep -exec 'source ../jobs/translate.' < <(echo 0)
# if we are about to overwrite the centered trajectory
file, give it a unique index number instead to preserve
genealogy
   if [ -s "trajout0c.xtc" ]; then source ../jobs/roll.
trajout0c.xtc; fi
# concatenate the centered frames into a new trajectory
file
   gmx trjcat -f /local/scratch/tump{0..2000}.xtc -o
trajout0c.xtc
# remove the no longer needed individual trajectory frames
   rm /local/scratch/tump*.xtc
```

```
# in the Jaeger lab cluster, we compile our GROMACS to run
in single precision. To perfectly center the pore requires
a second pass identical to the first. The same comments
would apply
   tfyle="trajout0c.xtc"
   source ../jobs/select. twatc
   s ../jobs/trans. ${boxcntr[0]} ${boxcntr[1]}
${boxcntr[2]} twatc.xvg
   gmx trjconv -f ${tfyle} -o /local/scratch/temp.gro -pbc
mol -sep -exec 'source ../jobs/translate.' < <(echo 0)
   source ../jobs/roll. trajout0c.xtc
   gmx trjcat -f /local/scratch/tump{0..2000}.xtc -o
trajout0c.xtc
   rm /local/scratch/tump*.xtc
   ;;
# this section handles the bottom channel. Other than
different file names and parameter options, it is a carbon
copy of the first section. The same comments would apply
 1)
   if [ -s trajout1.xtc ]; then
     bfyle="trajout1.xtc"
   else
     bfyle="trajout.xtc"
   fi

   if ! ( [ -a boxcntr.xvg ] ); then
     source ../jobs/boxdims.
   fi

   declare -a boxcntr
   boxcntr=($(cat boxcntr.xvg))

   if [ "${bfyle}" == "trajout1.xtc" -a ${bfyle} -nt
"bwatc.xvg" ]; then
     source ../jobs/roll. bwatc.xvg
     source ../jobs/select. bwatc
   fi

   if ! ( [ -s bwatc.xvg ] ); then source ../jobs/select.
bwatc; fi

   source ../jobs/trans. ${boxcntr[0]} ${boxcntr[1]}
${boxcntr[2]} bwatc.xvg

   gmx trjconv -f ${bfyle} -o /local/scratch/temp.gro -pbc
mol -sep -exec 'source ../jobs/translate.' < <(echo 0)
```

```
   if [ -s "trajout1c.xtc" ]; then source ../jobs/roll.
trajout1c.xtc; fi

   gmx trjcat -f /local/scratch/tump{0..2000}.xtc -o
trajout1c.xtc

   rm /local/scratch/tump*.xtc

   bfyle="trajout1c.xtc"
   source ../jobs/select. bwatc
   source ../jobs/trans. ${boxcntr[0]} ${boxcntr[1]}
${boxcntr[2]} bwatc.xvg
   gmx trjconv -f ${bfyle} -o /local/scratch/temp.gro -pbc
mol -sep -exec 'source ../jobs/translate.' < <(echo 0)
   source ../jobs/roll. trajout1c.xtc
   gmx trjcat -f /local/scratch/tump{0..2000}.xtc -o
trajout1c.xtc
   rm /local/scratch/tump*.xtc
   ;;

 *)
   echo center : ${1} is not a valid case
   ;;
esac
```

Now that the trajectory has been centered, a set of coordinates for every atom in every frame of the simulation is created (coord.xvg), no small task since its size is only 61 GB. This file is used by script *3dchrg4A.awk* to produce the composite electric charge map of the simulation volume elements. The other routines *3dcharg4H.awk* through *3dchrg4S.awk* calculate the same for individual moieties of interest in the system. Finally, these files are processed by *Bfintegr8.awk* to produce the electric field and potential maps (by *gnuplot* routine plfldmaps). In this example, channel 0 (top) is being processed. #!NOTE: Since my work has focused at this point on the analysis of only one of the two pores in any single simulation, the file names would conflict if both pores were analyzed simultaneously.

```
gmx trajectory -f trajout0c.xtc -ox coord.xvg -n index.ndx
< <(echo -e 'System\n')
source ../jobs/3dchrg4A.awk 50 chrgs3.xvg chrg5A.xvg 1 4 10
11
source ../jobs/3dchrg4H.awk 50 chrgs3.xvg chrg5H.xvg 1 4 10
11
source ../jobs/3dchrg4N.awk 50 chrgs3.xvg chrg5N.xvg 1 4 10
11
source ../jobs/3dchrg4C.awk 50 chrgs3.xvg chrg5C.xvg 1 4 10
11
source ../jobs/3dchrg4P.awk 50 chrgs3.xvg chrg5P.xvg 1 4 10
11
source ../jobs/3dchrg4W.awk 50 chrgs3.xvg chrg5W.xvg 1 4 10
11
source ../jobs/3dchrg4R.awk 50 chrgs3.xvg chrg5R.xvg 1 4 10
11
source ../jobs/3dchrg4T.awk 50 chrgs3.xvg chrg5T.xvg 1 4 10
11
source ../jobs/3dchrg4Y.awk 50 chrgs3.xvg chrg5Y.xvg 1 4 10
11
source ../jobs/3dchrg4L.awk 50 chrgs3.xvg chrg5L.xvg 1 4 10
11
source ../jobs/3dchrg4S.awk 50 chrgs3.xvg chrg5S.xvg 1 4 10
11
#!NOTE The above 11 lines can be invoked with source
lchrgall.
Source ../jobs/Bfintegr8.awk 50 chrg5A.xvg
source ../jobs/Bfintegr8.awk 50 chrg5H.xvg
source ../jobs/Bfintegr8.awk 50 chrg5N.xvg
source ../jobs/Bfintegr8.awk 50 chrg5C.xvg
source ../jobs/Bfintegr8.awk 50 chrg5P.xvg
source ../jobs/Bfintegr8.awk 50 chrg5W.xvg
source ../jobs/Bfintegr8.awk 50 chrg5R.xvg
source ../jobs/Bfintegr8.awk 50 chrg5T.xvg
source ../jobs/Bfintegr8.awk 50 chrg5Y.xvg
source ../jobs/Bfintegr8.awk 50 chrg5L.xvg
source ../jobs/Bfintegr8.awk 50 chrg5S.xvg
#!NOTE The above 11 invocations of Bfintegr8.awk can be
invoked with source integr8all.
# split apart the chrg5?.xvg files into the electric field
and electric potential sections
for s in A H N P W S C R T L Y; do
  s ../jobs/spltfld. ${ARG2:0:1} ${s}
done
# plot all the electric field and electric potential plots
gnuplot -c plfldmaps
#
```

```
# The Net Electric Flux maps show the resulting field and
potential for every atom in the system. The plot with the
fyld* name shows the electric field and the plot with the
fylt* name shows the potential. The other plots show the
same fields and potentials for the indicated moieties in
the system. Some of these are in the range of ten times the
magnitude of the overall field and potential. Each plot is
scaled to reveal all the detail present in the maps for
that moiety. The relative values of these scales are
contained in the a[subscript] array in the gnuplot script
plfldmaps.
```

Note that the *3dchrg4?.awk* routines have three or seven parameters passed to them. Every one of the routines on GitHub have comment section at the beginning that gives the necessary information for their use, including the description of the parameters passed to them, the options for them and their default values. Note that not all parameters can necessarily have a default value specified. All of these from the GitHub repository are listed in appendix A. Those for the *3dchrg4?.awk* and *Bfintegr8.awk* files are shown here for illustration in **Figs. S16-S18**

```
#!/bin/bash -l
#================================================================================
# 3dchrg4A.awk ### calculate avg charge in bins of volume for a trajectory       ###
#              ###                                                               ###
# parameters
# $1         number of horizontal (x-y) slices to create along z axis (2-?)
#   options: the output of this algorithm can be used to calculate non-uniform
#            electric fields or to produce computational x-rays. since the
#            computational cost of calculating electric fields scales as N6, and
#            that algorithm must process all the slices created here, balancing
#            resolution and computational time of both steps is required subject
#            to the requirements and limitations of each system on which it is used.
#            higher resolutions can be produced here if the intention is to
#            produce computational x-rays since they incure negligible scaling
# $2         name of file containing charges of each atom in system
# $3         output filname containing bins of average charge for each system volume
#            element
#_______________________________________________________________________________

ARG1=${1:-'50'}
ARG2=${2:-'chrgs2.xvg'}
ARG3=${3:-'chrg4.xvg'}
```

Fig. S16: *Parameters and explanation on use of routine 3dchrg4A.awk*

```
#!/bin/bash -l
#================================================================================
# 3dchrg4H.avg ### calculate avg charge in bins of volume for a moiety  ###
#              ### H is the lipid atoms between the phosphate group up  ###
#              ### to and including the ester groups.                   ###
# parameters
# $1         number of horizontal slices
# $2         file name, contains the charges for each atom in the .gro file
# $3         name of the output file
# $4         membrane channel to analyze
#   options  0 - top membrane, 1 - bottom membrane
# $5         number of extra water molecules in simulation
#   #! NOTE  there should have always been 11480 H2O molecules in each
#            simulation. Somewhere, somehow, another variant with 11484 H2O
#            molecules crept in. We have to know which we're dealing with.
# $6         number of ions in alpha bath
# $7         number of ions in beta bath
#________________________________________________________________________________

ARG1=${1:-'50'}
ARG2=${2:-'chrgs2.xvg'}
ARG3=${3:-'chrg4H.xvg'}
ARG4=${4:-'0'}
ARG5=${5:-'0'}
ARG6=${6:-'11'}
ARG7=${7:-'11'}
```

Fig. S17: *Parameters and explanation on use of routine 3dchrg4H.awk*

```
#!/usr/bin bash -l
#================================================================
# Bfintegr8.awk ### performs brute force calculation of     ###
#               ### electric field at location of each bin  ###
#               ### in simulation from charge in each bin   ###
# parameters
# $1    number of bins in z direction
# $2    charge file to integrate
# $3    output file name - outputs 2 sections, 1st is field on
#                          just that particular bin
#                          2nd is total field, i.e. accumulated
#                          field from pb to this posn
#   #!NOTE - the computational cost of this script scales as N6
#            of parameter $1. it gets very expensive very quick
#________________________________________________________________

ARG1=${1:-'50'}
```

Fig. S18: *Parameters and explanation on use of routine Bfintegr8.awk*

For those wishing to apply these tools to other works, note that in their current form that although the key algorithms have been carefully validated, certain sections of the

routines contain hard coded indexing schemes related to the structure of the topology files that were used for this work. Additional work is planned to create sections that can infer these indexing schemes directly from the topology files. If that becomes a higher priority to someone other than me or one of my students, please contact me so we can coordinate and minimized total invested effort, if you desire your application of this work to maintain compatibility with my planned extensions of functionality.

One last note: Any simulation specific molecular sub species you wish to define will require you become substantially proficient with the syntax of the GROMACS select syntax as documented in their documentation. Hopefully my tutorial on that subject in the SM of Section 1 will substantially decrease the challenge of that necessity.

Supporting Material – Section 3

To use PBMetaD in GROMACS, the Plumed package must first be installed. This is then used to patch the GROMACS source code which must subsequently be recompiled to propagate the patch to the binary executable files. The Plumed package is invoked in the GROMACS mdrun utility with the -plumed option followed by the name of the plumed options file, typically plumed.dat. The Plumed package has many capabilities; for a complete description of the options and the parameter settings to implement them, see the plumed website at plumed.org.

Plumed executables are used to perform this post-processing analysis. The required data is produced by invoking the plumed command line tool sum_hills, which integrates the HILLS file generated during the simulation for each of the collective variables. Since these files record the history of how the CV's are biased during the simulation as the free energy surface is explored, the sum_hills utility must be invoked in such a way to reveal details of the exploration process to quantify the extent of simulation convergence. A sample command for the native contacts CV is as follows:

```
plumed sum_hills --hills HILLS_native --stride 100000 --mintozero --outfile
fes.native.dat
```

The stride option creates a uniquely indexed instance of the output file fes.native.dat after each additional 100,000 ps (100 ns) of simulation data has been integrated into the cumulative total.

Production run .MDP file

```
integrator               = md                   ; MD integrator
tinit                    = 0
```

```
dt                      = 0.005               ; 2 fs timestep
nsteps                  = 400000000           ; Number of steps = 2000 ns
pbc                     = xyz                 ; Periodic boundary conditions
in xyz
comm-mode               = linear
nstcomm                 = 100
comm-grps               = System              ; Remove COM for monolayers
separately

; OUTPUT CONTROL OPTIONS
nstxout                 = 20000               ; Do not want .trr-files
nstxout-compressed      = 200  ; But do want .xtc-files
compressed-x-grps       = non-water
nstvout                 = 0                   ;no velocities in output
nstfout                 = 0                   ; No forces in output
;nstlist                = 10                  ; Update neighbor list between
cut-offs
;nstcalcenergy          = 10
;ns_type                = grid                ; Fastest option
nstenergy               = 200                 ; .edr-file output
nstlog                  = 200

; OPTIONS FOR ELECTROSTATICS AND VDW
cutoff-scheme           = Verlet
vdwtype                 = Cut-off
vdw_modifier            = Potential-switch
rvdw-switch             = 0.8
coulombtype             = PME                 ; Particle mesh Ewald, do not
change
pme_order               = 4                   ; 4th order interpolation
rcoulomb                = 1.0                 ; Real-space cut-off
                                              ; This is different from the
paper but
                                              ; since PME is used this will
only speed
                                              ; things up!
rlist                   = 1.0                 ; Short-range neighbor list
rvdw                    = 1.0                 ; Slightly different from the
original papers but we used this in our recent work,
                                              ; DOI: 10.1039/C3CP44472D.
Unofficial tests have shown that the bilayers still
                                              ; behaves properly when using
as short values as 1.0, but please verify this
                                              ; before using such a short
cut-off. rlistlong can then be removed.
DispCorr                = EnerPres            ; Dispersion corrections to
both the potential and pressure
table-extension         = 1
fourierspacing          = 0.12                ; PME grid

; OPTIONS FOR WEAK COUPLING ALGORITHMS
tcoupl                  = v-rescale           ; Thermostat, v-rescale is also
fine
tc-grps                 = System              ; Couple lipids and SOL
separately
tau-t                   = 0.5                 ; Time constant for temperature
coupling
ref-t                   = 310                 ; Desired temperature (K)
;Pcoupl                  = Parrinello-Rahman    ; Barostat
;Pcoupltype              = isotropic           ;
;ref-p                   = 1.013               ; Desired pressure (bar)
;tau-p                   = 10.0                 ; Time constant for pressure
coupling
```

```
;compressibility          = 4.5e-5              ; Same as for water

; CONSTRAINTS
continuation              = no                  ; Initial Simulation
constraints               = all-bonds           ; Constrain all bonds
constraint-algorithm      = Lincs               ; With Lincs

; VELOCITY GENERATION
gen_vel                   = yes
gen_temp                  = 310
gen_seed                  = -1
```

Simulation Convergence Details

TRPC				**GB1 (Hairpin)**		
Std. Dev.	**Abs(ΔA)**			**Std. Dev.**	**Abs(ΔA)**	
Kcal/mol	Kcal/mol	BenchM?	Parameters	Kcal/mol	Kcal/mol	BenchM?
0.342	0.204	1	AmbTipNa0	0.206	-0.560	1
1.180	-0.464	0	AmbTipNa150	0.238	-0.031	1
0.385	-0.348	1	AmbTipNa300	0.370	0.171	1
1.279	0.000	1	AmbTipNa1.0	0.579	0.147	1
0.932	-0.390	1	AmbSpcNa0	0.458	0.194	1
0.796	0.004	1	AmbSpcNa150	0.201	0.009	1
0.822	-0.539	1	AmbSpcNa300	0.244	-0.397	1
0.619	0.167	1	AmbSpcNa1.0	1.185	1.323	0
0.342	0.204	1	AmbTipK0	0.609	0.137	1
1.039	0.285	0	AmbTipK150	0.436	0.331	1
0.318	-0.174	1	AmbTipK300	0.437	0.053	1
0.433	-0.088	1	AmbTipK1.0	0.316	0.184	1
0.932	-0.390	1	AmbSpcK0	1.049	-0.033	1
0.212	-0.215	1	AmbSpcK150	0.107	0.090	1
0.333	0.005	1	AmbSpcK300	0.631	0.173	1
0.649	0.243	1	AmbSpcK1.0	0.127	0.034	1
0.483	-0.158	1	C36TipNa0	0.585	-0.273	1
0.273	0.099	1	C36TipNa150	1.663	-0.672	0
0.640	0.333	1	C36TipNa300	0.453	0.673	1
0.250	0.080	1	C36TipNa1.0	1.204	-0.433	0
0.483	-0.158	1	C36TipNa0	0.399	-0.149	1
0.445	0.202	1	C36TipNa150	0.383	0.252	1
0.455	0.459	1	C36TipNa300	0.299	0.240	1
0.157	0.064	1	C36TipNa1.0	0.817	0.034	1
0.768	-0.040	1	C27TipK0	2.344	-0.369	1

0.593	0.611	1	C27TipK150	1.210	0.828	0
0.329	1.018	1	C27TipK300	0.240	-0.081	1
1.202	-0.063	1	C27TipK1.0	0.284	-0.090	1
0.768	-0.040	1	C27TipK0	2.352	-0.738	0
0.636	-0.116	1	C27TipK150	0.542	0.054	1
1.195	-0.469	0	C27TipK300	0.377	-0.001	1
0.869	-0.243	1	C27TipK1.0	0.537	0.016	1
Benchmark Total		**29**		**Benchmark Total**		**27**

Figure S19: *Convergence benchmark evaluation details for TRPC and GB1*

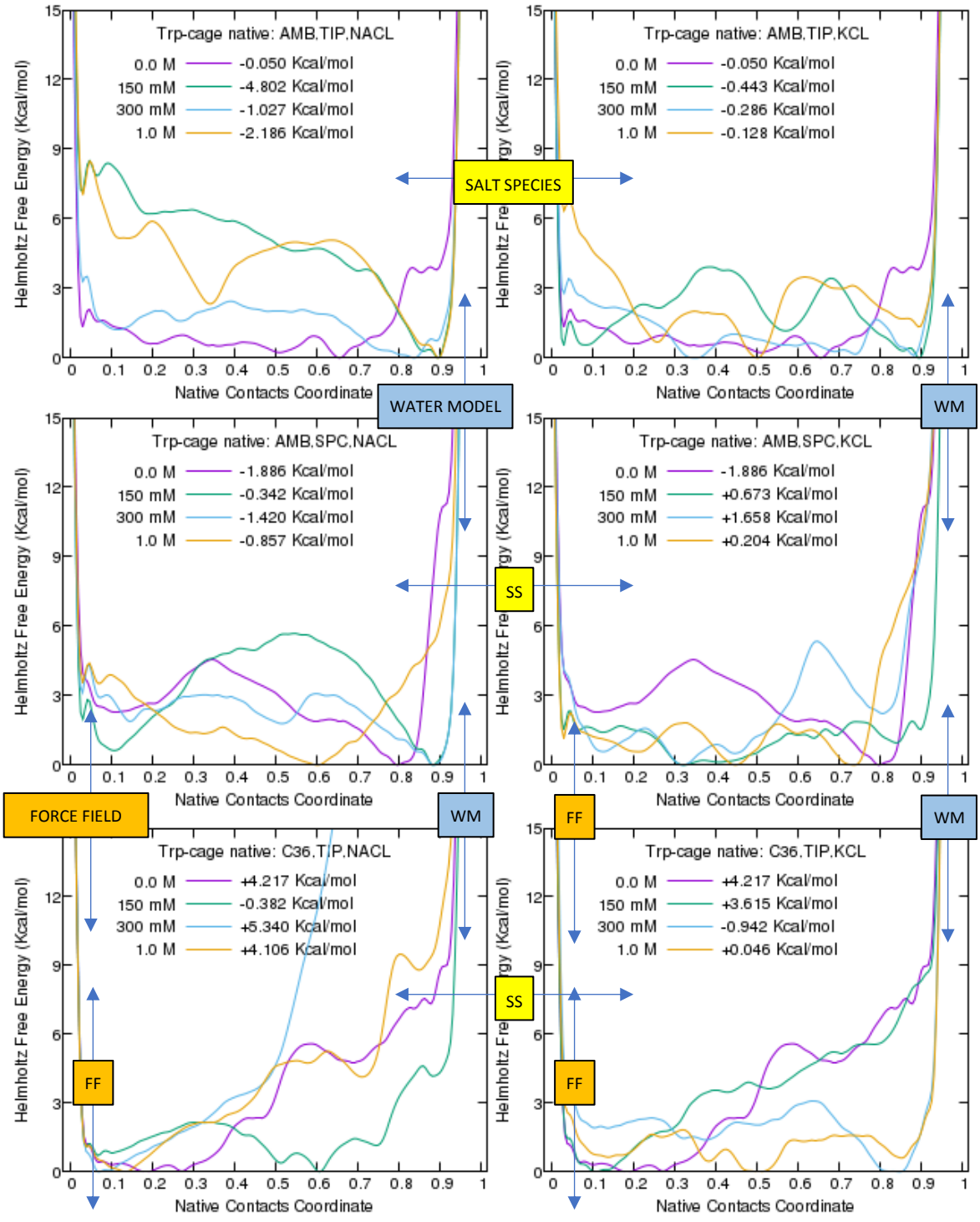


Fig. S20: *Panels 1-6, Folding free energy surface profiles and folding free energies for TRPC*

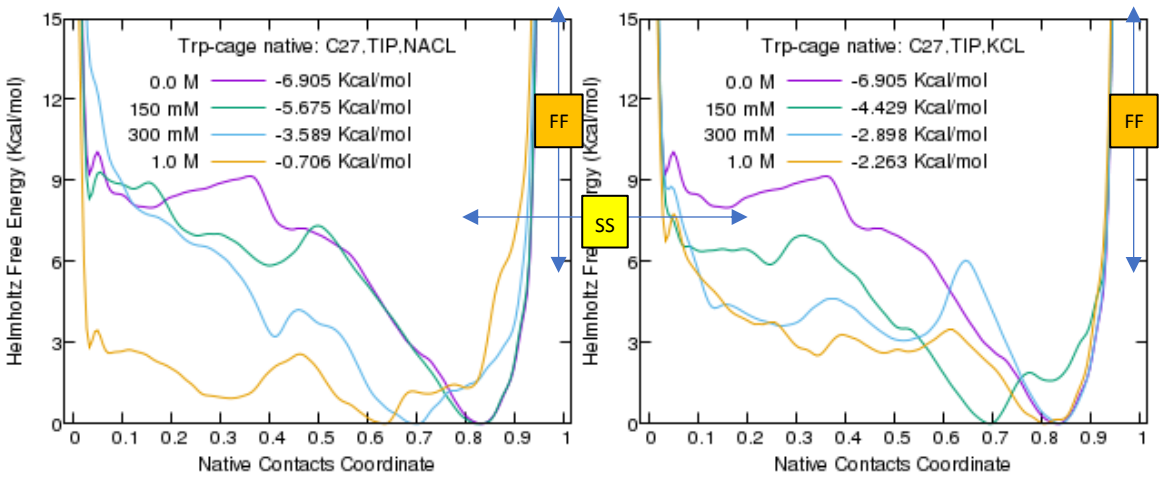


Fig. S20: *Panels 7-8, Folding free energy surface profiles and folding free energies for TRPC*

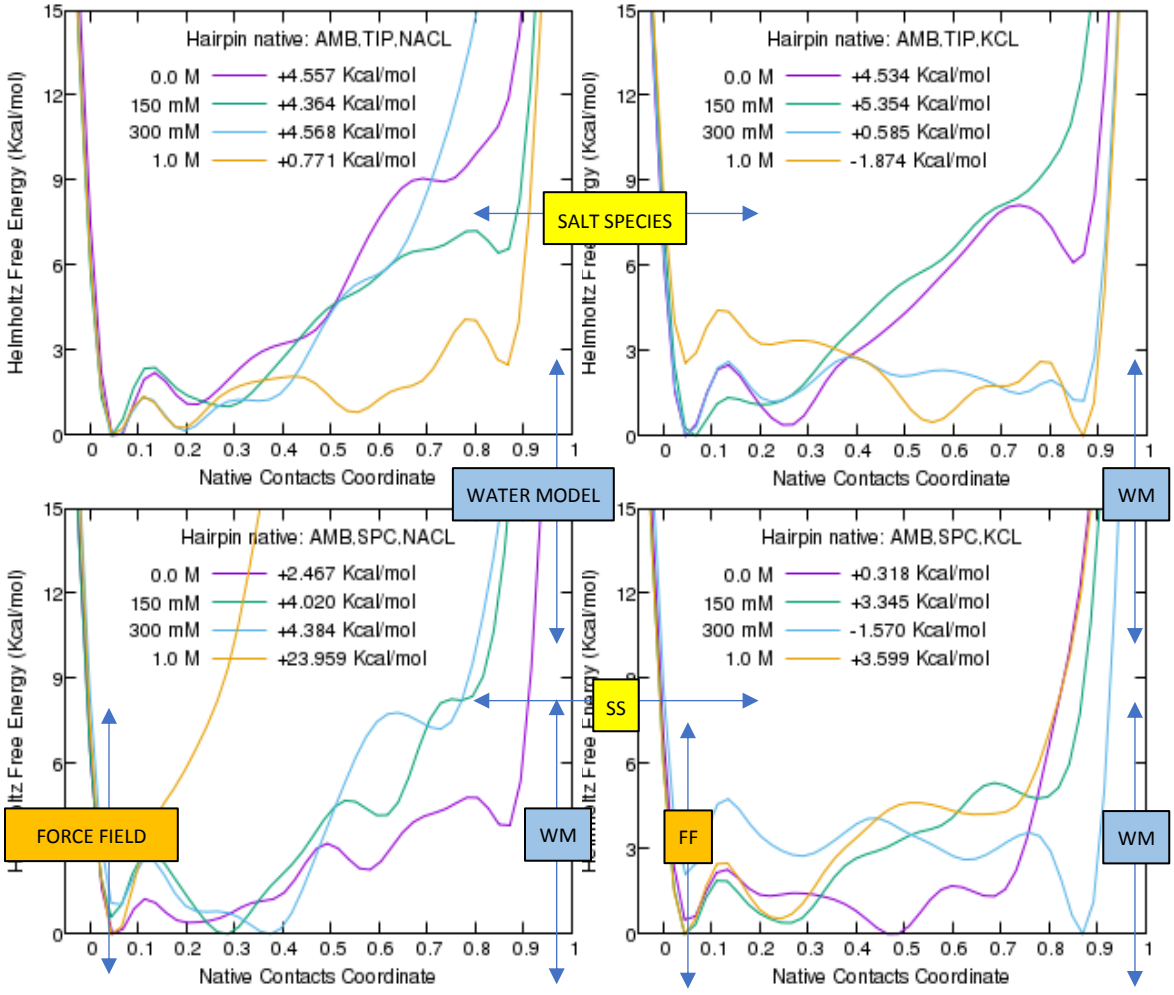


Fig. S21: *Panels 1-4, Folding free energy surface profiles and folding free energies for GB1.*

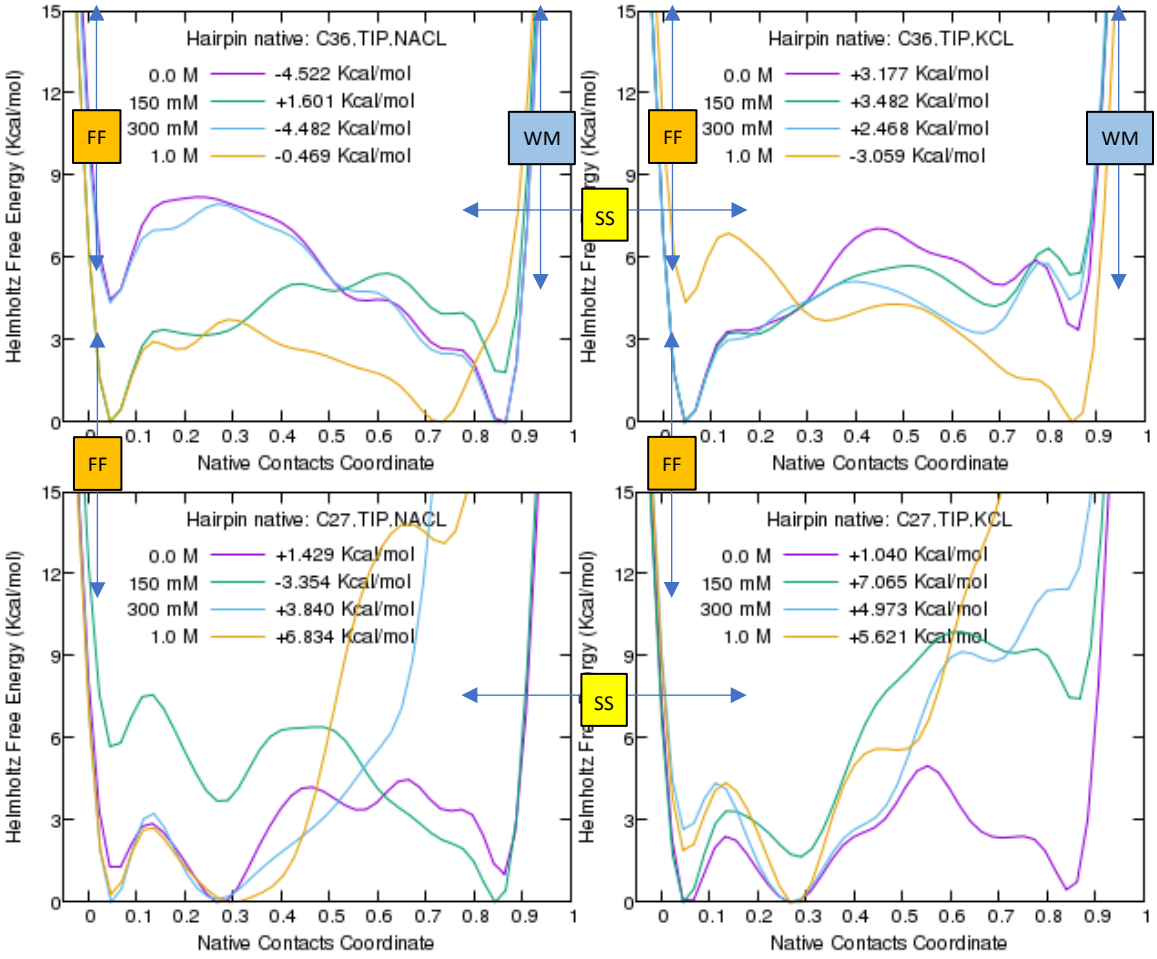


Fig. S21: *Panels 5-8, Folding free energy surface profiles and folding free energies for GB1.*

Supporting Material – Section 4

Production run .MDP file

```
include = -I/scratch/scratch/jpbria01/nitro/002/0
;
integrator              = md
dt                      = 0.004
nsteps                  = 250000000     ; 1 us
nstlog                  = 1000
nstxout                 = 250000
nstxout-compressed      = 12500
compressed-x-grps       = Non-water
nstvout                 = 0
nstfout                 = 0
nstcalcenergy           = 0
nstenergy               = 1000
;
cutoff-scheme           = Verlet
nstlist                 = 20
rlist                   = 1.0
coulombtype             = pme
rcoulomb                = 1.0
vdwtype                 = Cut-off
rvdw                    = 1.0
;
tcoupl                  = V-rescale
tc_grps                 = system
tau_t                   = 1.0
ref_t                   = 300.00
;
;pcoupl                  = no
pcoupl                  = Berendsen
pcoupltype              = isotropic
tau_p                   = 5.0
compressibility         = 4.5e-5
ref_p                   = 1.0
;
constraints             = h-bonds
constraint_algorithm    = LINCS
continuation            = no
gen_vel                 = yes
gen_temp                = 300.00
gen_seed                = -1
;
nstcomm                 = 100
comm_mode               = linear
comm_grps               = system
```

GAFF topology (.itp) files

For brevity, the [pairs], [angles], [dihedrals] and [diedrals] sections are not shown. None of the methods used to build this non-standard model had any effect on these sections.

```
; ADP_GMX.itp created by acpype (Rev: 0) on Thu Dec 26 11:22:20 2019

[ moleculetype ]
;name            nrexcl
 ADP              3

[ atoms ]
;   nr  type  resi  res  atom  cgnr     charge      mass
     1   p5     1   ADP     P    1     1.239089     30.97000
     2    o     1   ADP     O    2    -0.526624     16.00000
     3    o     1   ADP    O1    3    -0.526615     16.00000
     4    o     1   ADP    O2    4    -0.527687     16.00000
     5   p5     1   ADP    P1    5     1.202904     30.97000
     6    o     1   ADP    O3    6    -0.504924     16.00000
     7    o     1   ADP    O4    7    -0.472205     16.00000
     8   os     1   ADP    O5    8    -0.399328     16.00000
     9   os     1   ADP    O6    9    -0.433991     16.00000
    10   c3     1   ADP     C   10    -0.361351     12.01000
    11   c3     1   ADP    C1   11     0.421406     12.01000
    12   os     1   ADP    O7   12    -0.473731     16.00000
    13   c3     1   ADP    C2   13     0.261095     12.01000
    14   oh     1   ADP    O8   14    -0.726061     16.00000
    15   c3     1   ADP    C3   15     0.227341     12.01000
    16   oh     1   ADP    O9   16    -0.638082     16.00000
    17   c3     1   ADP    C4   17     0.333763     12.01000
    18   na     1   ADP     N   18    -0.149808     14.01000
    19   cc     1   ADP    C5   19     0.156787     12.01000
    20   na     1   ADP    N1   20    -0.167796     14.01000
    21   cc     1   ADP    C6   21    -0.472733     12.01000
    22   cc     1   ADP    C7   22     1.060063     12.01000
    23   n2     1   ADP    N2   23    -0.782525     14.01000
    24   nc     1   ADP    N3   24    -0.788932     14.01000
    25   cd     1   ADP    C8   25     0.715275     12.01000
    26   nd     1   ADP    N4   26    -0.818240     14.01000
    27   cc     1   ADP    C9   27     0.748062     12.01000
    28   h1     1   ADP     H   28     0.192865      1.00800
    29   h1     1   ADP    H1   29     0.178950      1.00800
    30   h1     1   ADP    H2   30     0.061042      1.00800
    31   h1     1   ADP    H3   31     0.044799      1.00800
    32   ho     1   ADP    H4   32     0.472163      1.00800
    33   h1     1   ADP    H5   33    -0.022546      1.00800
    34   ho     1   ADP    H6   34     0.390286      1.00800
    35   h2     1   ADP    H7   35     0.087727      1.00800
    36   h5     1   ADP    H8   36     0.164813      1.00800
    37   hn     1   ADP    H9   37     0.356710      1.00800
    38   hn     1   ADP   H10   38     0.404982      1.00800
    39   h5     1   ADP   H11   39     0.073059      1.00800
```

```
[ bonds ]
;   ai     aj  funct   r             k
     1      2   1    1.4866e-01    4.0125e+05
     1      3   1    1.4866e-01    4.0125e+05
     1      4   1    1.4866e-01    4.0125e+05
     1      8   1    1.6147e-01    2.7665e+05
     5      6   1    1.4866e-01    4.0125e+05
     5      7   1    1.4866e-01    4.0125e+05
     5      8   1    1.6147e-01    2.7665e+05
     5      9   1    1.6147e-01    2.7665e+05
     9     10   1    1.4316e-01    2.5824e+05
    10     11   1    1.5375e-01    2.5179e+05
    10     28   1    1.0969e-01    2.7665e+05
    10     29   1    1.0969e-01    2.7665e+05
    11     12   1    1.4316e-01    2.5824e+05
    11     13   1    1.5375e-01    2.5179e+05
    11     30   1    1.0969e-01    2.7665e+05
    12     17   1    1.4316e-01    2.5824e+05
    13     14   1    1.4233e-01    2.6501e+05
    13     15   1    1.5375e-01    2.5179e+05
    13     31   1    1.0969e-01    2.7665e+05
    14     32   1    9.7300e-02    3.1079e+05
    15     16   1    1.4233e-01    2.6501e+05
    15     17   1    1.5375e-01    2.5179e+05
    15     33   1    1.0969e-01    2.7665e+05
    16     34   1    9.7300e-02    3.1079e+05
    17     18   1    1.4629e-01    2.7422e+05
    17     35   1    1.0961e-01    2.7757e+05
    18     19   1    1.3802e-01    3.5631e+05
    18     27   1    1.3802e-01    3.5631e+05
    19     20   1    1.3802e-01    3.5631e+05
    19     36   1    1.0819e-01    2.9430e+05
    20     21   1    1.3802e-01    3.5631e+05
    20     37   1    1.0100e-01    3.4175e+05
    21     22   1    1.4278e-01    3.5129e+05
    21     27   1    1.4278e-01    3.5129e+05
    22     23   1    1.2905e-01    4.8208e+05
    22     24   1    1.3694e-01    3.6911e+05
    23     38   1    1.0230e-01    3.2267e+05
    24     25   1    1.3172e-01    4.3965e+05
    25     26   1    1.3694e-01    3.6911e+05
    25     39   1    1.0818e-01    2.9439e+05
    26     27   1    1.3172e-01    4.3965e+05

; Include Position restraint file
#ifdef POSRES
#include "../../002/+/nitrogenase/posre_Protein_chain_E4.itp"
#endif

; ACP_GMX.itp created by acpype (Rev: 0) on Thu Dec 26 10:04:39 2019

[ moleculetype ]
;name            nrexcl
 ACP              3

[ atoms ]
```

```
;   nr  type  resi  res  atom  cgnr     charge      mass
     1   p5     1   ACP     P    1     1.289652     30.97000
     2    o     1   ACP     O    2    -0.503385     16.00000
     3    o     1   ACP    O1    3    -0.409278     16.00000
     4    o     1   ACP    O2    4    -0.428459     16.00000
     5   p5     1   ACP    P1    5     1.075045     30.97000
     6    o     1   ACP    O3    6    -0.414531     16.00000
     7    o     1   ACP    O4    7    -0.422927     16.00000
     8   p5     1   ACP    P2    8     1.143977     30.97000
     9    o     1   ACP    O5    9    -0.493836     16.00000
    10    o     1   ACP    O6   10    -0.552672     16.00000
    11   os     1   ACP    O7   11    -0.378968     16.00000
    12   os     1   ACP    O8   12    -0.394184     16.00000
    13   c3     1   ACP     C   13    -0.303689     12.01000
    14   c3     1   ACP    C1   14     0.333471     12.01000
    15   os     1   ACP    O9   15    -0.469591     16.00000
    16   c3     1   ACP    C2   16     0.329692     12.01000
    17   oh     1   ACP   O10   17    -0.732964     16.00000
    18   c3     1   ACP    C3   18     0.117914     12.01000
    19   oh     1   ACP   O11   19    -0.646764     16.00000
    20   c3     1   ACP    C4   20     0.402315     12.01000
    21   na     1   ACP     N   21    -0.205091     14.01000
    22   cc     1   ACP    C5   22     0.143875     12.01000
    23   na     1   ACP    N1   23    -0.152255     14.01000
    24   cc     1   ACP    C6   24    -0.504626     12.01000
    25   cc     1   ACP    C7   25     1.060181     12.01000
    26   n2     1   ACP    N2   26    -0.803425     14.01000
    27   nc     1   ACP    N3   27    -0.783112     14.01000
    28   cd     1   ACP    C8   28     0.702647     12.01000
    29   nd     1   ACP    N4   29    -0.815295     14.01000
    30   cc     1   ACP    C9   30     0.783343     12.01000
    31   os     1   ACP   O12   31    -0.443229     16.00000
    32   h1     1   ACP     H   32     0.190142      1.00800
    33   h1     1   ACP    H1   33     0.157799      1.00800
    34   h1     1   ACP    H2   34     0.078124      1.00800
    35   h1     1   ACP    H3   35     0.037752      1.00800
    36   ho     1   ACP    H4   36     0.474061      1.00800
    37   h1     1   ACP    H5   37     0.033365      1.00800
    38   ho     1   ACP    H6   38     0.406994      1.00800
    39   h2     1   ACP    H7   39     0.081276      1.00800
    40   h5     1   ACP    H8   40     0.180491      1.00800
    41   hn     1   ACP    H9   41     0.354485      1.00800
    42   hn     1   ACP   H10   42     0.405741      1.00800
    43   h5     1   ACP   H11   43     0.075940      1.00800

[ bonds ]
;   ai     aj  funct   r             k
     1      2   1    1.4866e-01    4.0125e+05
     1      3   1    1.4866e-01    4.0125e+05
     1      4   1    1.4866e-01    4.0125e+05
     1     31   1    1.6147e-01    2.7665e+05
     5      6   1    1.4866e-01    4.0125e+05
     5      7   1    1.4866e-01    4.0125e+05
     5     11   1    1.6147e-01    2.7665e+05
     5     31   1    1.6147e-01    2.7665e+05
     8      9   1    1.4866e-01    4.0125e+05
     8     10   1    1.4866e-01    4.0125e+05
```

```
     8     11    1    1.6147e-01    2.7665e+05
     8     12    1    1.6147e-01    2.7665e+05
    12     13    1    1.4316e-01    2.5824e+05
    13     14    1    1.5375e-01    2.5179e+05
    13     32    1    1.0969e-01    2.7665e+05
    13     33    1    1.0969e-01    2.7665e+05
    14     15    1    1.4316e-01    2.5824e+05
    14     16    1    1.5375e-01    2.5179e+05
    14     34    1    1.0969e-01    2.7665e+05
    15     20    1    1.4316e-01    2.5824e+05
    16     17    1    1.4233e-01    2.6501e+05
    16     18    1    1.5375e-01    2.5179e+05
    16     35    1    1.0969e-01    2.7665e+05
    17     36    1    9.7300e-02    3.1079e+05
    18     19    1    1.4233e-01    2.6501e+05
    18     20    1    1.5375e-01    2.5179e+05
    18     37    1    1.0969e-01    2.7665e+05
    19     38    1    9.7300e-02    3.1079e+05
    20     21    1    1.4629e-01    2.7422e+05
    20     39    1    1.0961e-01    2.7757e+05
    21     22    1    1.3802e-01    3.5631e+05
    21     30    1    1.3802e-01    3.5631e+05
    22     23    1    1.3802e-01    3.5631e+05
    22     40    1    1.0819e-01    2.9430e+05
    23     24    1    1.3802e-01    3.5631e+05
    23     41    1    1.0100e-01    3.4175e+05
    24     25    1    1.4278e-01    3.5129e+05
    24     30    1    1.4278e-01    3.5129e+05
    25     26    1    1.2905e-01    4.8208e+05
    25     27    1    1.3694e-01    3.6911e+05
    26     42    1    1.0230e-01    3.2267e+05
    27     28    1    1.3172e-01    4.3965e+05
    28     29    1    1.3694e-01    3.6911e+05
    28     43    1    1.0818e-01    2.9439e+05
    29     30    1    1.3172e-01    4.3965e+05

; Include Position restraint file
#ifdef POSRES
#include "../../002/+/nitrogenase/posre_Protein_chain_F3.itp"
#endif

; HCA_GMX.itp created by acpype (Rev: 0) on Thu Dec  5 13:26:11 2019

[ moleculetype ]
;name            nrexcl
 HCA              3

[ atoms ]
;   nr  type  resi  res  atom  cgnr     charge      mass
     1    c     1   HCA     C    1     0.729821     12.01000
     2   c3     1   HCA    C1    2    -0.513822     12.01000
     3   c3     1   HCA    C2    3     0.473279     12.01000
     4   c3     1   HCA    C3    4    -0.377786     12.01000
     5   c3     1   HCA    C4    5    -0.167644     12.01000
     6    c     1   HCA    C5    6     0.742316     12.01000
     7    c     1   HCA    C6    7     0.549150     12.01000
     8    o     1   HCA     O    8    -0.371960     16.00000
```

```
     9     o      1    HCA     O1     9     -0.383845     16.00000
    10     o      1    HCA     O2    10     -0.364331     16.00000
    11     o      1    HCA     O3    11     -0.390665     16.00000
    12     o      1    HCA     O4    12     -0.216910     16.00000
    13     o      1    HCA     O5    13     -0.420707     16.00000
    14    oh      1    HCA     O6    14     -0.601278     16.00000
    15    hc      1    HCA      H    15      0.204971      1.00800
    16    hc      1    HCA     H1    16      0.181412      1.00800
    17    hc      1    HCA     H2    17      0.133291      1.00800
    18    hc      1    HCA     H3    18      0.143576      1.00800
    19    hc      1    HCA     H4    19      0.118779      1.00800
    20    hc      1    HCA     H5    20      0.081238      1.00800
    21    ho      1    HCA     H6    21      0.451115      1.00800

[ bonds ]
;   ai     aj  funct   r             k
     1      2    1    1.5241e-01    2.6192e+05
     1      8    1    1.2183e-01    5.3363e+05
     1      9    1    1.2183e-01    5.3363e+05
     2      3    1    1.5375e-01    2.5179e+05
     2     15    1    1.0969e-01    2.7665e+05
     2     16    1    1.0969e-01    2.7665e+05
     3      4    1    1.5375e-01    2.5179e+05
     3      7    1    1.5241e-01    2.6192e+05
     3     14    1    1.4233e-01    2.6501e+05
     4      5    1    1.5375e-01    2.5179e+05
     4     17    1    1.0969e-01    2.7665e+05
     4     18    1    1.0969e-01    2.7665e+05
     5      6    1    1.5241e-01    2.6192e+05
     5     19    1    1.0969e-01    2.7665e+05
     5     20    1    1.0969e-01    2.7665e+05
     6     10    1    1.2183e-01    5.3363e+05
     6     11    1    1.2183e-01    5.3363e+05
     7     12    1    1.2183e-01    5.3363e+05
     7     13    1    1.2183e-01    5.3363e+05
    14     21    1    9.7300e-02    3.1079e+05

; Include Position restraint file
#ifdef POSRES
#include "../+/nitrogenase/posre_Protein_chain_A2.itp"
#endif

; SF4_GMX.itp created by acpype (Rev: 0) on Thu Dec  5 13:48:48 2019

[ moleculetype ]
;name            nrexcl
 SF4              3

[ atoms ]
;   nr  type  resi  res  atom  cgnr     charge      mass
     1   FE     1   SF4    FE    1     0.430000     55.84500
     2   FE     1   SF4   FE1    2     0.430000     55.84500
     3   FE     1   SF4   FE2    3     0.430000     55.84500
     4   FE     1   SF4   FE3    4     0.430000     55.84500
     5   sx     1   SF4     S    5    -0.430000     32.06000
     6   sx     1   SF4    S1    6    -0.430000     32.06000
     7   sx     1   SF4    S2    7    -0.430000     32.06000
```

```
     8   sx     1   SF4    S3    8    -0.430000     32.06000

[ bonds ]
;   ai     aj funct   r             k
     1      6   1    2.1945e-01    0.8672e+05
     1      7   1    2.1945e-01    0.8672e+05
     1      8   1    2.1945e-01    0.8672e+05
     2      5   1    2.1945e-01    0.8672e+05
     2      7   1    2.1945e-01    0.8672e+05
     2      8   1    2.1945e-01    0.8672e+05
     3      5   1    2.1945e-01    0.8672e+05
     3      6   1    2.1945e-01    0.8672e+05
     3      8   1    2.1945e-01    0.8672e+05
     4      5   1    2.1945e-01    0.8672e+05
     4      6   1    2.1945e-01    0.8672e+05
     4      7   1    2.1945e-01    0.8672e+05

; Include Position restraint file
#ifdef POSRES
#include "../+/nitrogenase/posre_Protein_chain_E2.itp"
#endif

; CLF_GMX.itp created by acpype (Rev: 0) on Mon Dec 23 11:48:15 2019

[ moleculetype ]
;name            nrexcl
 CLF              3

[ atoms ]
;   nr  type  resi  res  atom  cgnr     charge      mass
     1   FE     1   CLF    FE    1     0.359521     32.06000
     2   FE     1   CLF   FE1    2     0.505082     32.06000
     3   FE     1   CLF   FE2    3     0.455699     32.06000
     4   FE     1   CLF   FE3    4     0.337948     32.06000
     5   s6     1   CLF     S    5    -0.629283     32.06000
     6   sx     1   CLF    S1    6    -0.474445     32.06000
     7   sx     1   CLF    S2    7    -0.457870     32.06000
     8   sx     1   CLF    S3    8    -0.407254     32.06000
     9   FE     1   CLF   FE4    9     0.368484     32.06000
    10   FE     1   CLF   FE5   10     0.511675     32.06000
    11   FE     1   CLF   FE6   11     0.442635     32.06000
    12   FE     1   CLF   FE7   12     0.323678     32.06000
    13   sx     1   CLF    S4   13    -0.473587     32.06000
    14   sx     1   CLF    S5   14    -0.463964     32.06000
    15   sx     1   CLF    S6   15    -0.398317     32.06000

[ bonds ]
;   ai     aj funct   r             k
     1      5   1    2.4346e-01    0.8672e+05
     1      6   1    2.2945e-01    0.8672e+05
     1      8   1    2.2945e-01    0.8672e+05
     2      5   1    2.4346e-01    0.8672e+05
     2      6   1    2.2945e-01    0.8672e+05
     2      7   1    2.2945e-01    0.8672e+05
     3      6   1    2.3378e-01    0.8672e+05
     3      7   1    2.3378e-01    0.8672e+05
     3      8   1    2.3378e-01    0.8672e+05
```

```
     4       5   1    2.4346e-01    0.8672e+05
     4       7   1    2.2945e-01    0.8672e+05
     4       8   1    2.2945e-01    0.8672e+05
     5       9   1    2.4346e-01    0.8672e+05
     5      10   1    2.4346e-01    0.8672e+05
     5      12   1    2.4346e-01    0.8672e+05
     9      13   1    2.2945e-01    0.8672e+05
     9      15   1    2.2945e-01    0.8672e+05
    10      13   1    2.2945e-01    0.8672e+05
    10      14   1    2.2945e-01    0.8672e+05
    11      13   1    2.3378e-01    0.8672e+05
    11      14   1    2.3378e-01    0.8672e+05
    11      15   1    2.3378e-01    0.8672e+05
    12      14   1    2.2945e-01    0.8672e+05
    12      15   1    2.2945e-01    0.8672e+05

; Include Position restraint file
#ifdef POSRES
#include "posre_Protein_chain_A4.itp"
#endif

; ICS_GMX.itp created by acpype (Rev: 0) on Fri Dec 20 16:14:54 2019

[ moleculetype ]
;name            nrexcl
 ICS              3

[ atoms ]
;   nr  type  resi  res  atom  cgnr     charge      mass
     1   FE     1   ICS    FE    1     0.512069     55.84500
     2   MO     1   ICS    MO    2     0.895089     95.95000
     3   FE     1   ICS   FE1    3    -0.074600     55.84500
     4   FE     1   ICS   FE2    4    -0.074600     55.84500
     5   FE     1   ICS   FE3    5    -0.074600     55.84500
     6   FE     1   ICS   FE4    6    -0.237079     55.84500
     7   FE     1   ICS   FE5    7    -0.237079     55.84500
     8   FE     1   ICS   FE6    8    -0.237079     55.84500
     9   c3     1   ICS     C    9     1.422853     12.01000
    10   s4     1   ICS     S   10    -0.248527     32.06000
    11   s4     1   ICS    S1   11    -0.209127     32.06000
    12   s4     1   ICS    S2   12    -0.248527     32.06000
    13   ss     1   ICS    S3   13    -0.174004     32.06000
    14   ss     1   ICS    S4   14    -0.174004     32.06000
    15   s4     1   ICS    S5   15    -0.209127     32.06000
    16   s4     1   ICS    S6   16    -0.248527     32.06000
    17   s4     1   ICS    S7   17    -0.209127     32.06000
    18   ss     1   ICS    S8   18    -0.174004     32.06000

[ bonds ]
;   ai     aj funct   r             k
     1     10   1    2.6652e-01    0.8672e+05
     1     12   1    2.6652e-01    0.8672e+05
     1     16   1    2.6652e-01    0.8672e+05
     2     11   1    2.2590e-01    0.8672e+05
     2     15   1    2.2590e-01    0.8672e+05
     2     17   1    2.2590e-01    0.8672e+05
     3      9   1    1.9832e-01    0.8672e+05
```

```
     3      10    1    2.2666e-01    0.8672e+05
     3      12    1    2.2666e-01    0.8672e+05
     3      13    1    2.2181e-01    0.8672e+05
     4       9    1    1.9832e-01    0.8672e+05
     4      12    1    2.2666e-01    0.8672e+05
     4      16    1    2.2666e-01    0.8672e+05
     4      18    1    2.2181e-01    0.8672e+05
     5       9    1    1.9832e-01    0.8672e+05
     5      10    1    2.2666e-01    0.8672e+05
     5      14    1    2.2181e-01    0.8672e+05
     5      16    1    2.2666e-01    0.8672e+05
     6       9    1    2.0179e-01    0.8672e+05
     6      11    1    2.2489e-01    0.8672e+05
     6      14    1    2.2353e-01    0.8672e+05
     6      17    1    2.2489e-01    0.8672e+05
     7       9    1    2.0179e-01    0.8672e+05
     7      11    1    2.2489e-01    0.8672e+05
     7      13    1    2.2353e-01    0.8672e+05
     7      15    1    2.2489e-01    0.8672e+05
     8       9    1    2.0179e-01    0.8672e+05
     8      15    1    2.2489e-01    0.8672e+05
     8      17    1    2.2489e-01    0.8672e+05
     8      18    1    2.2353e-01    0.8672e+05

; Include Position restraint file
#ifdef POSRES
#include "../+/nitrogenase/posre_Protein_chain_A3.itp"
#endif
```

QM ESP calculation results

Atom	ADP	ACP	HCA	4Fe-4S	P-Cluster	Fe-Mo
1	1.290	1.239	0.730	0.430	0.360	0.512
2	-0.503	-0.527	-0.514	0.430	0.505	0.895
3	-0.409	-0.527	0.473	0.430	0.456	-0.075
4	-0.428	-0.528	-0.378	0.430	0.338	-0.075
5	1.075	1.203	-0.168	-0.430	-0.629	-0.075
6	-0.415	-0.505	0.742	-0.430	-0.474	-0.237
7	-0.423	-0.472	0.549	-0.430	-0.458	-0.237
8	1.144	-0.399	-0.372	-0.430	-0.407	-0.237
9	-0.494	-0.434	-0.384		0.368	1.423
10	-0.553	-0.361	-0.364		0.512	-0.249
11	-0.379	0.421	-0.391		0.443	-0.209
12	-0.394	-0.474	-0.217		0.324	-0.249
13	-0.304	0.261	-0.421		-0.474	-0.174
14	0.333	-0.726	-0.601		-0.464	-0.174
15	-0.470	0.227	0.205		-0.398	-0.209
16	0.330	-0.638	0.181			-0.249
17	-0.733	0.334	0.133			-0.209
18	0.118	-0.150	0.144			-0.174

Atom	ADP	ACP	HCA	4Fe-4S	P-Cluster	Fe-Mo
19	-0.647	0.157	0.119			
20	0.402	-0.168	0.081			
21	-0.205	-0.473	0.451			
22	0.144	1.060				
23	-0.152	-0.783				
24	-0.505	-0.789				
25	1.060	0.715				
26	-0.803	-0.818				
27	-0.783	0.748				
28	0.703	0.193				
29	-0.815	0.179				
30	0.783	0.061				
31	-0.443	0.045				
32	0.190	0.472				
33	0.158	-0.023				
34	0.078	0.390				
35	0.038	0.088				
36	0.474	0.165				
37	0.033	0.357				
38	0.407	0.405				
39	0.081	0.073				
40	0.180					
41	0.354					
42	0.406					
43	0.076					

Fig. S22: *QM ESP calculation results. This table is the source of the charges present in the GAFF .itp files.*

REFERENCES

1. Shi, J. F., Y. F. Ma, J. Zhu, Y. X. Chen, Y. T. Sun, Y. C. Yao, Z. G. Yang, and J. Xie. 2018. A Review on Electroporation-Based Intracellular Delivery. In Molecules.
2. Nakamura, H., and S. Watano. 2018. Direct Permeation of Nanoparticles Across Cell Membrane: A Review. Kona Powder Part J(35):49-65.
3. Kar, S., M. Loganathan, K. Dey, P. Shinde, H. Y. Chang, M. Nagai, and T. S. Santra. 2018. Single-cell electroporation: current trends, applications and future prospects. J Micromech Microeng 28(12).
4. Boukany, P. E., A. Morss, W. C. Liao, B. Henslee, H. C. Jung, X. L. Zhang, B. Yu, X. M. Wang, Y. Wu, L. Li, K. L. Gao, X. Hu, X. Zhao, O. Hemminger, W. Lu, G. P. Lafyatis, and L. J. Lee. 2011. Nanochannel electroporation delivers precise amounts of biomolecules into living cells. Nat Nanotechnol 6(11):747-754.
5. Potter, H. 2003. Transfection by electroporation. Curr Protoc Cell Biol Chapter 20:Unit 20 25.
6. Saulis, G. 1997. Pore disappearance in a cell after electroporation: Theoretical simulation and comparison with experiments. Biophys J 73(3):1299-1309.
7. Li, S., J. Cutrera, R. Heller, and J. Teissie. 2014. Electroporation protocols : preclinical and clinical gene medicine. Humana Press ;, New York.
8. Kim, K., and W. G. Lee. 2017. Electroporation for nanomedicine: a review. J Mater Chem B 5(15):2726-2738.
9. Betters, E., R. M. Charney, and M. I. Garcia-Castro. 2018. Electroporation and in vitro culture of early rabbit embryos. Data Brief 21:316-320.
10. Cruz-Ramon, J., F. J. Fernandez, A. Mejia, and F. Fierro. 2019. Electroporation of germinated conidia and young mycelium as an efficient transformation system for Acremonium chrysogenum. Folia Microbiol 64(1):33-39.
11. Wang, L., L. J. Yang, X. Wen, Z. Y. Chen, Q. Y. Liang, J. L. Li, and W. Wang. 2019. Rapid and high efficiency transformation of Chlamydomonas reinhardtii by square-wave electroporation. Bioscience Rep 39.
12. Mukherjee, P., S. S. P. Nathamgari, J. A. Kessler, and H. D. Espinosa. 2018. Combined Numerical and Experimental Investigation of Localized Electroporation-Based Cell Transfection and Sampling. Acs Nano 12(12):12118-12128.
13. Germann, M., T. Latychevskaia, C. Escher, and H. W. Fink. 2010. Nondestructive Imaging of Individual Biomolecules. Phys Rev Lett 104(9).
14. Lyubartsev, A. P., and A. L. Rabinovich. 2016. Force Field Development for Lipid Membrane Simulations. Bba-Biomembranes 1858(10):2483-2497.
15. Sandoval-Perez, A., K. Pluhackova, and R. A. Bockmann. 2017. Critical Comparison of Biomembrane Force Fields: Protein-Lipid Interactions at the

Membrane Interface. Journal of Chemical Theory and Computation 13(5):2310-2321.
16. Sajadi, F., and C. N. Rowley. 2018. Simulations of lipid bilayers using the CHARMM36 force field with the TIP3P-FB and TIP4P-FB water models. Peerj 6.
17. Leonard, A. N., R. W. Pastor, and J. B. Klauda. 2018. Parameterization of the CHARMM All-Atom Force Field for Ether Lipids and Model Linear Ethers. J. Phys. Chem. B 122(26):6744-6754.
18. Park, S., and W. Im. 2019. Analysis of Lipid Order States and Domains in Lipid Bilayer Simulations. Journal of Chemical Theory and Computation 15(1):688-697.
19. Tarek, M. 2005. Membrane electroporation: A molecular dynamics simulation. Biophys J 88(6):4045-4053.
20. Kong, Z., H. B. Wang, L. J. Liang, Z. S. Zhang, S. B. Ying, Q. Hu, and J. W. Shen. 2017. Investigation of the morphological transition of a phospholipid bilayer membrane in an external electric field via molecular dynamics simulation. J Mol Model 23(4).
21. Kutzner, C., H. Grubmuller, B. L. de Groot, and U. Zachariae. 2011. Computational Electrophysiology: The Molecular Dynamics of Ion Channel Permeation and Selectivity in Atomistic Detail. Biophys J 101(4):809-817.
22. Hyvonen, M. T., T. T. Rantala, and M. AlaKorpela. 1997. Structure and dynamic properties of diunsaturated 1-palmitoyl-2-linoleoyl-sn-glycero-3-phosphatidylcholine lipid bilayer from molecular dynamics simulation. Biophys J 73(6):2907-2923.
23. Im, W., and B. Roux. 2002. Ions and counterions in a biological channel: A molecular dynamics simulation of OmpF porin from Escherichia coli in an explicit membrane with 1 M KCl aqueous salt solution. J Mol Biol 319(5):1177-1197.
24. Skelton, A. A., V. M. Khedkar, and J. R. Fried. 2016. All-atom molecular dynamics simulations of an artificial sodium channel in a lipid bilayer: the effect of water solvation/desolvation of the sodium ion. J Biomol Struct Dyn 34(3):529-539.
25. Zachariae, U., C. Kutzner, H. Grubmuller, and B. de Groot. 2012. Computational Electrophysiology Reveals Ion Channel Permeation in Atomistic Detail. Biophys J 102(3):22a-22a.
26. Zachariae, U. 2016. Computational Electrophysiology: Close-UPS of Ion Permeation and Migration in Membrane Proteins. Biophys J 110(3):376a-376a.
27. Kutzner, C., R. T. Ullmann, B. L. de Groot, U. Zachariae, and H. Grubmueller. 2017. Ions in Action - Studying Ion Channels by Computational Electrophysiology in GROMACS. Biophys J 112(3):139a-139a.
28. Djedovic, N., R. Ferdani, E. Harder, J. Pajewska, R. Pajewski, M. E. Weber, P. H. Schlesinger, and G. W. Gokel. 2005. The C- and N-terminal residues of synthetic heptapeptide ion channels influence transport efficacy through phospholipid bilayers. New J Chem 29(2):291-305.
29. Schlesinger, P. H., R. Ferdani, J. Liu, J. Pajewska, R. Pajewski, M. Saito, H. Shabany, and G. W. Gokel. 2002. SCMTR: A chloride-selective, membrane-

anchored peptide channel that exhibits voltage gating. J Am Chem Soc 124(9):1848-1849.
30. Minooei, F., M. D. Martin, J. R. Fried, and J. P. Brian. 2018. Electrophysiological measurements reveal that a succinyl linker enhances performance of the synthetic chloride channel SCMTR. Chem Commun 54(37):4689-4691.
31. Burkhardt, J. B., A. A. Skelton, and J. R. Fried. 2013. The water-channel forming ability of heptapeptide-based anion channels: insights from molecular dynamics simulations. Soft Matter 9(17):4444-4454.
32. Chernomordik, L. V., and I. G. Abidor. 1980. Cation Selectivity of Blm in a Stress State. Biofizika+ 25(5):918-919.
33. Gokel, G. W. 2000. Hydraphiles: design, synthesis and analysis of a family of synthetic, cation-conducting channels. Chem Commun(1):1-9.
34. Gokel, G. W. 1992. Lariat Ethers - from Simple Sidearms to Supramolecular Systems. Chem Soc Rev 21(1):39-47.
35. Bennett, W. F. D., C. K. Hong, Y. Wang, and D. P. Tieleman. 2016. Antimicrobial Peptide Simulations and the Influence of Force Field on the Free Energy for Pore Formation in Lipid Bilayers. Journal of Chemical Theory and Computation 12(9):4524-4533.
36. Tang, J. C., H. R. Yin, J. L. Ma, W. F. Bo, Y. Yang, J. Xu, Y. Y. Liu, and Y. B. Gong. 2018. Terahertz Electric Field-Induced Membrane Electroporation by Molecular Dynamics Simulations. J Membrane Biol 251(5-6):681-693.
37. Cevc, G. M., D. 1987. Phospholipid Bilayers - Physical Principles and Models. John Wiley & Sons: New York.
38. Ollila, O. H. S. V. 2010. Molecular Simulations and Biomembranes; From Biophysics to Function. M. S. P. Sansom, Biggin, P. C., editor. The Royal Society of Chemistry, London, pp. 26-55.
39. Orsi, M., M. G. Noro, and J. W. Essex. 2011. Dual-resolution molecular dynamics simulation of antimicrobials in biomembranes. J R Soc Interface 8(59):826-841.
40. Cantor, R. S. 1997. Lateral pressures in cell membranes: A mechanism for modulation of protein function. J. Phys. Chem. B 101(10):1723-1725.
41. Patra, M. 2005. Lateral pressure profiles in cholesterol-DPPC bilayers. Eur Biophys J Biophy 35(1):79-88.
42. Ollila, S., M. T. Hyvonen, and I. Vattulainen. 2007. Polyunsaturation in lipid membranes: Dynamic properties and lateral pressure profiles. J. Phys. Chem. B 111(12):3139-3150.
43. Pera, H., J. M. Kleijn, and F. A. M. Leermakers. 2014. Linking lipid architecture to bilayer structure and mechanics using self-consistent field modelling. J Chem Phys 140(6).
44. Galanis, J., and Y. Tsori. 2014. Interface initiation and propagation in liquid demixing with electric fields. J Chem Phys 141(21).
45. Galanis, J., and Y. Tsori. 2014. Phase separation dynamics of simple liquids in non-uniform electric fields. J Chem Phys 140(12).
46. Galanis, J., and Y. Tsori. 2019. Surface tension and domain growth in nonuniform electric fields. Eur Phys J-Spec Top 227(18):2675-2687.

47. Woo, S. Y., and H. Lee. 2017. Effect of lipid shape on toroidal pore formation and peptide orientation in lipid bilayers. Phys Chem Chem Phys 19(32):21340-21349.
48. Tieleman, D. P., H. Leontiadou, A. E. Mark, and S. J. Marrink. 2003. Simulation of pore formation in lipid bilayers by mechanical stress and electric fields. J Am Chem Soc 125(21):6382-6383.
49. Pastor, R. W., and A. D. MacKerell. 2011. Development of the CHARMM Force Field for Lipids. J Phys Chem Lett 2(13):1526-1532.
50. 2019. CHARMM-GUI. charmm-gui.org. Effective simulation input generator and more.
51. Klauda, J. B., R. M. Venable, J. A. Freites, J. W. O'Connor, D. J. Tobias, C. Mondragon-Ramirez, I. Vorobyov, A. D. MacKerell, and R. W. Pastor. 2010. Update of the CHARMM All-Atom Additive Force Field for Lipids: Validation on Six Lipid Types. J. Phys. Chem. B 114(23):7830-7843.
52. Boonstra, S., P. R. Onck, and E. van der Giessen. 2016. CHARMM TIP3P Water Model Suppresses Peptide Folding by Solvating the Unfolded State. J. Phys. Chem. B 120(15):3692-3698.
53. GROMACS User Manual version 2018 (www.gromacs.org).
54. Miyamoto, S., and P. A. Kollman. 1992. Settle - an Analytical Version of the Shake and Rattle Algorithm for Rigid Water Models. J Comput Chem 13(8):952-962.
55. Berendsen, H. J. C., D. Vanderspoel, and R. Vandrunen. 1995. Gromacs - a Message-Passing Parallel Molecular-Dynamics Implementation. Comput Phys Commun 91(1-3):43-56.
56. Essmann, U., L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. G. Pedersen. 1995. A Smooth Particle Mesh Ewald Method. J Chem Phys 103(19):8577-8593.
57. Hess, B., H. Bekker, H. J. C. Berendsen, and J. G. E. M. Fraaije. 1997. LINCS: A linear constraint solver for molecular simulations. J Comput Chem 18(12):1463-1472.
58. Lindahl, E., B. Hess, and D. van der Spoel. 2001. GROMACS 3.0: a package for molecular simulation and trajectory analysis. J Mol Model 7(8):306-317.
59. Van der Spoel, D., E. Lindahl, B. Hess, G. Groenhof, A. E. Mark, and H. J. C. Berendsen. 2005. GROMACS: Fast, flexible, and free. J Comput Chem 26(16):1701-1718.
60. Hess, B., C. Kutzner, D. van der Spoel, and E. Lindahl. 2008. GROMACS 4: Algorithms for highly efficient, load-balanced, and scalable molecular simulation. Journal of Chemical Theory and Computation 4(3):435-447.
61. Pronk, S., S. Pall, R. Schulz, P. Larsson, P. Bjelkmar, R. Apostolov, M. R. Shirts, J. C. Smith, P. M. Kasson, D. van der Spoel, B. Hess, and E. Lindahl. 2013. GROMACS 4.5: a high-throughput and highly parallel open source molecular simulation toolkit. Bioinformatics 29(7):845-854.
62. Pall, S., M. J. Abraham, C. Kutzner, B. Hess, and E. Lindahl. 2015. Tackling Exascale Software Challenges in Molecular Dynamics Simulations with GROMACS. Lect Notes Comput Sc 8759:3-27.
63. Vanommeslaeghe, K., E. Hatcher, C. Acharya, S. Kundu, S. Zhong, J. Shim, E. Darian, O. Guvench, P. Lopes, I. Vorobyov, and A. D. MacKerell. 2010.

CHARMM General Force Field: A Force Field for Drug-Like Molecules Compatible with the CHARMM All-Atom Additive Biological Force Fields. J Comput Chem 31(4):671-690.
64. Mukherjee, G., K. Vanommeslaeghe, and A. Mackerell. 2016. Ongoing developments in the CHARMM general force field. Abstr Pap Am Chem S 252.
65. 2019. Force fields in GROMACS.
66. Hoover, W. G. 1989. Generalization of Noses Isothermal Molecular-Dynamics - Non-Hamiltonian Dynamics for the Canonical Ensemble. Phys Rev A 40(5):2814-2815.
67. Parrinello, M., and A. Rahman. 1981. Polymorphic Transitions in Single-Crystals - a New Molecular-Dynamics Method. J Appl Phys 52(12):7182-7190.
68. Yoo, J., and A. Aksimentiev. 2018. New tricks for old dogs: improving the accuracy of biomolecular force fields by pair-specific corrections to non-bonded interactions. Phys Chem Chem Phys 20(13):8432-8449.
69. Schlesinger, P. H., N. K. Djedovic, R. Ferdani, J. Pajewska, R. Pajewski, and G. W. Gokel. 2003. Anchor chain length alters the apparent mechanism of chloride channel function in SCMTR derivatives. Chem Commun(3):308-309.
70. Pajewski, R., R. Ferdani, J. Pajewska, N. Djedovic, P. H. Schlesinger, and G. W. Gokel. 2005. Evidence for dimer formation by an amphiphilic heptapeptide that mediates chloride and carboxyfluorescein release from liposomes. Org Biomol Chem 3(4):619-625.
71. Berendsen, H. J. C., J. P. M. Postma, W. F. Vangunsteren, A. Dinola, and J. R. Haak. 1984. Molecular-Dynamics with Coupling to an External Bath. J Chem Phys 81(8):3684-3690.
72. Humphrey, W., A. Dalke, and K. Schulten. 1996. VMD: Visual molecular dynamics. J Mol Graph Model 14(1):33-38.
73. Palankar, R., B. E. Pinchasik, B. N. Khlebtsov, T. A. Kolesnikova, H. Mohwald, M. Winterhalter, and A. G. Skirtach. 2014. Nanoplasmonically-Induced Defects in Lipid Membrane Monitored by Ion Current: Transient Nanopores versus Membrane Rupture. Nano Lett 14(8):4273-4279.
74. Yan, J. Y., S. X. Liu, J. Hu, X. H. Gui, G. L. Wang, and Y. J. Yan. 2011. Enzymatic enrichment of polyunsaturated fatty acids using novel lipase preparations modified by combination of immobilization and fish oil treatment. Bioresource Technol 102(14):7154-7158.
75. Lau, R. M., F. van Rantwijk, K. R. Seddon, and R. A. Sheldon. 2000. Lipase-catalyzed reactions in ionic liquids. Org Lett 2(26):4189-4191.
76. Kim, K. W., B. Song, M. Y. Choi, and M. J. Kim. 2001. Biocatalysis in ionic liquids: Markedly enhanced enantioselectivity of lipase. Org Lett 3(10):1507-1509.
77. Nordwald, E. M., R. Brunecky, M. E. Himmel, G. T. Beckham, and J. L. Kaar. 2014. Charge Engineering of Cellulases Improves Ionic Liquid Tolerance and Reduces Lignin Inhibition. Biotechnol Bioeng 111(8):1541-1549.
78. Brissos, V., N. Goncalves, E. P. Melo, and L. O. Martins. 2014. Improving Kinetic or Thermodynamic Stability of an Azoreductase by Directed Evolution. Plos One 9(1).

79. Pramanik, S., G. V. Dhoke, K. E. Jaeger, U. Schwaneberg, and M. D. Davari. 2019. How To Engineer Ionic Liquids Resistant Enzymes: Insights from Combined Molecular Dynamics and Directed Evolution Study. Acs Sustain Chem Eng 7(13):11293-11302.
80. Zhang, Y., J. Hao, Y. Q. Zhang, X. L. Chen, B. B. Xie, M. Shi, B. C. Zhou, Y. Z. Zhang, and P. Y. Li. 2017. Identification and Characterization of a Novel Salt-Tolerant Esterase from the Deep-Sea Sediment of the South China Sea. Front Microbiol 8.
81. Pedersen, J. N., Y. Zhou, Z. Guo, and B. Perez. 2019. Genetic and chemical approaches for surface charge engineering of enzymes and their applicability in biocatalysis: A review. Biotechnol Bioeng 116(7):1795-1812.
82. Green, A. A. 1931. Studies in the Physical Chemistry of the Protiens: VIII. The solubility of Hemoglobin in concentrated salt solutions. A study of the salting out of proteins. J Biol Chem 93:12.
83. H.M. Berman, J. W., Z. Feng, G. Gilliland, T.N. Bhat, H. Weissig, I.N. Shindyalov, P.E. Bourne. 2000. The Protein Data Bank *Nucleic Acids Research*, 28: 235-242.
84. Kaar, J. L., A. M. Jesionowski, J. A. Berberich, R. Moulton, and A. J. Russell. 2003. Impact of ionic liquid physical properties on lipase activity and stability. J Am Chem Soc 125(14):4125-4131.
85. Jaeger, V. W., and J. Pfaendtner. 2013. Structure, Dynamics, and Activity of Xylanase Solvated in Binary Mixtures of Ionic Liquid and Water. Acs Chem Biol 8(6):1179-1186.
86. Tung, H. J., and J. Pfaendtner. 2016. Kinetics and mechanism of ionic-liquid induced protein unfolding: application to the model protein HP35. Mol Syst Des Eng 1(4):382-390.
87. Hornak, V., R. Abel, A. Okur, B. Strockbine, A. Roitberg, and C. Simmerling. 2006. Comparison of multiple amber force fields and development of improved protein backbone parameters. Proteins 65(3):712-725.
88. Jorgensen, W. L., J. Chandrasekhar, J. D. Madura, R. W. Impey, and M. L. Klein. 1983. Comparison of Simple Potential Functions for Simulating Liquid Water. J Chem Phys 79(2):926-935.
89. Wang, J. M., R. M. Wolf, J. W. Caldwell, P. A. Kollman, and D. A. Case. 2004. Development and testing of a general amber force field. J Comput Chem 25(9):1157-1174.
90. Pfaendtner Jim, J., and M. Bonomi. 2015. Efficient Sampling of High-Dimensional Free-Energy Landscapes with Parallel Bias Metadynamics. Journal of Chemical Theory and Computation 11(11):5062-5067.
91. Parrinello, M. 2016. Path integral metadynamics. Abstr Pap Am Chem S 251.
92. Laio, A., and M. Parrinello. 2006. Computing free energies and accelerating rare events with metadynamics. Lect Notes Phys 703:315-+.
93. Barducci, A., G. Bussi, and M. Parrinello. 2008. Well-tempered metadynamics: A smoothly converging and tunable free-energy method. Phys Rev Lett 100(2).
94. Best, R. B., and G. Hummer. 2009. Optimized Molecular Dynamics Force Fields Applied to the Helix-Coil Transition of Polypeptides. J. Phys. Chem. B 113(26):9004-9015.

95. Klauda, J. B., B. R. Brooks, A. D. MacKerell, R. M. Venable, and R. W. Pastor. 2005. An ab initio study on the torsional surface of alkanes and its effect on molecular simulations of alkanes and a DPPC bilayer. J. Phys. Chem. B 109(11):5300-5311.
96. Berendsen, H. J. C., J. R. Grigera, and T. P. Straatsma. 1987. The Missing Term in Effective Pair Potentials. J Phys Chem-Us 91(24):6269-6271.
97. Gronenborn, A. M., D. R. Filpula, N. Z. Essig, A. Achari, M. Whitlow, P. T. Wingfield, and G. M. Clore. 1991. A Novel, Highly Stable Fold of the Immunoglobulin Binding Domain of Streptococcal Protein-G. Science 253(5020):657-661.
98. Neidigh, J. W., R. M. Fesinmeyer, and N. H. Andersen. 2002. Designing a 20-residue protein. Nat Struct Biol 9(6):425-430.
99. Bonomi, M., D. Branduardi, G. Bussi, C. Camilloni, D. Provasi, P. Raiteri, D. Donadio, F. Marinelli, F. Pietrucci, R. A. Broglia, and M. Parrinello. 2009. PLUMED: A portable plugin for free-energy calculations with molecular dynamics. Comput Phys Commun 180(10):1961-1972.
100. Tribello, G. A., M. Bonomi, D. Branduardi, C. Camilloni, and G. Bussi. 2014. PLUMED 2: New feathers for an old bird. Comput Phys Commun 185(2):604-613.
101. Bonomi, M., G. Bussi, C. Camilloni, G. A. Tribello, P. Banas, A. Barducci, M. Bernetti, P. G. Bolhuis, S. Bottaro, D. Branduardi, R. Capelli, P. Carloni, M. Ceriotti, A. Cesari, H. C. Chen, W. Chen, F. Colizzi, S. De, M. De La Pierre, D. Donadio, V. Drobot, B. Ensing, A. L. Ferguson, M. Filizola, J. S. Fraser, H. H. Fu, P. Gasparotto, F. L. Gervasio, F. Giberti, A. Gil-Ley, T. Giorgino, G. T. Heller, G. M. Hocky, M. Iannuzzi, M. Invernizzi, K. E. Jelfs, A. Jussupow, E. Kirilin, A. Laio, V. Limongelli, K. Lindorff-Larsen, T. Lohr, F. Marinelli, L. Martin-Samos, M. Masetti, R. Meyer, A. Michaelides, C. Molteni, T. Morishita, M. Nava, C. Paissoni, E. Papaleo, M. Parrinello, J. Pfaendtner, P. Piaggi, G. Piccini, A. Pietropaolo, F. Pietrucci, S. Pipolo, D. Provasi, D. Quigley, P. Raiteri, S. Raniolo, J. Rydzewski, M. Salvalaglio, G. C. Sosso, V. Spiwok, J. Sponer, D. W. H. Swenson, P. Tiwary, O. Valsson, M. Vendruscolo, G. A. Voth, and A. White. 2019. Promoting transparency and reproducibility in enhanced molecular simulations. Nature Methods 16(8):670-673.
102. Wafer, L. N. R., W. W. Streicher, and G. I. Makhatadze. 2010. Thermodynamics of the Trp-cage miniprotein unfolding in urea. Proteins 78(6):1376-1381.
103. 2008. Raw Material Reserves. In International Fertilizer Industry Association. Fertilizer Statistics.
104. Danyal, K., A. J. Rasmussen, S. M. Keable, B. S. Inglet, S. Shaw, O. A. Zadvornyy, S. Duval, D. R. Dean, S. Raugei, J. W. Peters, and L. C. Seefeldt. 2015. Fe Protein-Independent Substrate Reduction by Nitrogenase MoFe Protein Variants. Biochemistry-Us 54(15):2456-2462.
105. Seefeldt, L. C., B. M. Hoffman, and D. R. Dean. 2009. Mechanism of Mo-Dependent Nitrogenase. Annu Rev Biochem 78:701-722.
106. Seefeldt, L. C., I. G. Dance, and D. R. Dean. 2004. Substrate interactions with nitrogenase: Fe versus Mo. Biochemistry-Us 43(6):1401-1409.

107. Raugei, S., L. C. Seefeldt, and B. M. Hoffman. 2018. Critical computational analysis illuminates the reductive-elimination mechanism that activates nitrogenase for N-2 reduction. Proceedings of the National Academy of Sciences of the United States of America 115(45):E10521-E10530.
108. Tezcan, F. A., J. T. Kaiser, D. Mustafi, M. Y. Walton, J. B. Howard, and D. C. Rees. 2005. Nitrogenase complexes: Multiple docking sites for a nucleotide switch protein. Science 309(5739):1377-1380.
109. Owens, C. P., F. E. H. Katz, C. H. Carter, M. A. Luca, and F. A. Tezcan. 2015. Evidence for Functionally Relevant Encounter Complexes in Nitrogenase Catalysis. J Am Chem Soc 137(39):12704-12712.
110. Roth, L. E., J. C. Nguyen, and F. A. Tezcan. 2010. ATP- and Iron-Protein-Independent Activation of Nitrogenase Catalysis by Light. J Am Chem Soc 132(39):13672-13674.
111. Hurst, L. D., and N. G. C. Smith. 1998. The evolution of concerted evolution. P Roy Soc B-Biol Sci 265(1391):121-127.
112. Corbett, M. C., F. A. Tezcan, O. Einsle, M. Y. Walton, D. C. Rees, M. J. Latimer, B. Hedman, and K. O. Hodgson. 2005. MoK- and L-edge X-ray absorption spectroscopic study of the ADP center dot AlF(4)(-)-stabilized nitrogenase complex: comparison with MoFe protein in solution and single crystal. J Synchrotron Radiat 12:28-34.
113. Khadka, N., R. D. Milton, S. Shaw, D. Lukoyanov, D. R. Dean, S. D. Minteer, S. Raugei, B. M. Hoffrnan, and L. C. Seefeldt. 2017. Mechanism of Nitrogenase H-2 Formation by Metal-Hydride Protonation Probed by Mediated Electrocatalysis and H/D Isotope Effects. J Am Chem Soc 139(38):13518-13524.
114. deGroot, B. L., D. M. F. vanAalten, A. Amadei, and H. J. C. Berendsen. 1996. The consistency of large concerted motions in proteins in molecular dynamics simulations. Biophys J 71(4):1707-1713.
115. Tezcan, F. A., J. T. Kaiser, J. B. Howard, and D. C. Rees. 2015. Structural evidence for asymmetrical nucleotide interactions in nitrogenase. J Am Chem Soc 137(1):146-149.
116. Lindorff-Larsen, K., S. Piana, K. Palmo, P. Maragakis, J. L. Klepeis, R. O. Dror, and D. E. Shaw. 2010. Improved side-chain torsion potentials for the Amber ff99SB protein force field. Proteins 78(8):1950-1958.
117. Joung, I. S., and T. E. Cheatham. 2008. Determination of alkali and halide monovalent ion parameters for use in explicitly solvated biomolecular simulations. In J. Phys. Chem. B. 9020-9041.
118. Kohn, W., and C. D. Sherrill. 2014. Editorial: Reflections on fifty years of density functional theory. J Chem Phys 140(18).
119. Kohn, W. 1999. Thoughts about density functional theory in 1998. J Comput Chem 20(1):1-1.
120. Kohn, W. 1983. Upsilon-Representability and Density Functional Theory. Phys Rev Lett 51(17):1596-1598.
121. Becke, A. D. 1988. Non-Local Density Functional Theories in Quantum-Chemistry - an Overview. Abstr Pap Am Chem S 195:36-Phys.

122. Lee, C. T., G. Fitzgerald, and W. T. Yang. 1993. Nonlocal Density Functional Calculations - Comparison of 2 Implementation Schemes. J Chem Phys 98(4):2971-2974.
123. Pritchard, B. P., D. Altarawy, B. Didier, T. D. Gibson, and T. L. Windus. 2019. New Basis Set Exchange: An Open, Up-to-Date Resource for the Molecular Sciences Community. J Chem Inf Model 59(11):4814-4820.
124. Wadt, W. R., P. J. Hay, and L. R. Kahn. 1978. Relativistic and Nonrelativistic Effective Core Potentials for Xenon - Applications to Xef, Xe2, and Xe2+. J Chem Phys 68(4):1752-1759.
125. *Gaussian 09* (Gaussian, Inc., Wallingford CT).
126. Pople, J. A., J. S. Binkley, R. Krishnan, R. Seeger, and R. Whiteside. 1977. Theoretical Computation of Properties of Small Molecules - Dependence on Basis Set and Level of Treatment of Correlation Energy. Abstr Pap Am Chem S 174(Sep):82-82.
127. Feller, D. 2008. aug-cc-pVDZ-PP.
128. Wang, J. M., W. Wang, and P. A. Kollman. 2001. Antechamber: An accessory software package for molecular mechanical calculations. Abstr Pap Am Chem S 222:U403-U403.
129. O'Boyle, N. M., M. Banck, C. A. James, C. Morley, T. Vandermeersch, and G. R. Hutchison. 2011. Open Babel: An open chemical toolbox. J Cheminformatics 3.
130. Sousa da Silva, A. W., and W. F. Vranken. 2012. ACPYPE - AnteChamber PYthon Parser interfacE. BMC Research Notes 5(1):367.
131. Yachandra, V. K., J. Hare, A. Gewirth, R. S. Czernuszewicz, T. Kimura, R. H. Holm, and T. G. Spiro. 1983. Resonance Raman-Spectra of Spinach Ferredoxin and Adrenodoxin and of Analog Complexes. J Am Chem Soc 105(21):6462-6468.
132. Maiorov, V. N., and G. M. Crippen. 1995. Size-Independent Comparison of Protein 3-Dimensional Structures. Proteins-Structure Function and Genetics 22(3):273-283.
133. Dahl, A. C. E., M. Chavent, and M. S. P. Sansom. 2012. Bendix: intuitive helix geometry analysis and abstraction. Bioinformatics 28(16):2193-2194.
134. Sasaki, D. Y., T. A. Waggoner, J. A. Last, and P. G. Kotula. 2001. Stacked lipid bilayer super-supramolecular structures. Abstr Pap Am Chem S 221:U340-U340.

APPENDIX

TABLE OF CONTENTS

Appendix A

Section 1: Production Run Simulation Histories

All production run history plots

For each charge imbalance, there are 48 plots. The first 24 are from simulations where the SCMTR molecules are not associated with the transmembrane pore. The second set of 24 are from simulations where the SMTR molecules are associated/present in the transmembrane pore.

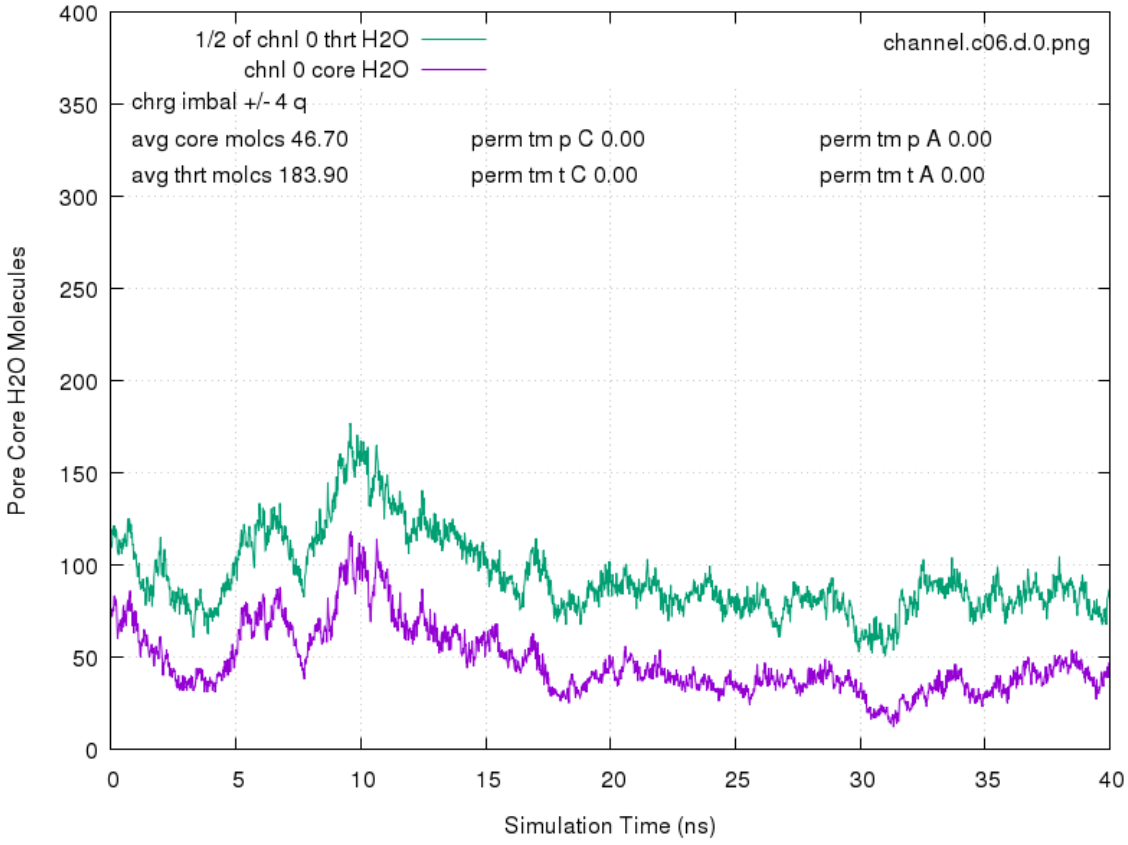
The core water plot (purple) shows the water molecule count in the central 1.5 nm of the pore. The throat (abbr. thrt & t) water plot (evergreen) shows the water molecule count in the central 3.0 nm of the pore. Note that the throat water molecules are plotted at half scale. To accurately read the water molecules value represented by the evergreen trace, the value from the left-hand scale must be doubled.

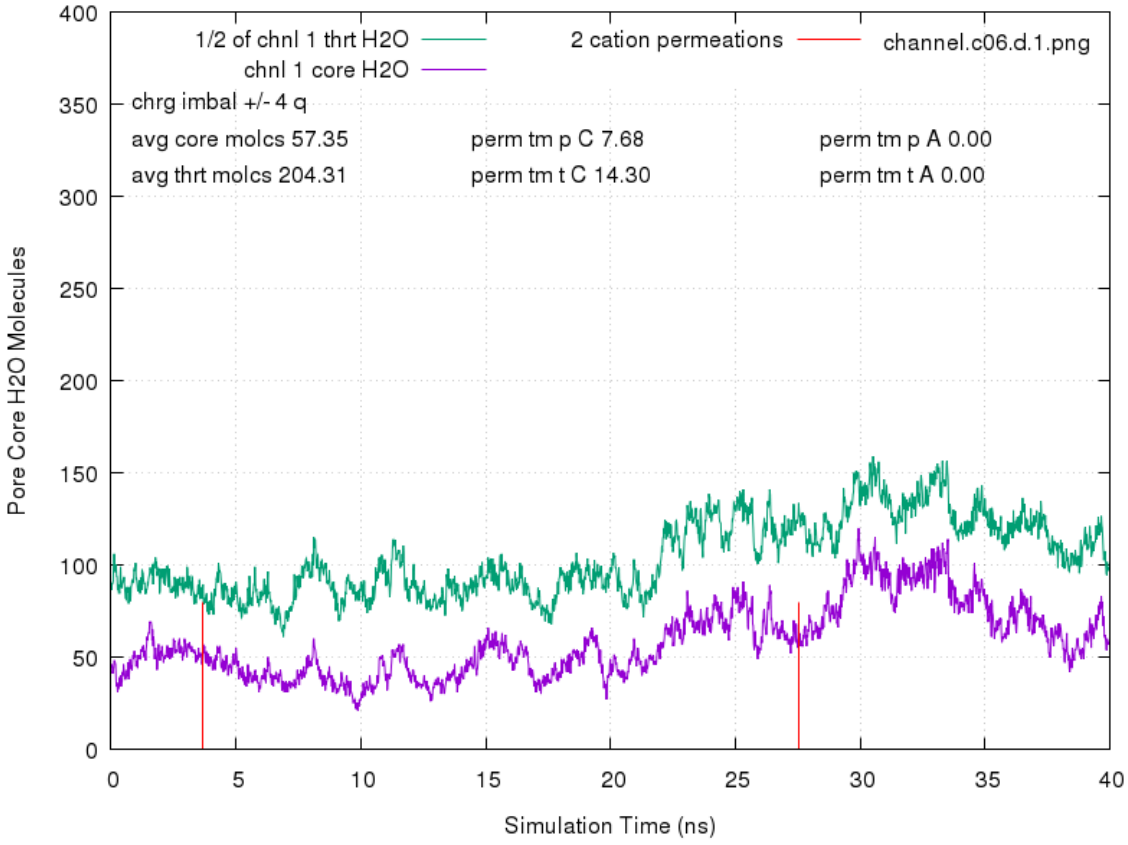
The key will also include the number of ion permeations if there are any. Ion permeation events times are indicated by the vertical red and green lines at the bottom of

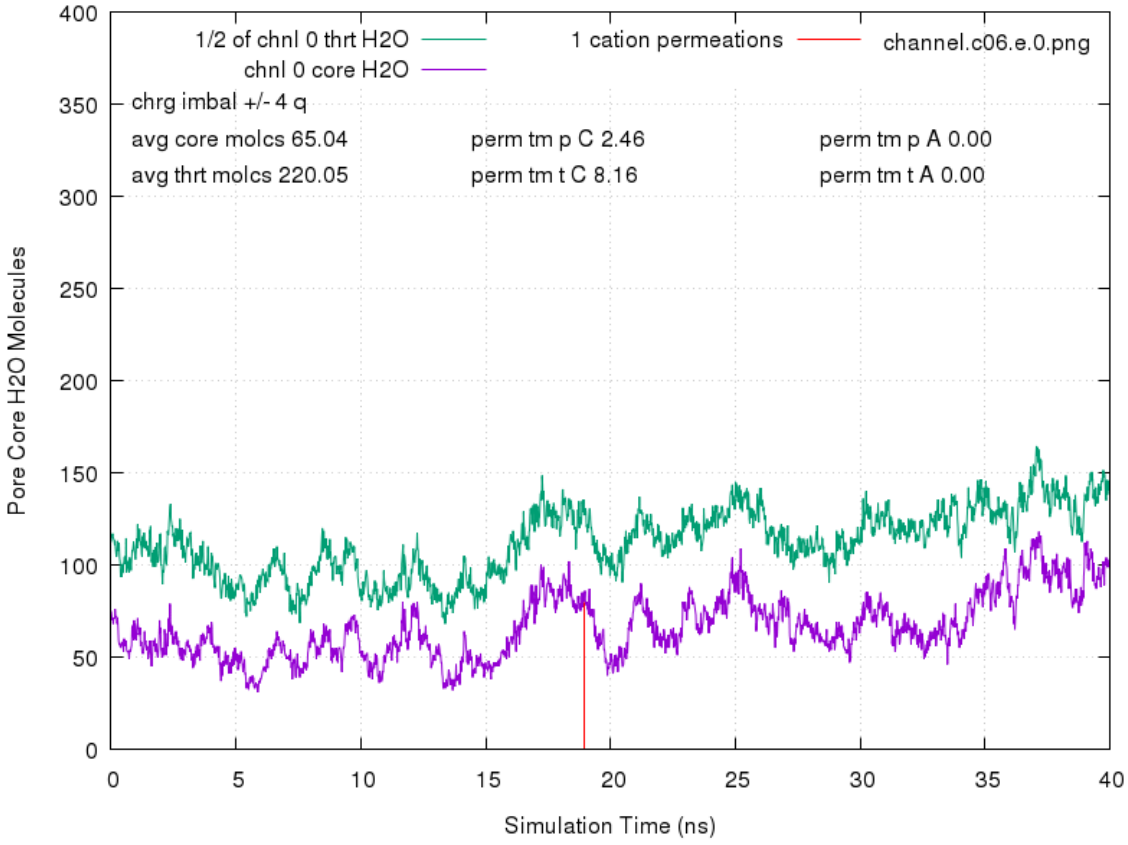
the graph. The green lines (which are plotted after the red lines) are plotted at half height so that an underlying red line will not be totally obscured. Multiple red or green lines have occasionally obscured another of the same color. A few plots will have a larger number of ions shown in the key than the corresponding vertical lines.

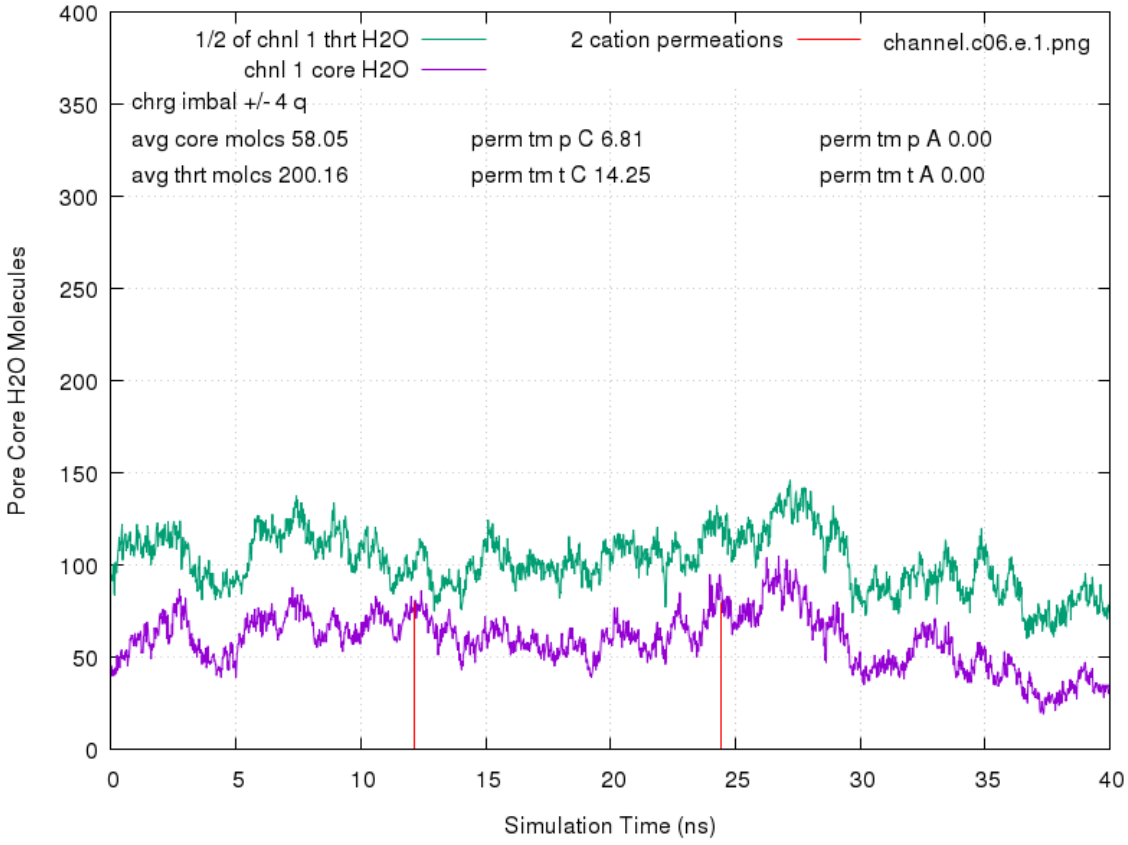
Below the key are the simulation charge imbalance followed by the average pore and throat water molecule counts. Following this are two sets of numbers, first for the average pore and throat Cation permeation times and second set is the same pair for Anions. These represent the average, total time each ion spends in the throat and pore regions for each permeation during the simulation. The units are ns/permeation.

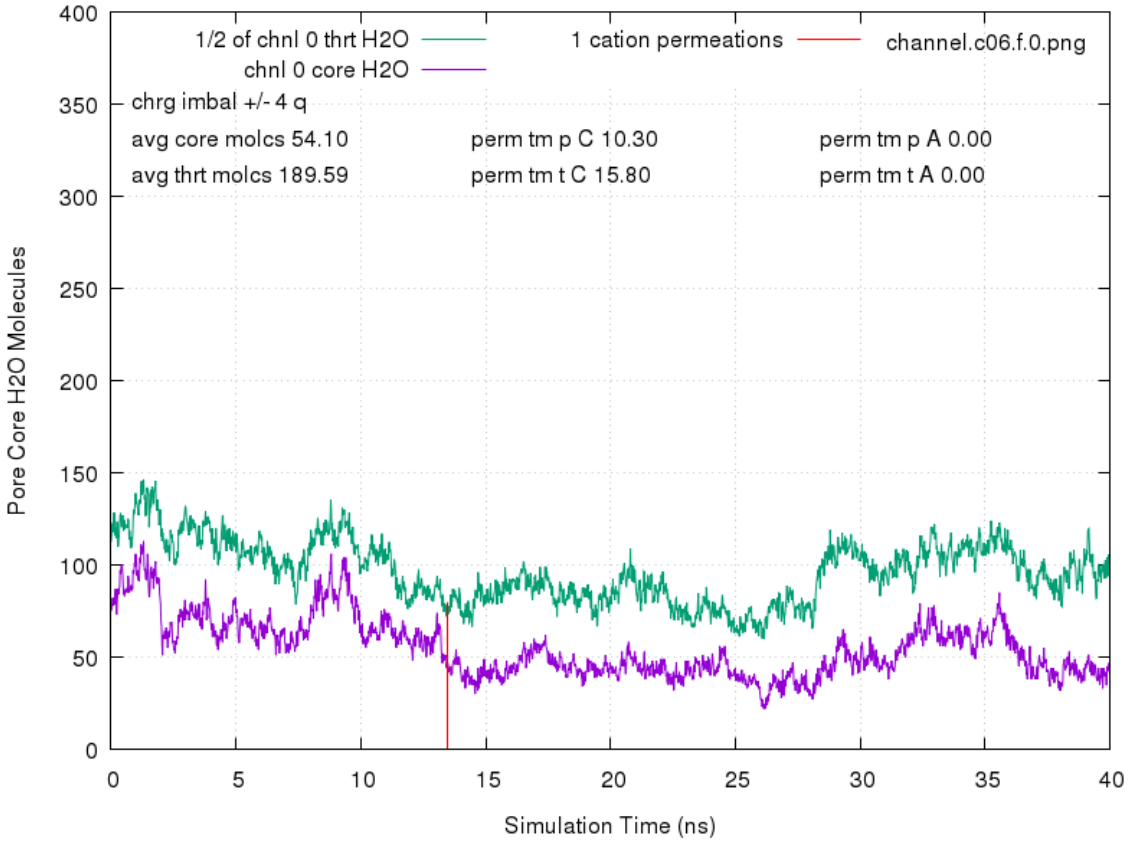
Plots where the purple trace decays at a point to a value of zero represent the data from a membrane pore that has totally disintegrated.

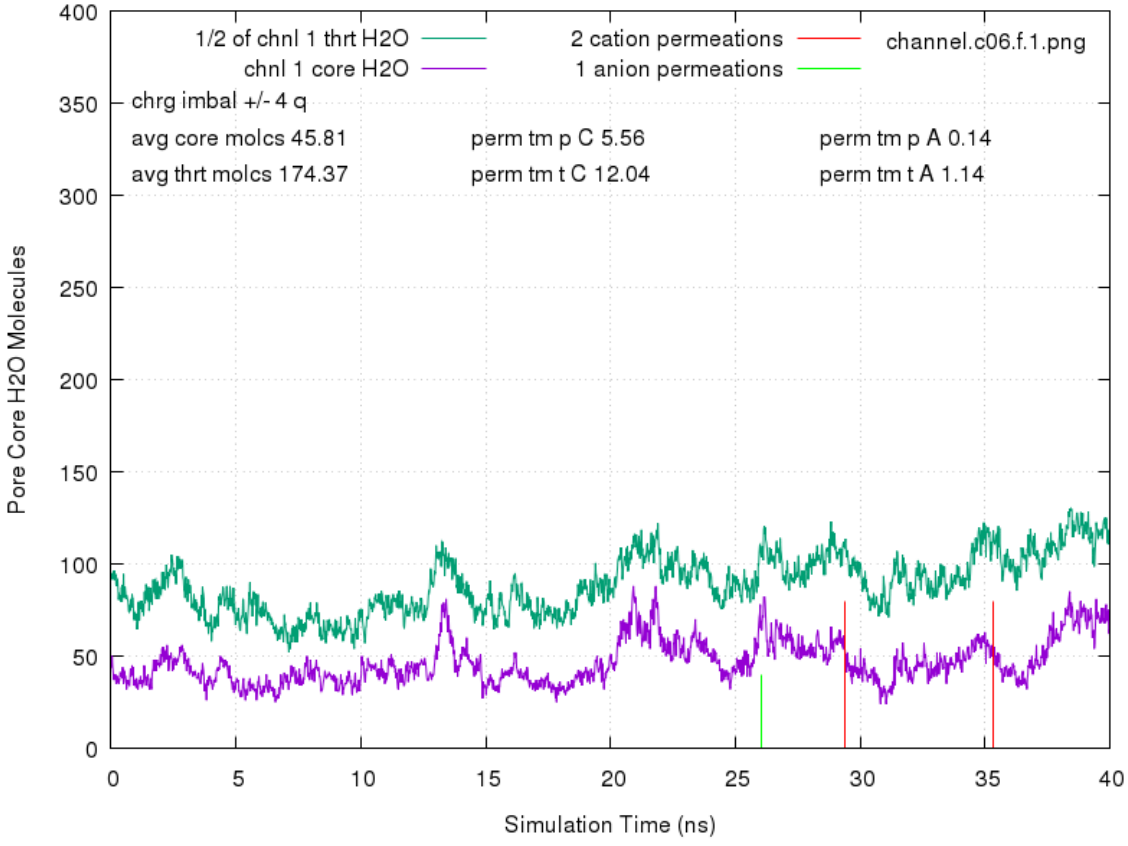

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c06.d.0.png
chrg imbal +/- 4 q
avg core molcs 46.70
avg thrt molcs 183.90
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

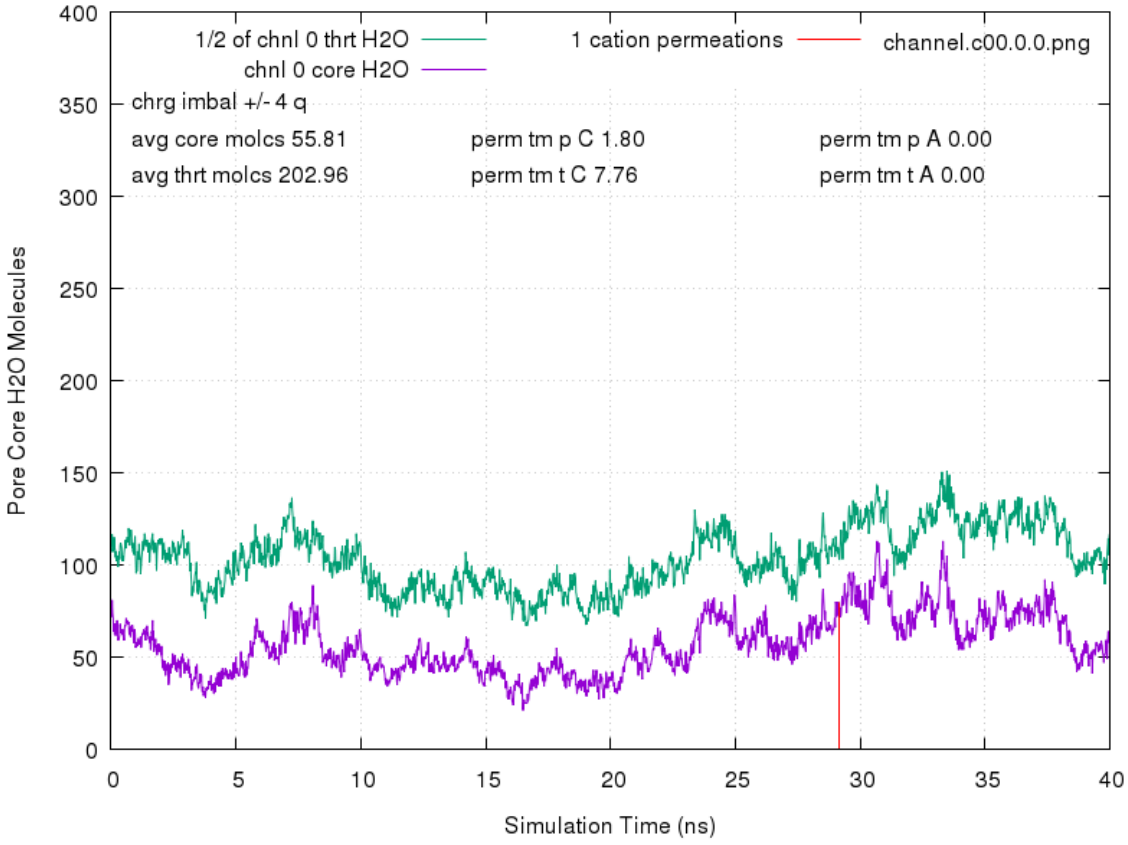

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
channel.c06.d.1.png
chrg imbal +/- 4 q
avg core molcs 57.35
avg thrt molcs 204.31
perm tm p C 7.68
perm tm t C 14.30
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

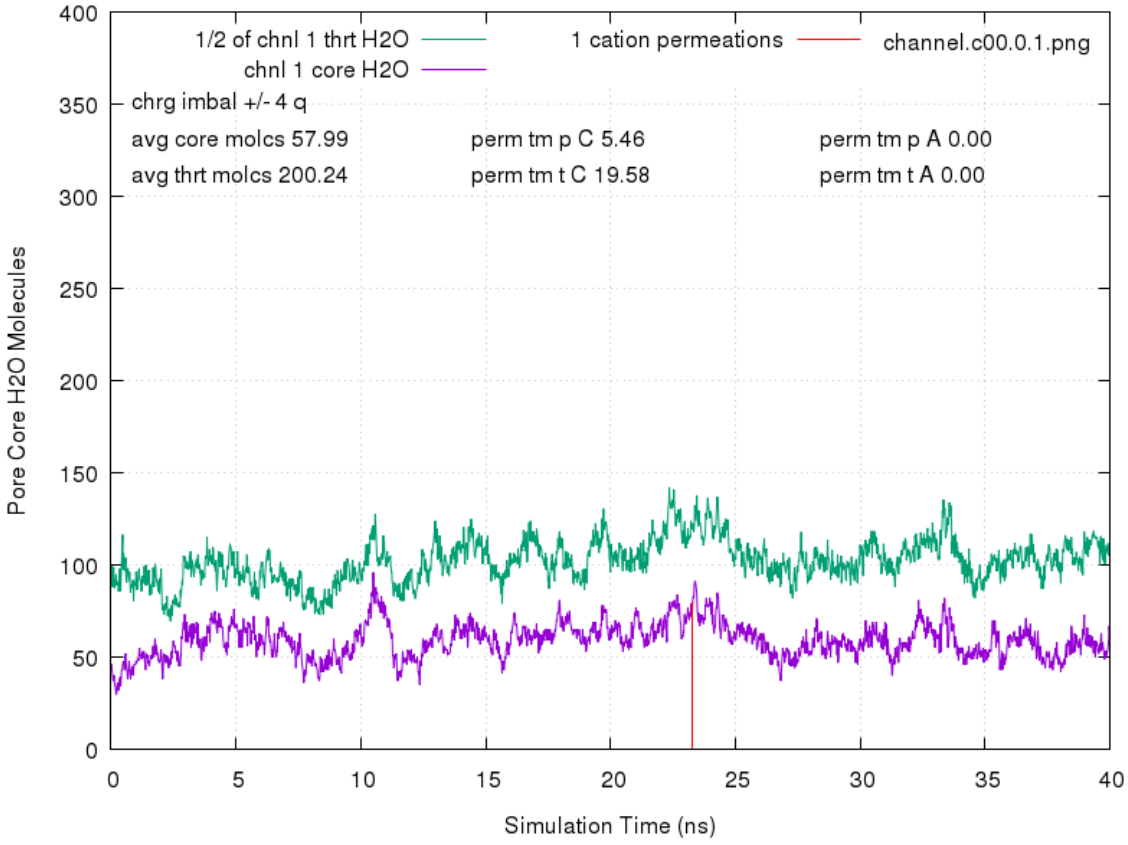

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c06.e.0.png
chrg imbal +/- 4 q
avg core molcs 65.04
avg thrt molcs 220.05
perm tm p C 2.46
perm tm t C 8.16
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

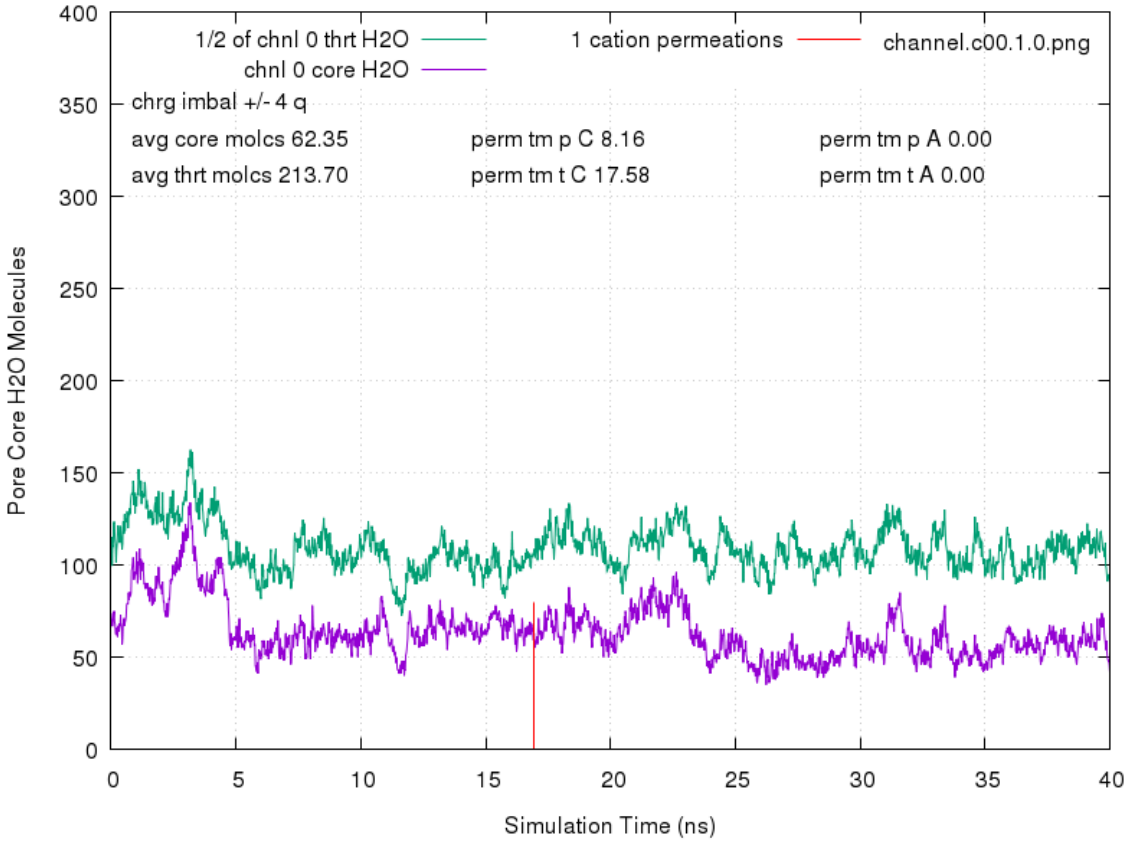

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
channel.c06.e.1.png
chrg imbal +/- 4 q
avg core molcs 58.05
avg thrt molcs 200.16
perm tm p C 6.81
perm tm t C 14.25
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

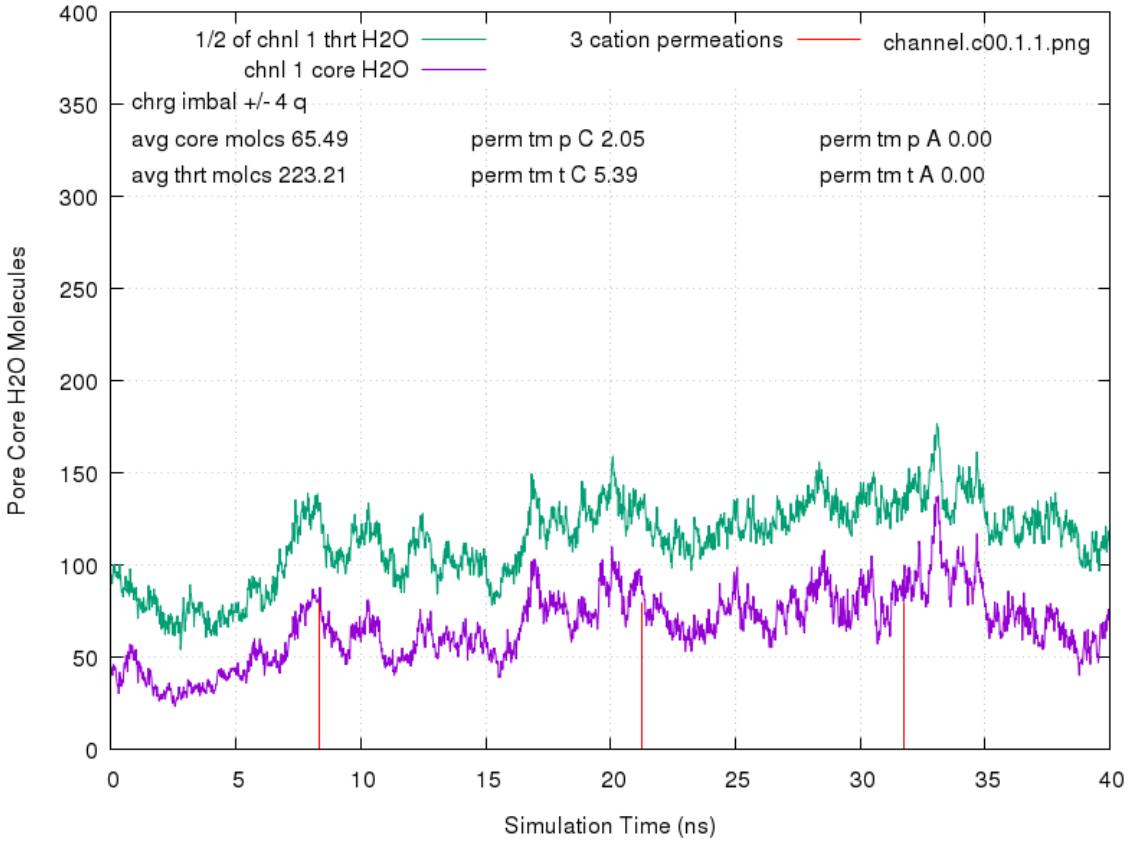

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c06.f.0.png
chrg imbal +/- 4 q
avg core molcs 54.10
avg thrt molcs 189.59
perm tm p C 10.30
perm tm t C 15.80
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

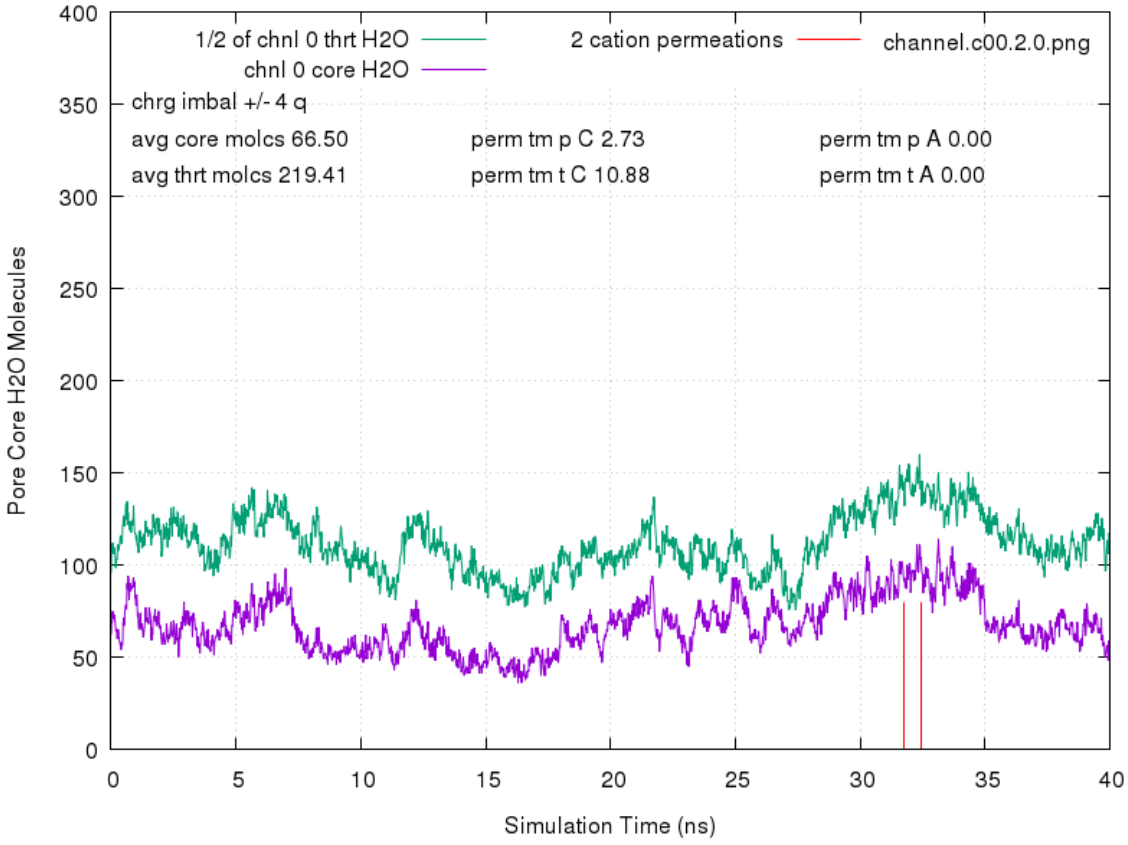

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
1 anion permeations
channel.c06.f.1.png
chrg imbal +/- 4 q
avg core molcs 45.81
avg thrt molcs 174.37
perm tm p C 5.56
perm tm t C 12.04
perm tm p A 0.14
perm tm t A 1.14
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

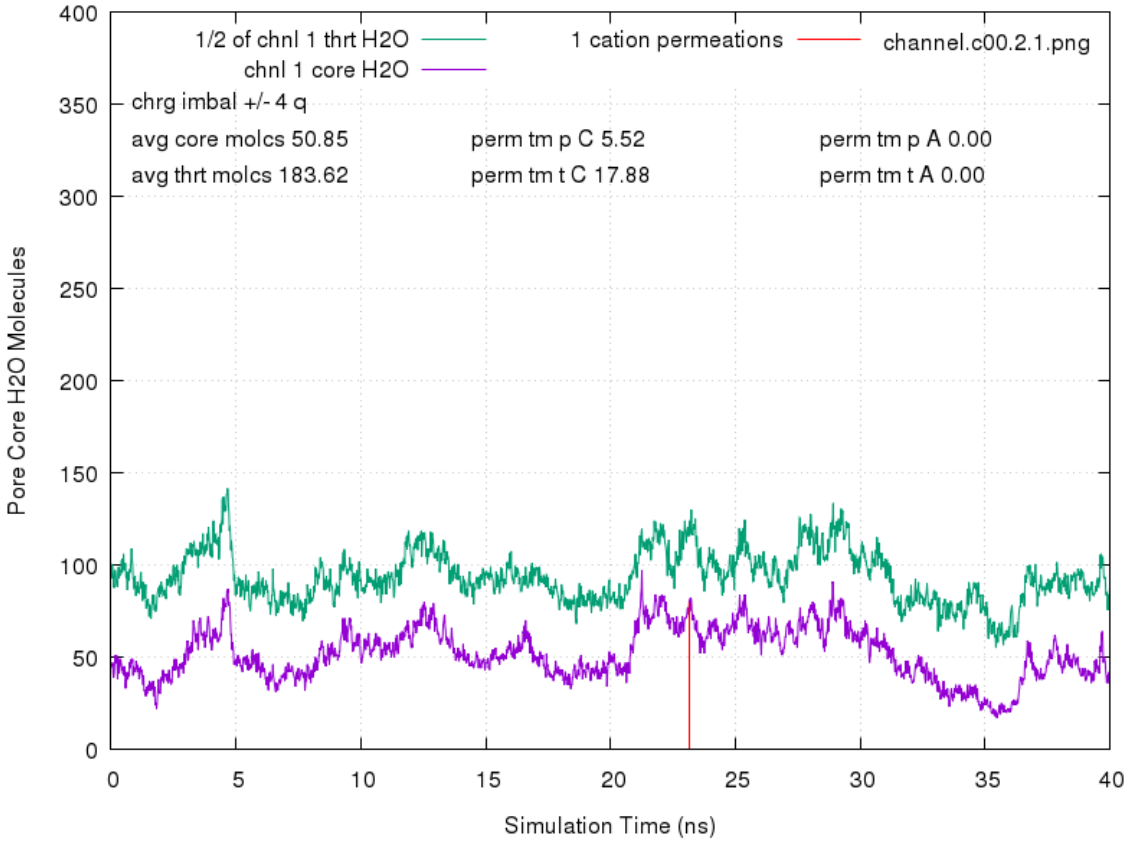

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c00.0.0.png
chrg imbal +/- 4 q
avg core molcs 55.81
avg thrt molcs 202.96
perm tm p C 1.80
perm tm t C 7.76
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

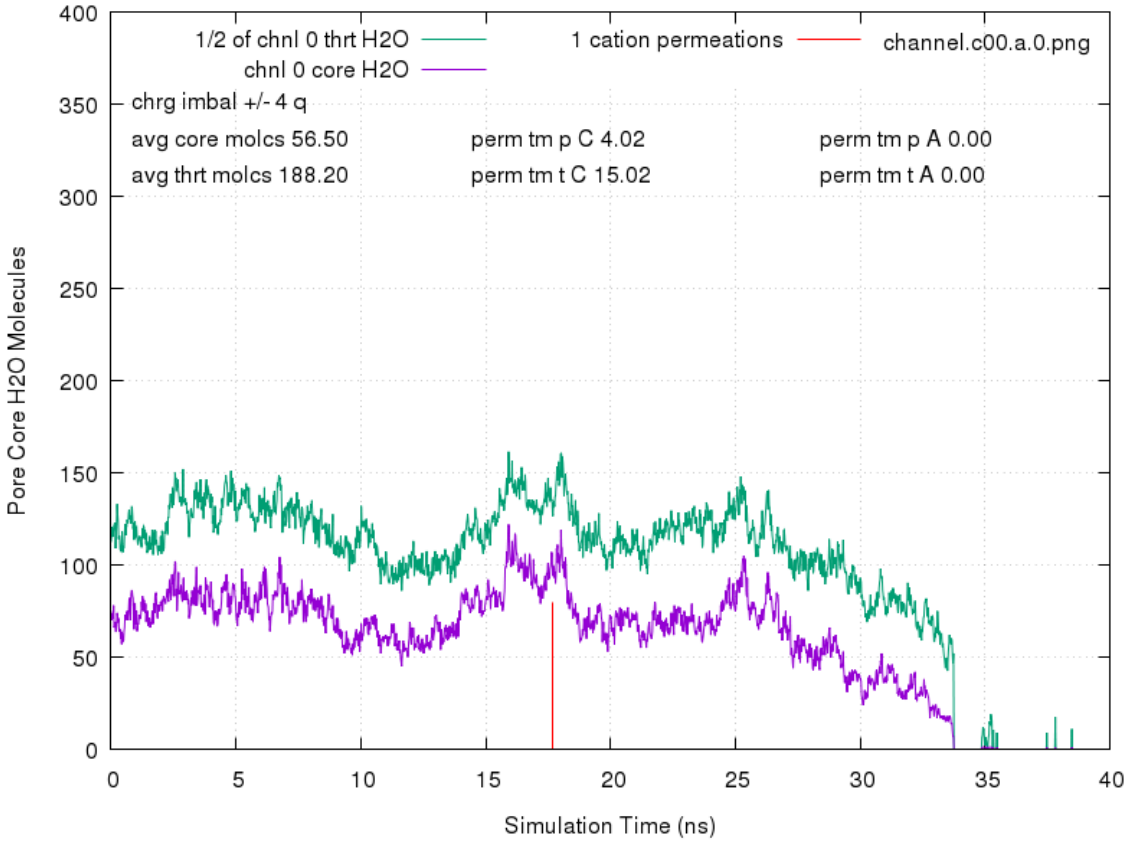

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c00.0.1.png
chrg imbal +/- 4 q
avg core molcs 57.99
avg thrt molcs 200.24
perm tm p C 5.46
perm tm t C 19.58
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

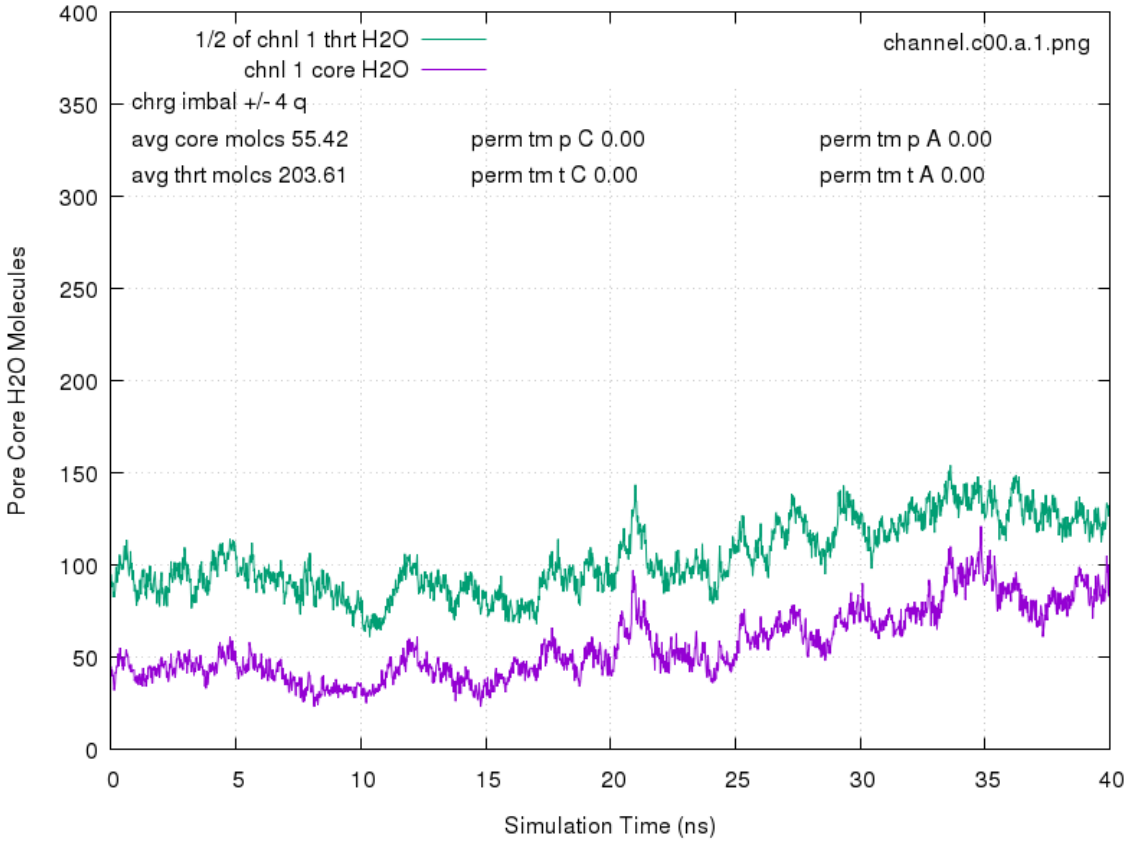

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c00.1.0.png
chrg imbal +/- 4 q
avg core molcs 62.35
avg thrt molcs 213.70
perm tm p C 8.16
perm tm t C 17.58
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

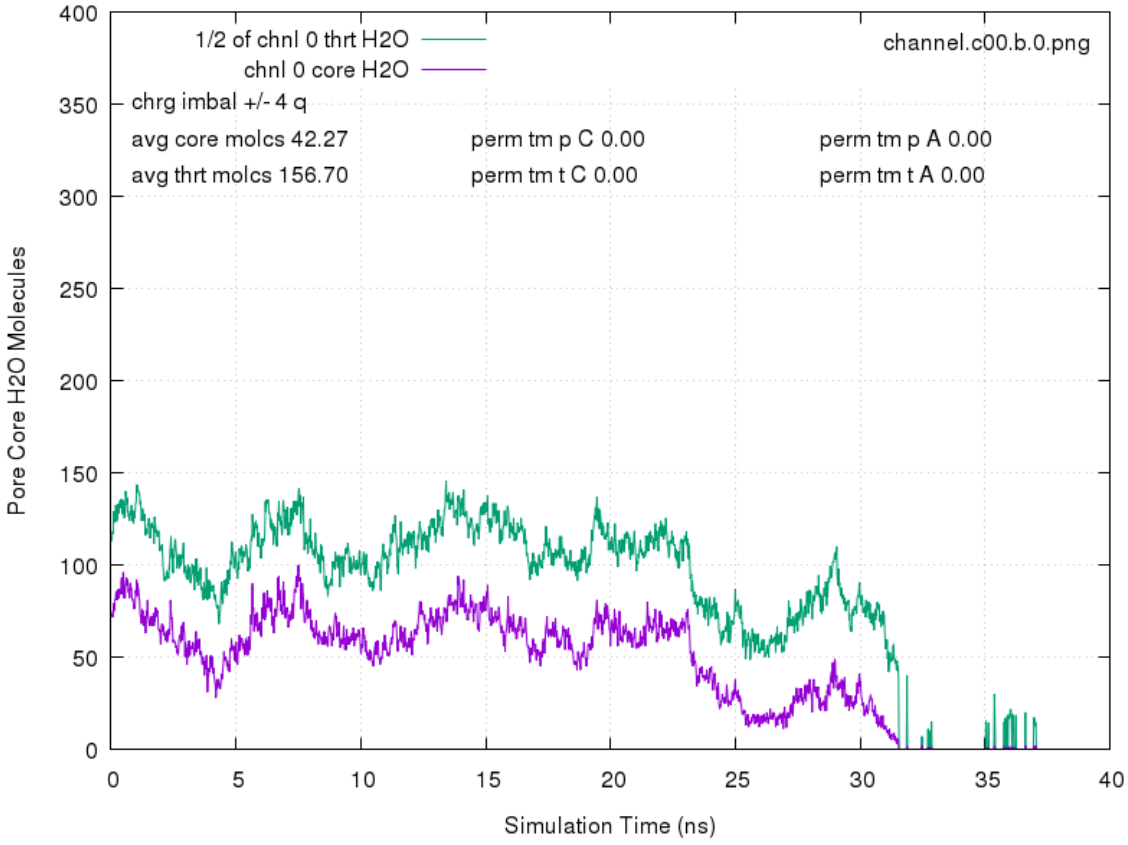

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
channel.c00.1.1.png
chrg imbal +/- 4 q
avg core molcs 65.49
avg thrt molcs 223.21
perm tm p C 2.05
perm tm t C 5.39
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

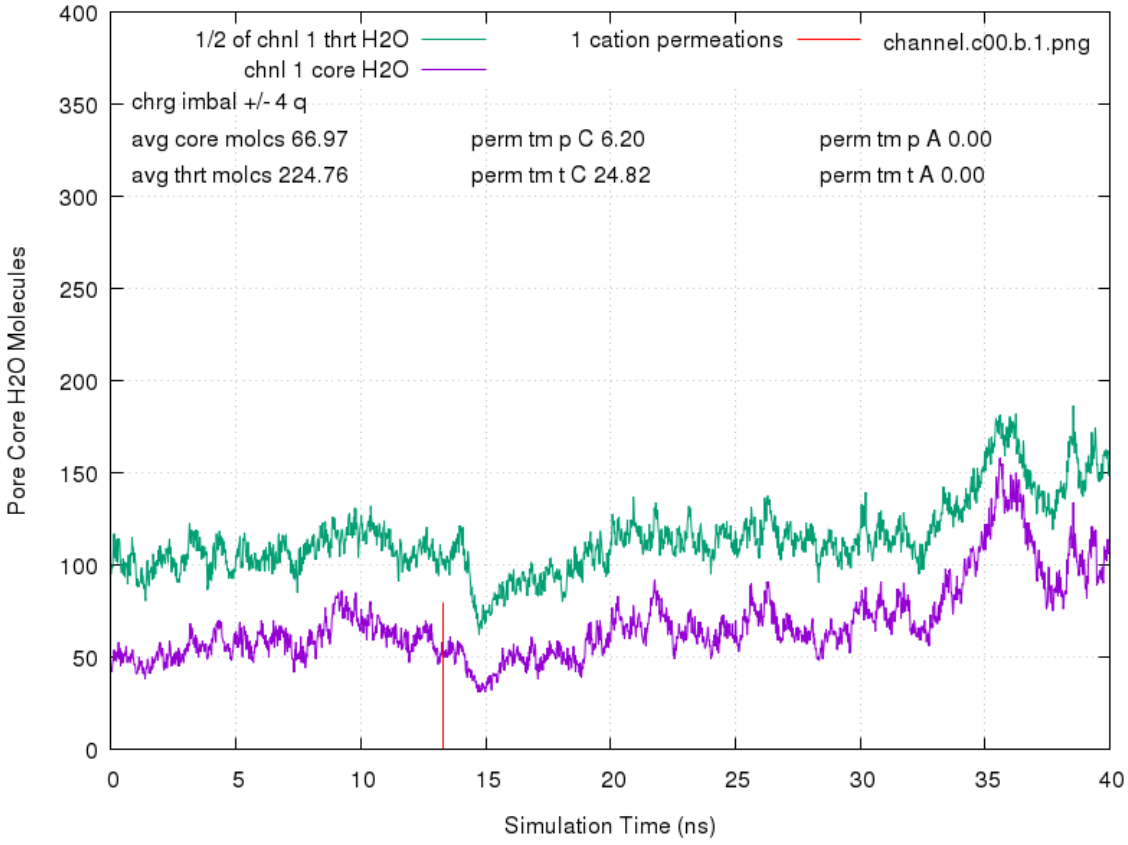

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
channel.c00.2.0.png
chrg imbal +/- 4 q
avg core molcs 66.50
avg thrt molcs 219.41
perm tm p C 2.73
perm tm t C 10.88
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

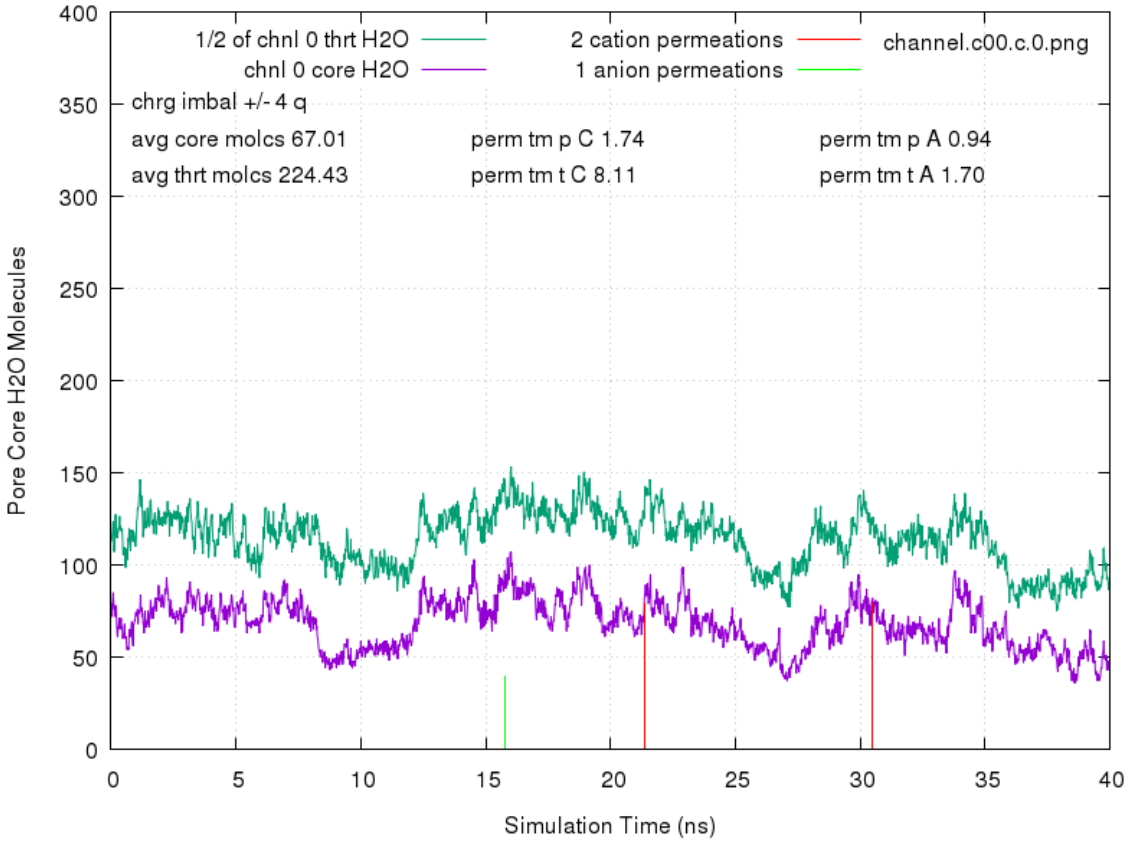

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c00.2.1.png
chrg imbal +/- 4 q
avg core molcs 50.85
avg thrt molcs 183.62
perm tm p C 5.52
perm tm t C 17.88
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

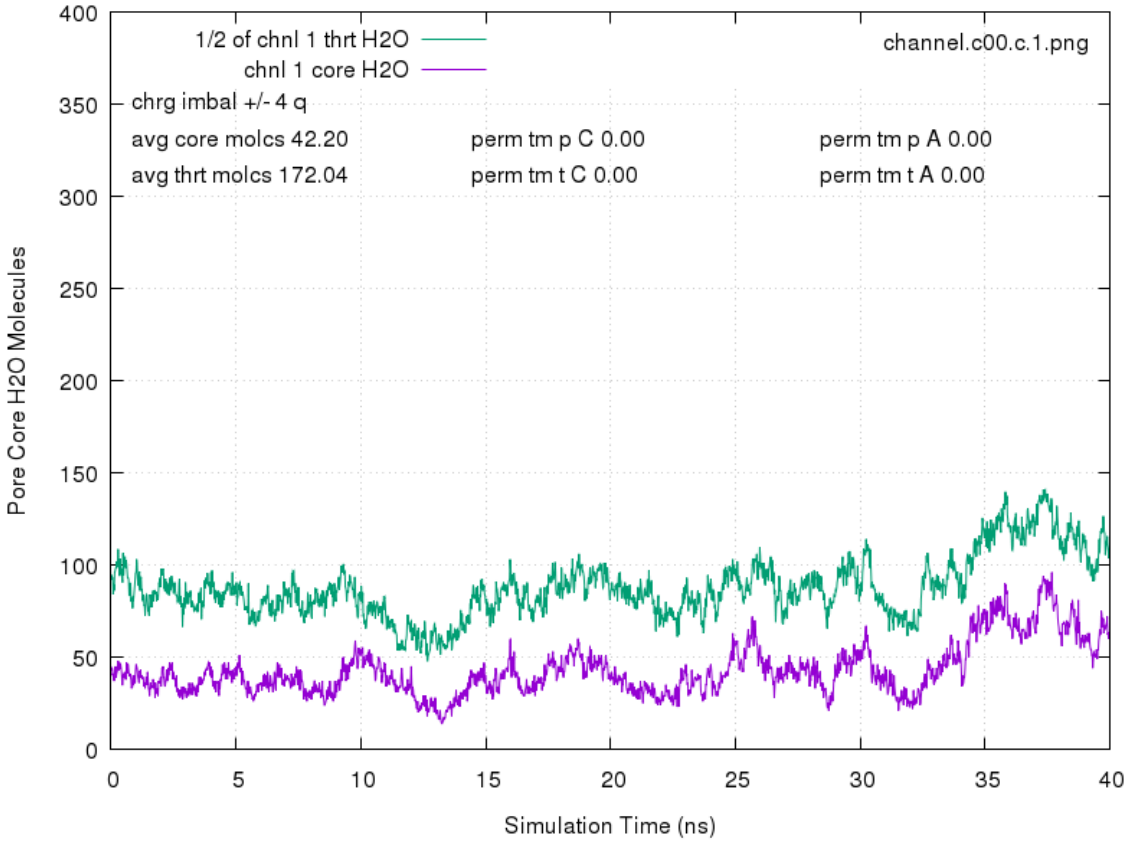

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c00.a.0.png
chrg imbal +/- 4 q
avg core molcs 56.50
avg thrt molcs 188.20
perm tm p C 4.02
perm tm t C 15.02
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

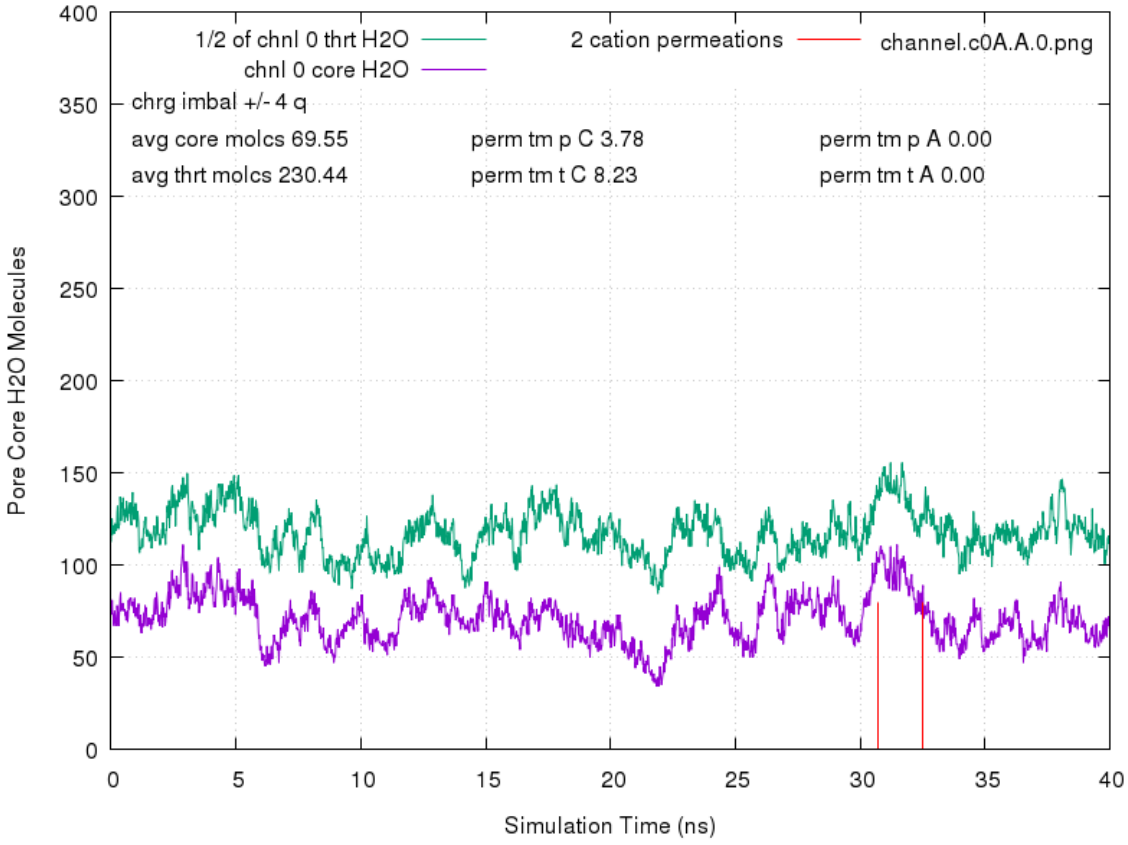

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c00.a.1.png
chrg imbal +/- 4 q
avg core molcs 55.42
avg thrt molcs 203.61
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

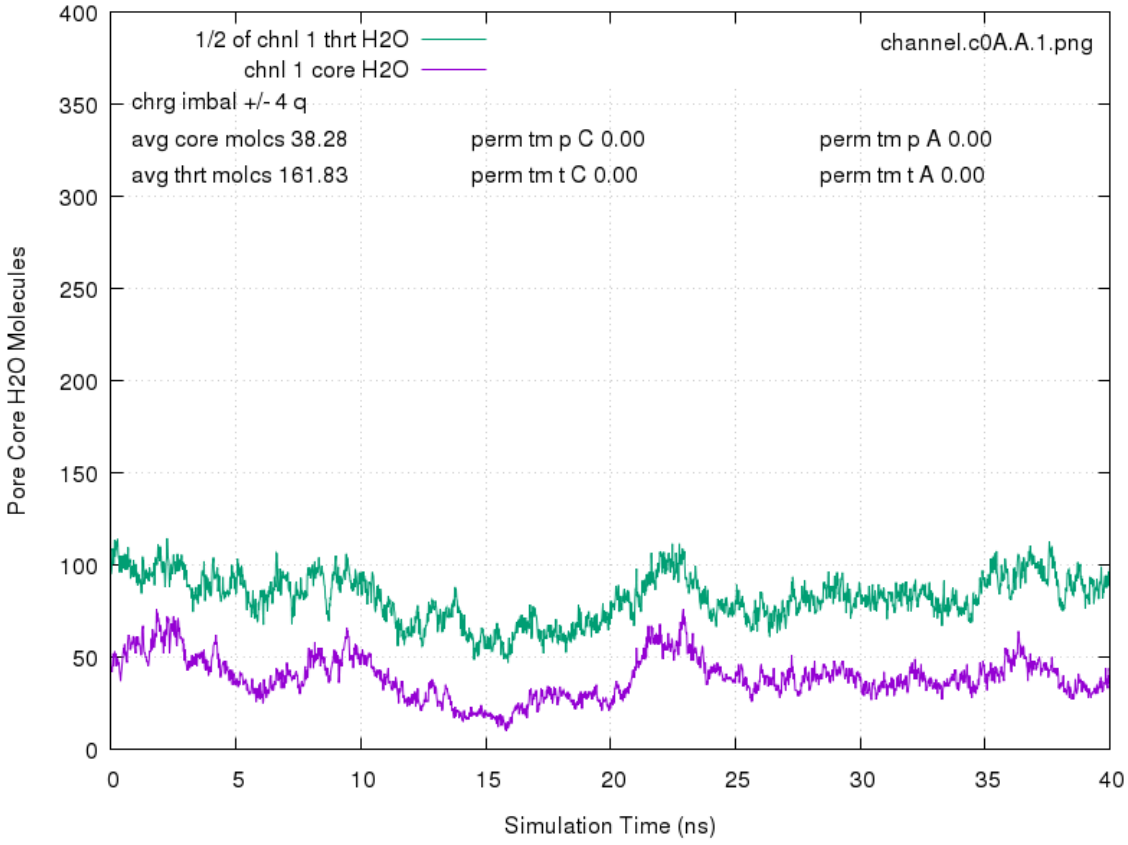

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c00.b.0.png
chrg imbal +/- 4 q
avg core molcs 42.27
avg thrt molcs 156.70
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

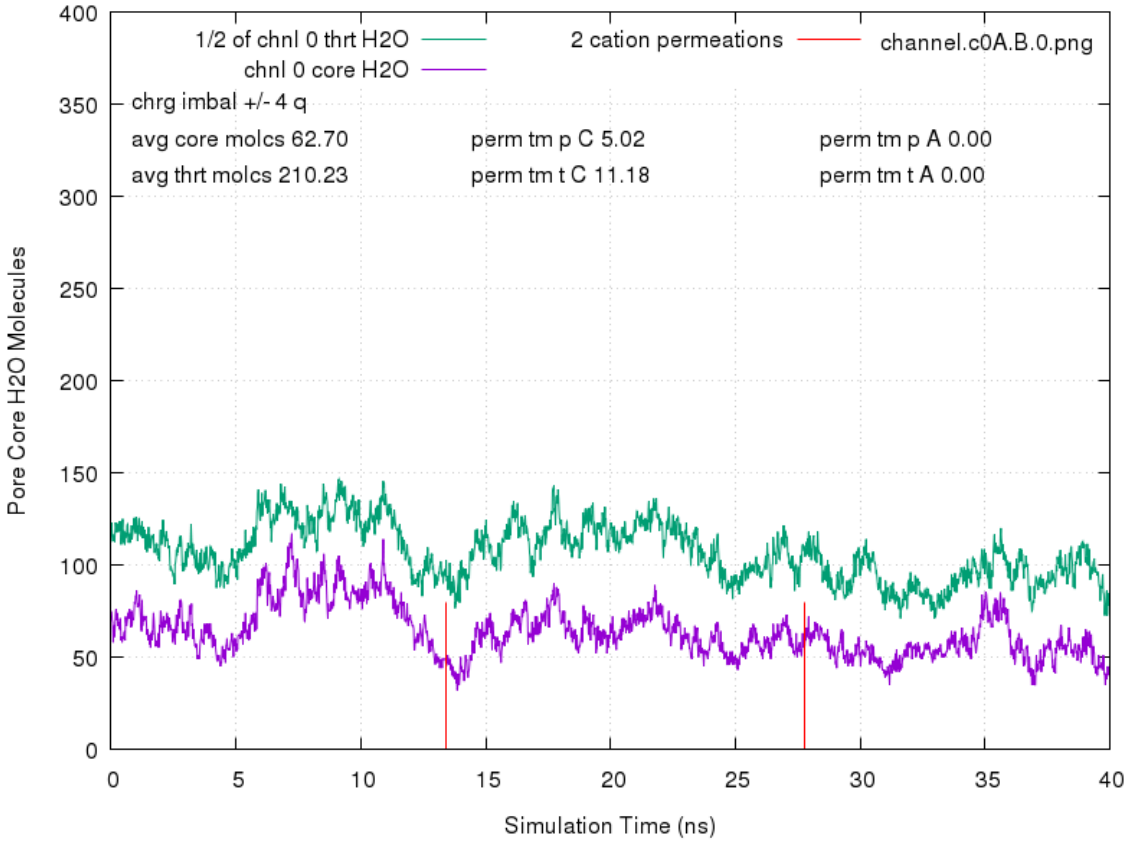

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c00.b.1.png
chrg imbal +/- 4 q
avg core molcs 66.97
avg thrt molcs 224.76
perm tm p C 6.20
perm tm t C 24.82
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

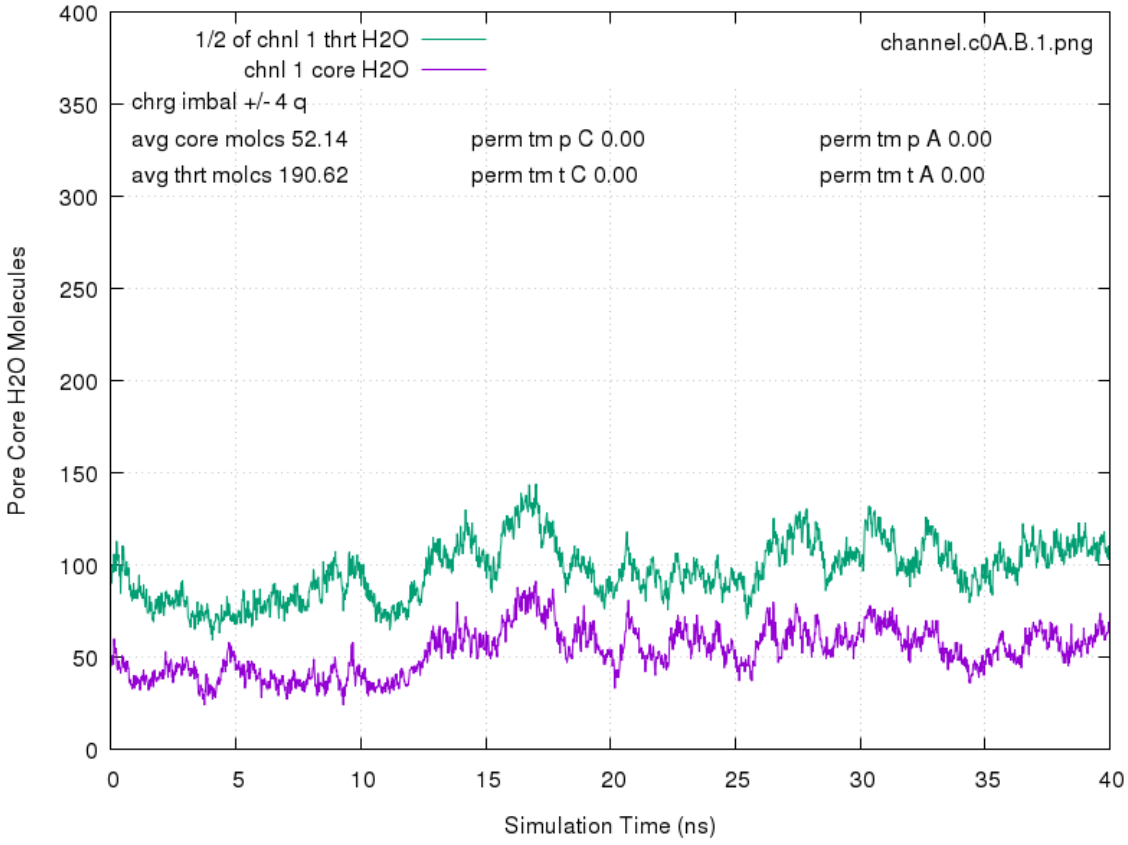

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
1 anion permeations
channel.c00.c.0.png
chrg imbal +/- 4 q
avg core molcs 67.01
avg thrt molcs 224.43
perm tm p C 1.74
perm tm t C 8.11
perm tm p A 0.94
perm tm t A 1.70
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

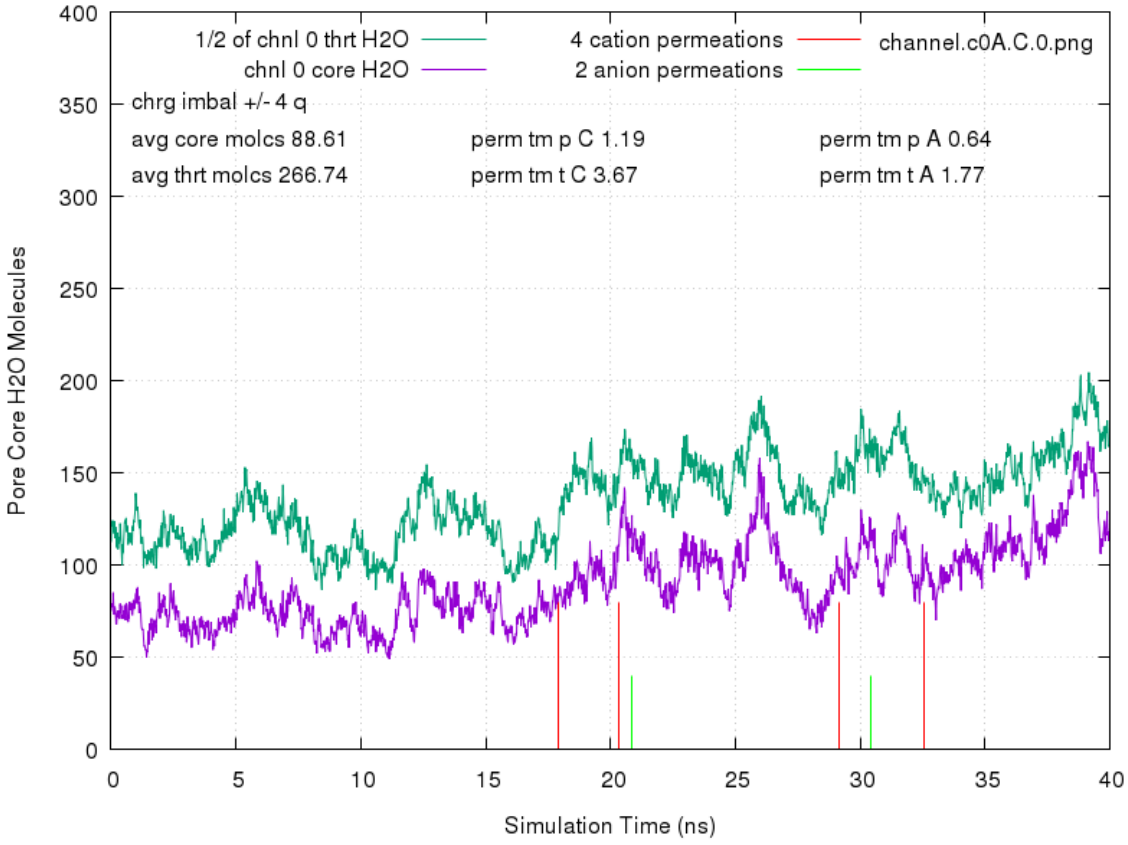

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c00.c.1.png
chrg imbal +/- 4 q
avg core molcs 42.20
avg thrt molcs 172.04
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

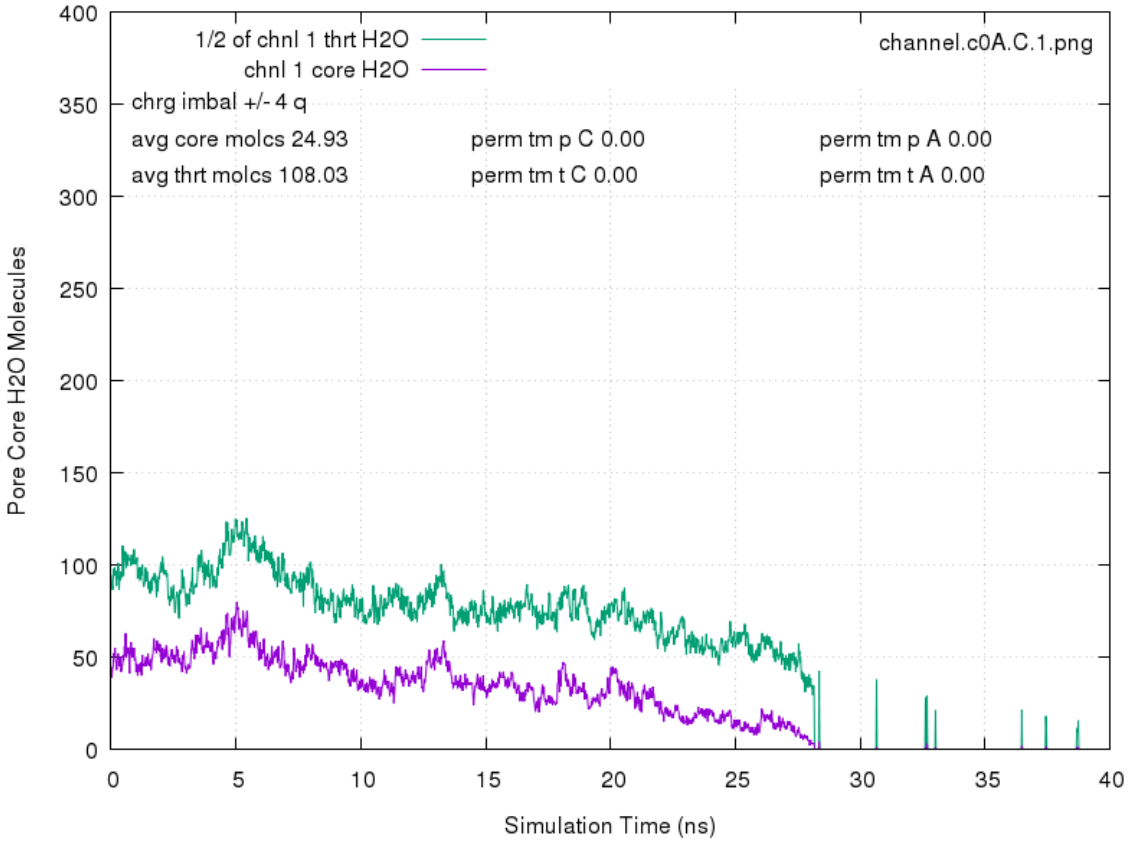

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
channel.c0A.A.0.png
chrg imbal +/- 4 q
avg core molcs 69.55
avg thrt molcs 230.44
perm tm p C 3.78
perm tm t C 8.23
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

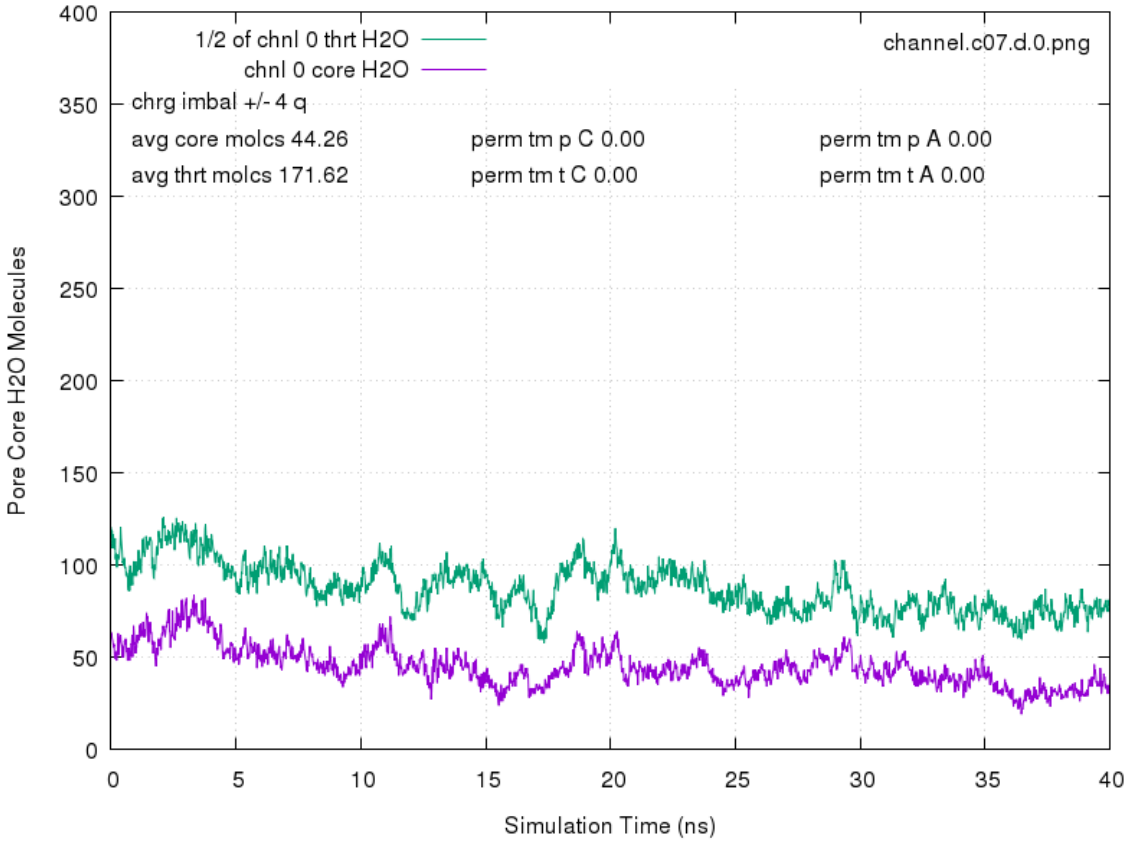

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c0A.A.1.png
chrg imbal +/- 4 q
avg core molcs 38.28
avg thrt molcs 161.83
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

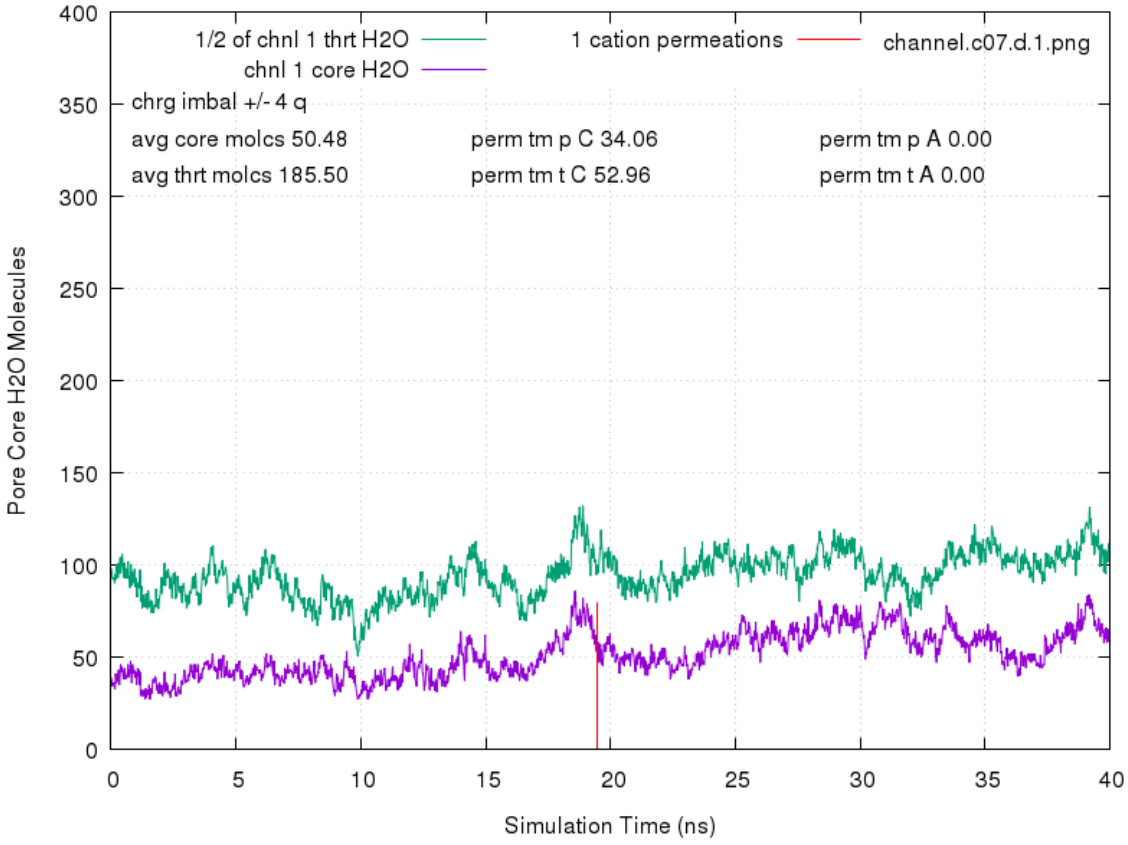

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
channel.c0A.B.0.png
chrg imbal +/- 4 q
avg core molcs 62.70
avg thrt molcs 210.23
perm tm p C 5.02
perm tm t C 11.18
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

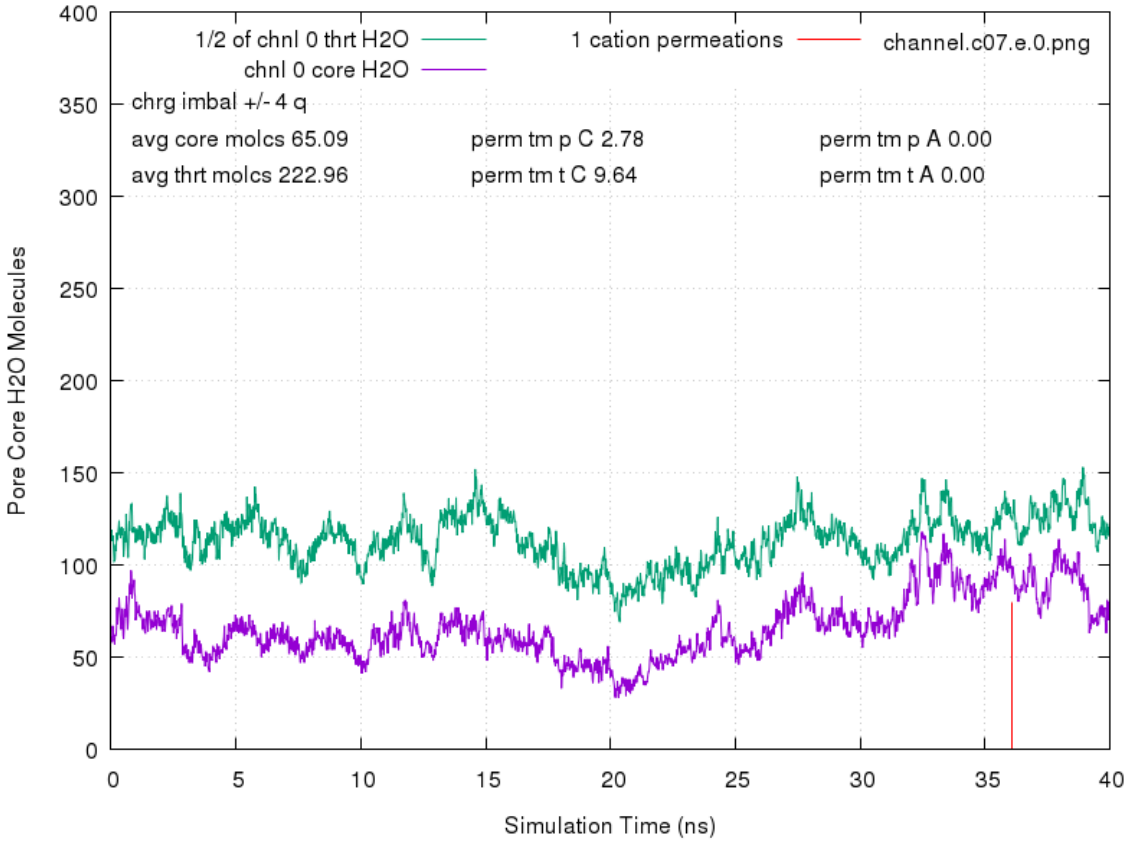

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c0A.B.1.png
chrg imbal +/- 4 q
avg core molcs 52.14
avg thrt molcs 190.62
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

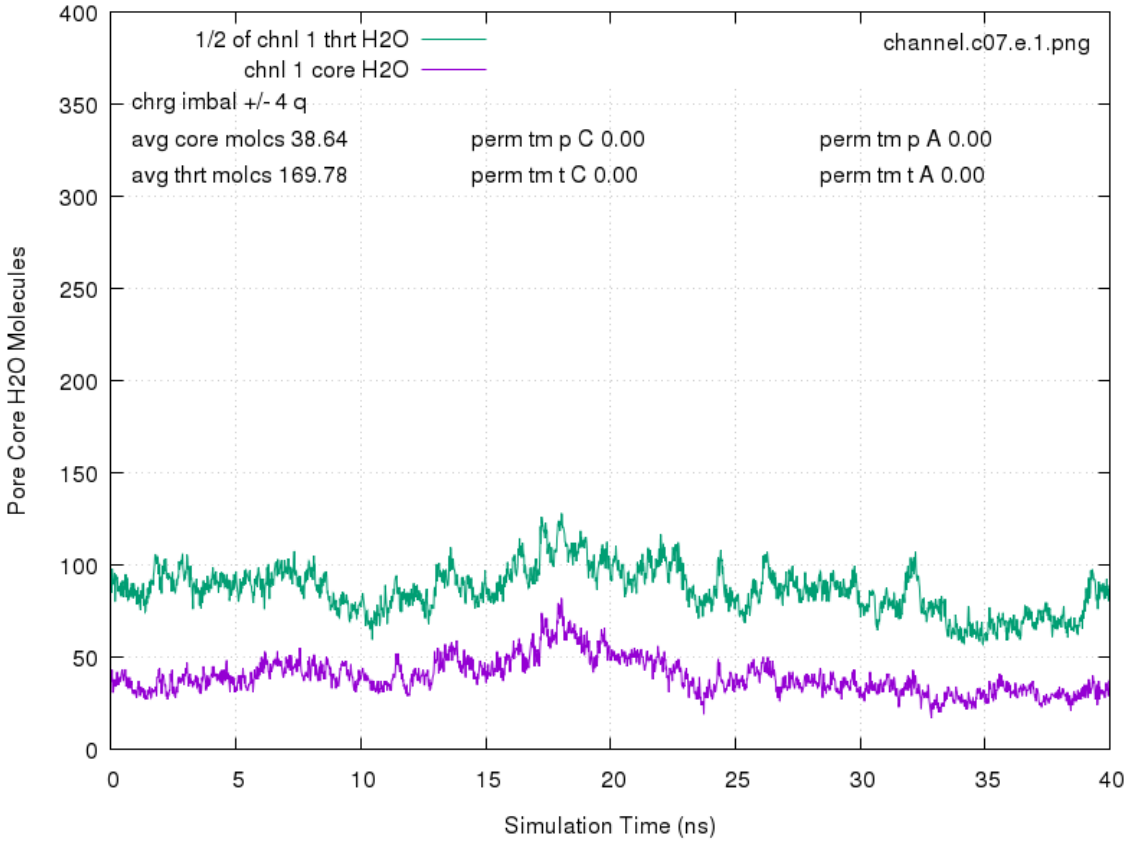

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
4 cation permeations
2 anion permeations
channel.c0A.C.0.png
chrg imbal +/- 4 q
avg core molcs 88.61
avg thrt molcs 266.74
perm tm p C 1.19
perm tm t C 3.67
perm tm p A 0.64
perm tm t A 1.77
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

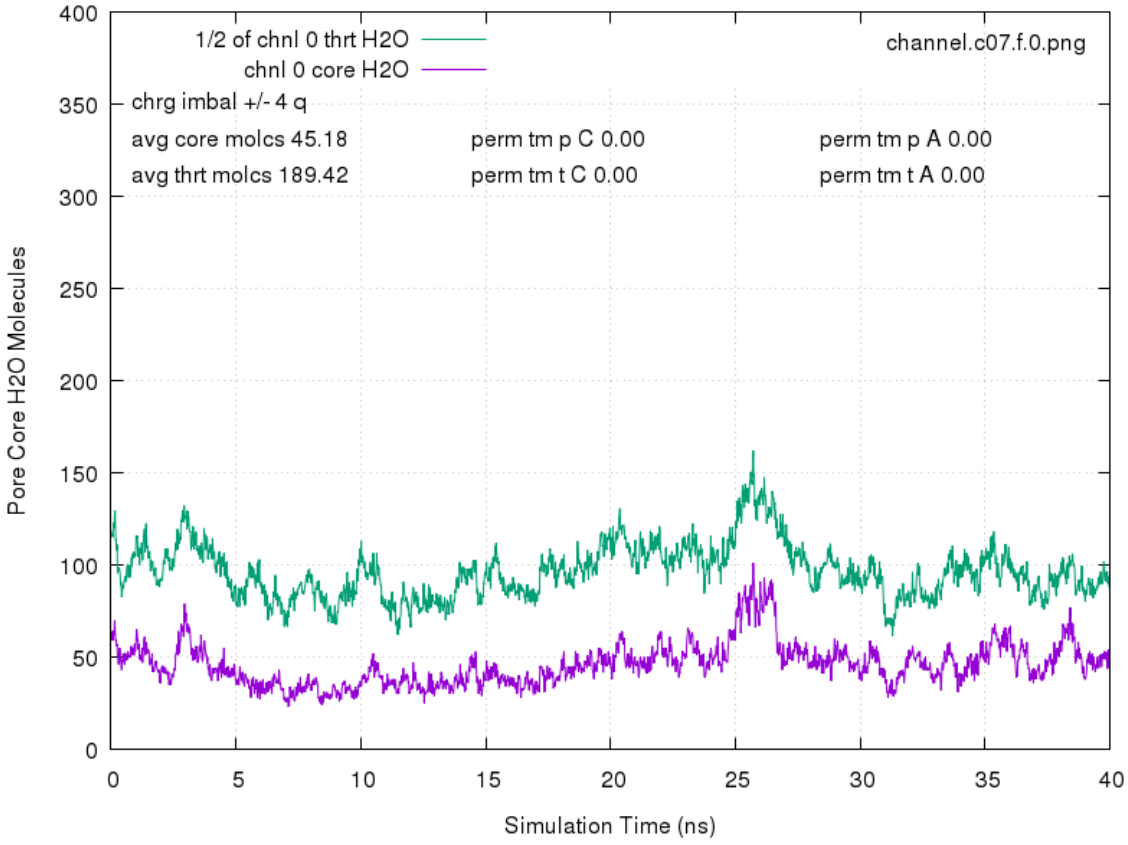

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c0A.C.1.png
chrg imbal +/- 4 q
avg core molcs 24.93
avg thrt molcs 108.03
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

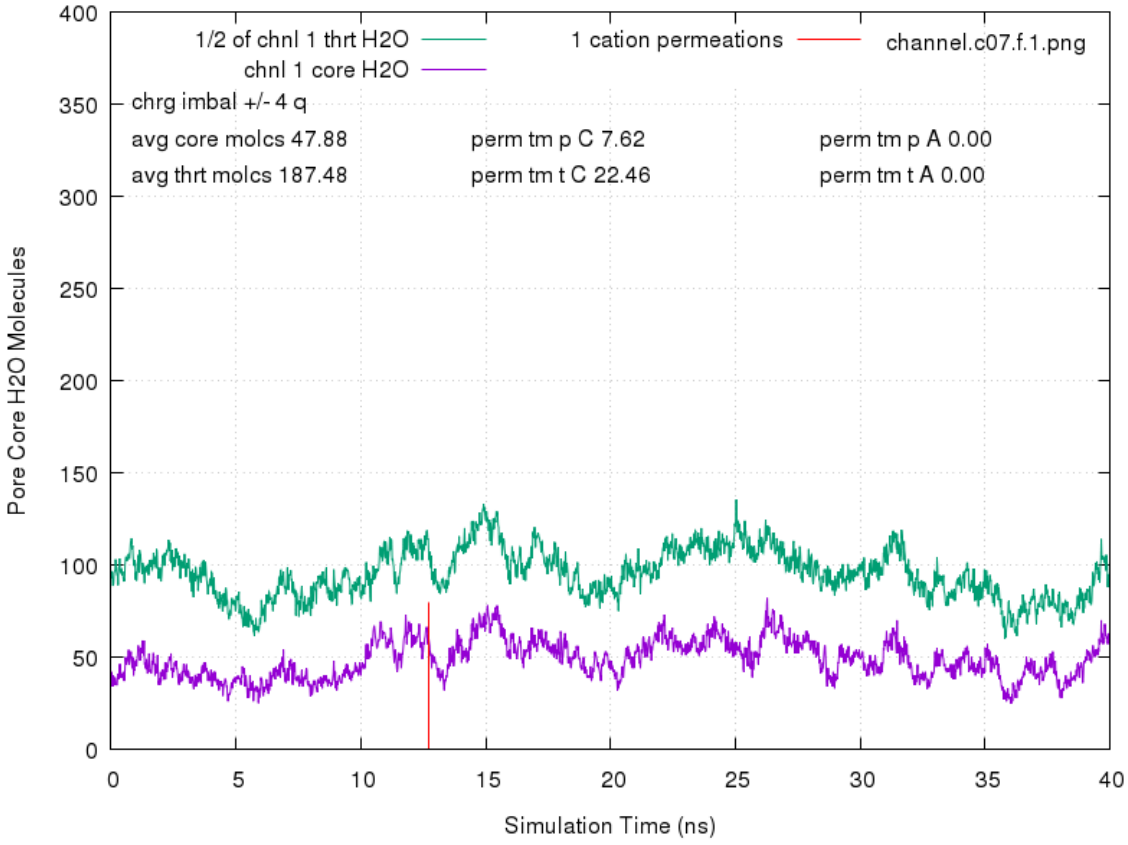

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c07.d.0.png
chrg imbal +/- 4 q
avg core molcs 44.26
avg thrt molcs 171.62
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

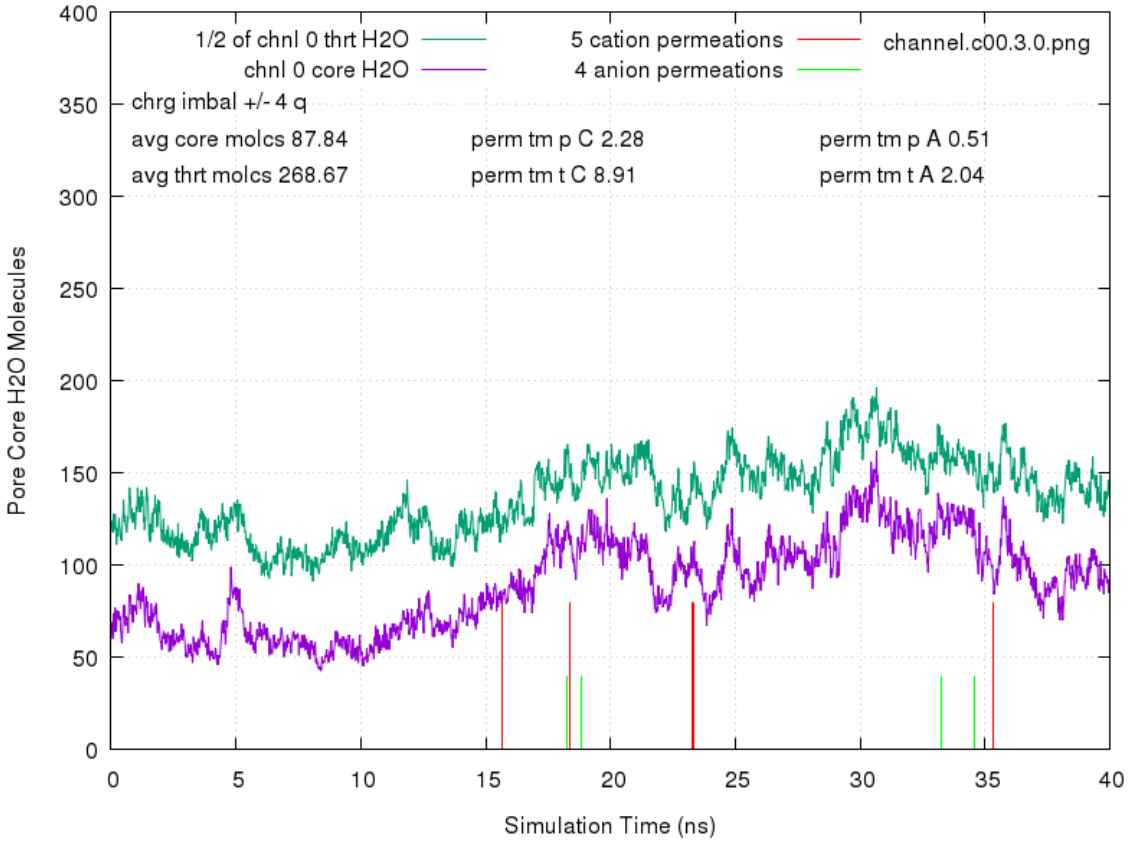

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c07.d.1.png
chrg imbal +/- 4 q
avg core molcs 50.48
avg thrt molcs 185.50
perm tm p C 34.06
perm tm t C 52.96
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

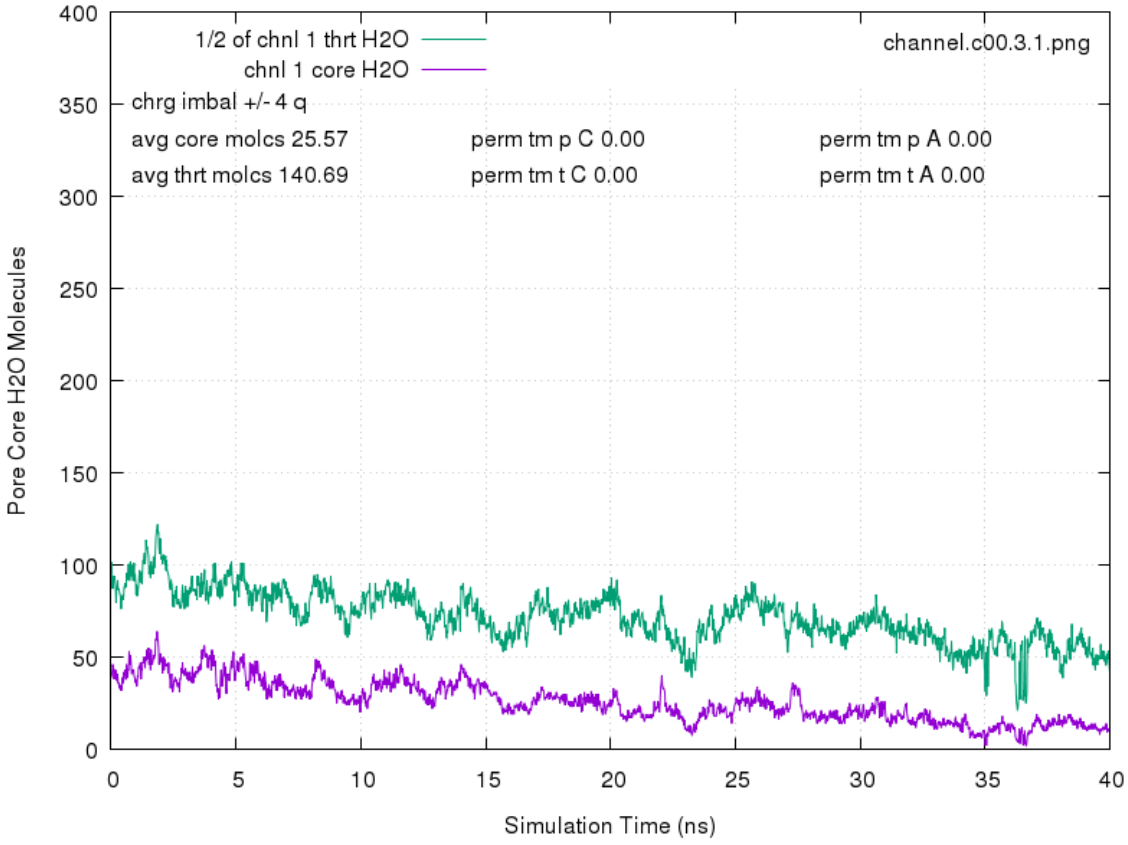

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c07.e.0.png
chrg imbal +/- 4 q
avg core molcs 65.09
avg thrt molcs 222.96
perm tm p C 2.78
perm tm t C 9.64
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

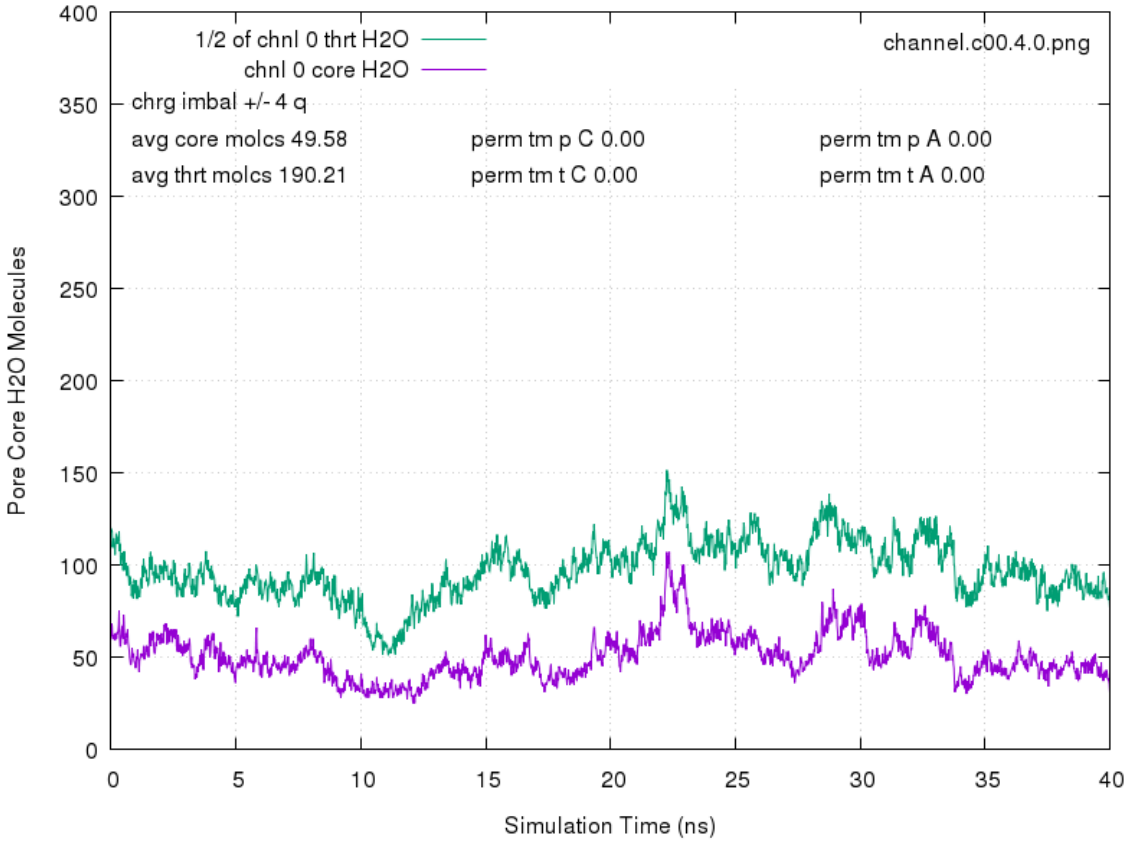

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c07.e.1.png
chrg imbal +/- 4 q
avg core molcs 38.64
avg thrt molcs 169.78
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

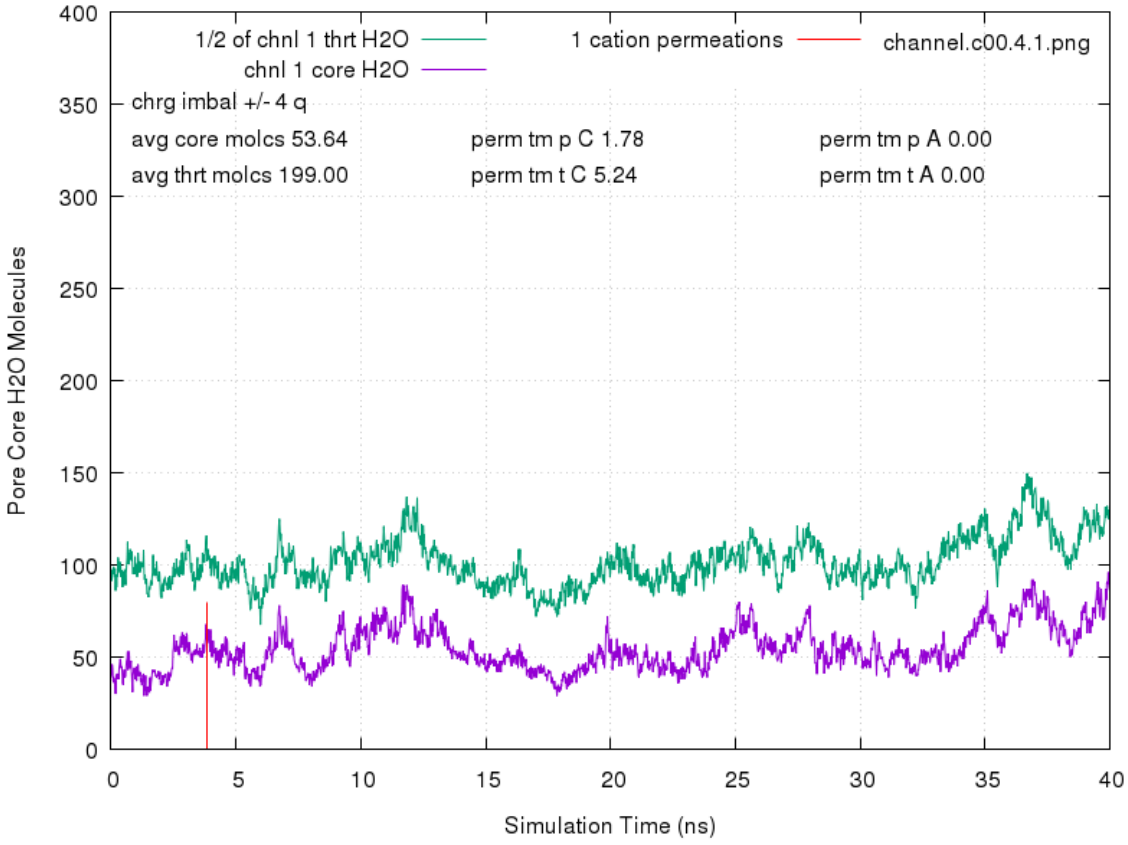

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c07.f.0.png
chrg imbal +/- 4 q
avg core molcs 45.18
avg thrt molcs 189.42
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

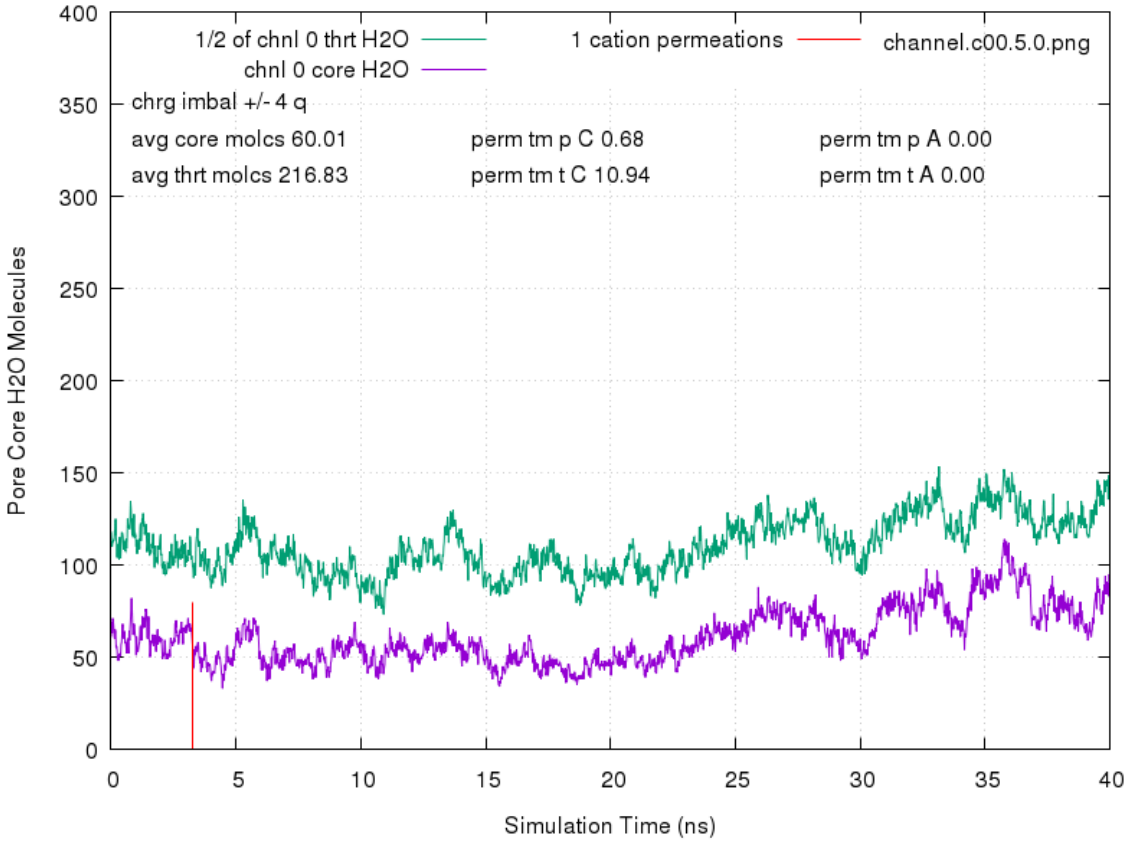

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c07.f.1.png
chrg imbal +/- 4 q
avg core molcs 47.88
avg thrt molcs 187.48
perm tm p C 7.62
perm tm t C 22.46
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

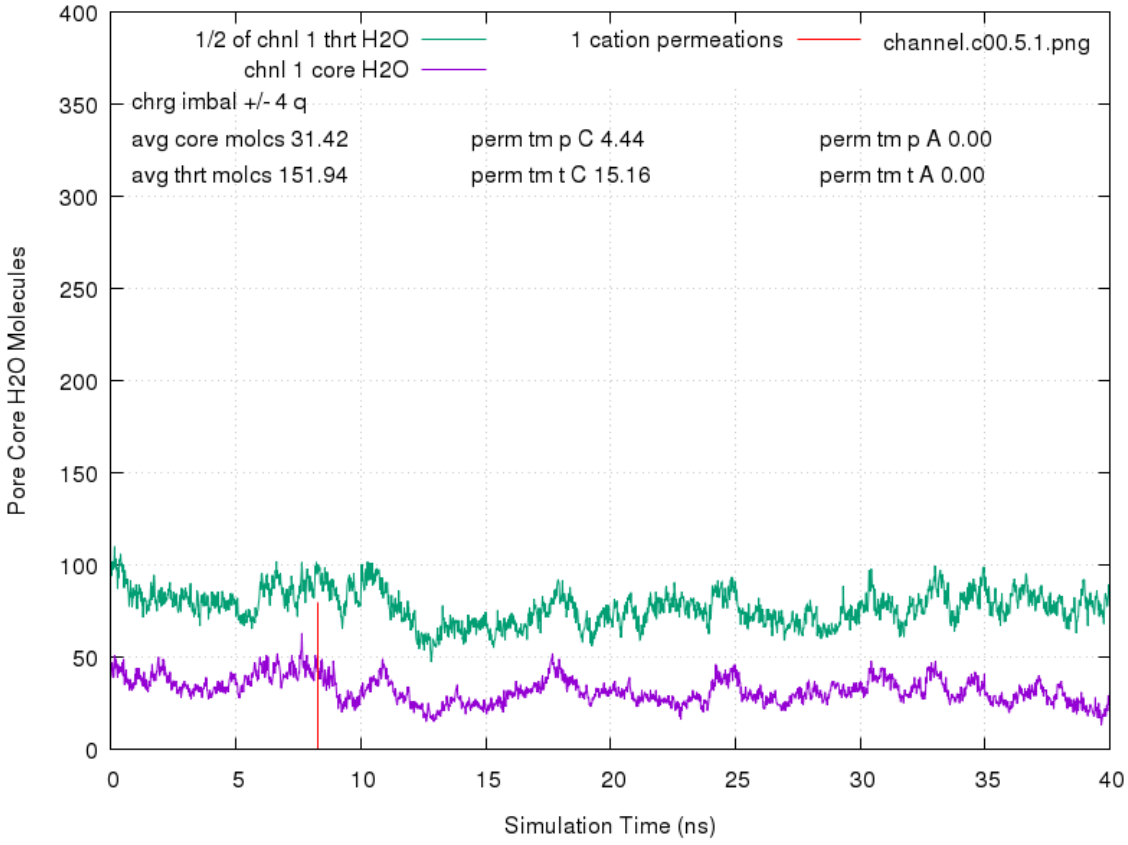

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
4 anion permeations
channel.c00.3.0.png
chrg imbal +/- 4 q
avg core molcs 87.84
avg thrt molcs 268.67
perm tm p C 2.28
perm tm t C 8.91
perm tm p A 0.51
perm tm t A 2.04
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

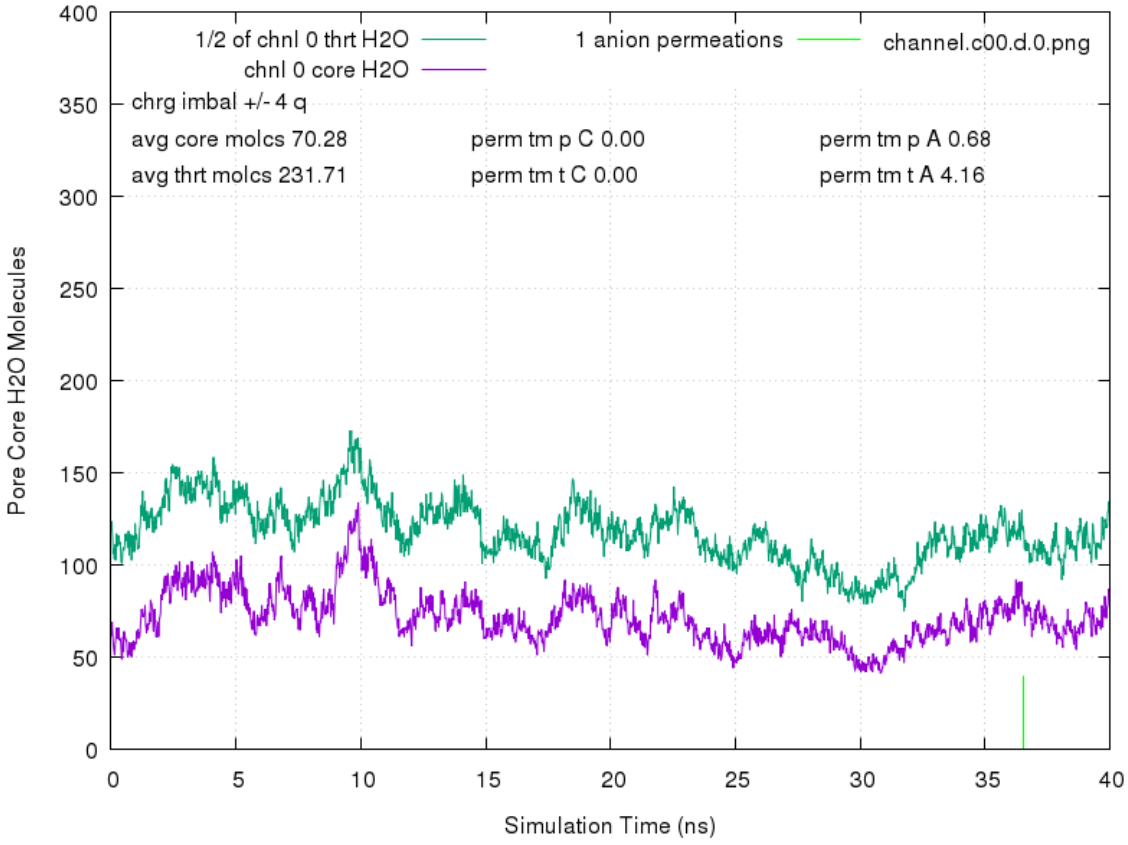

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c00.3.1.png
chrg imbal +/- 4 q
avg core molcs 25.57
avg thrt molcs 140.69
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

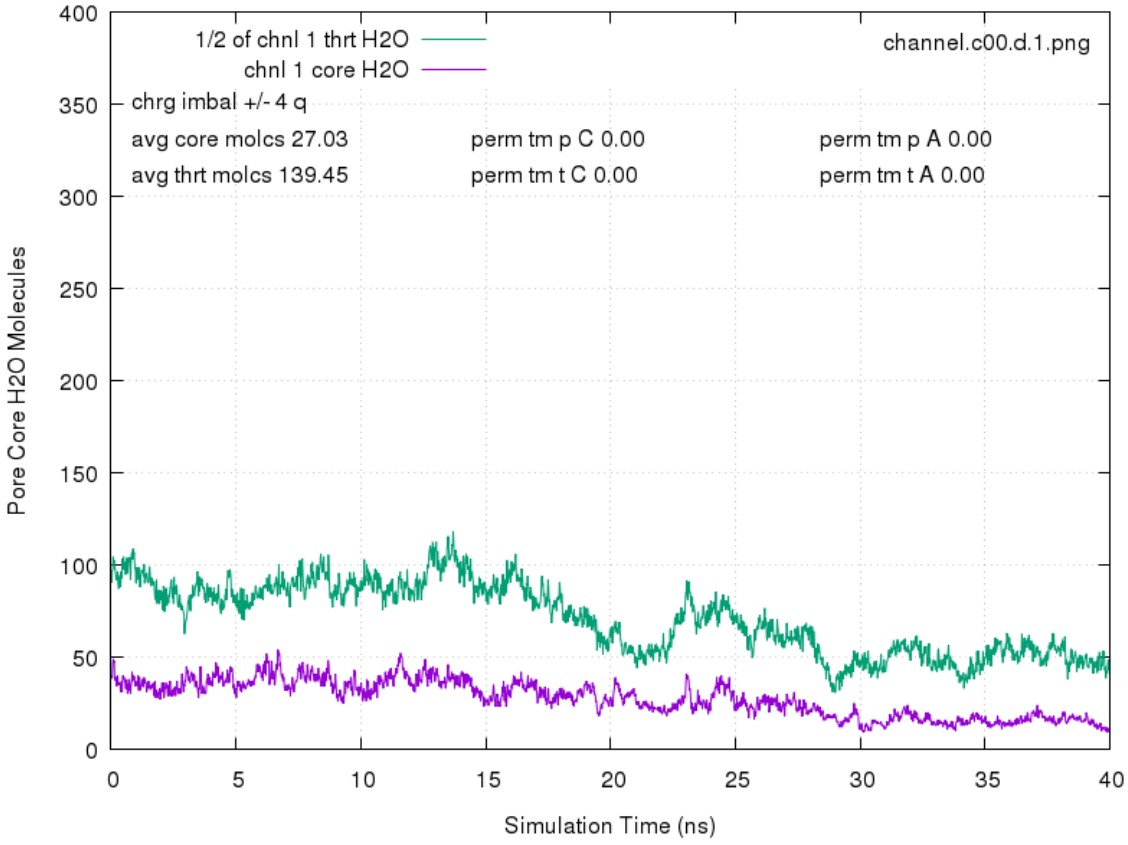

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c00.4.0.png
chrg imbal +/- 4 q
avg core molcs 49.58
avg thrt molcs 190.21
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

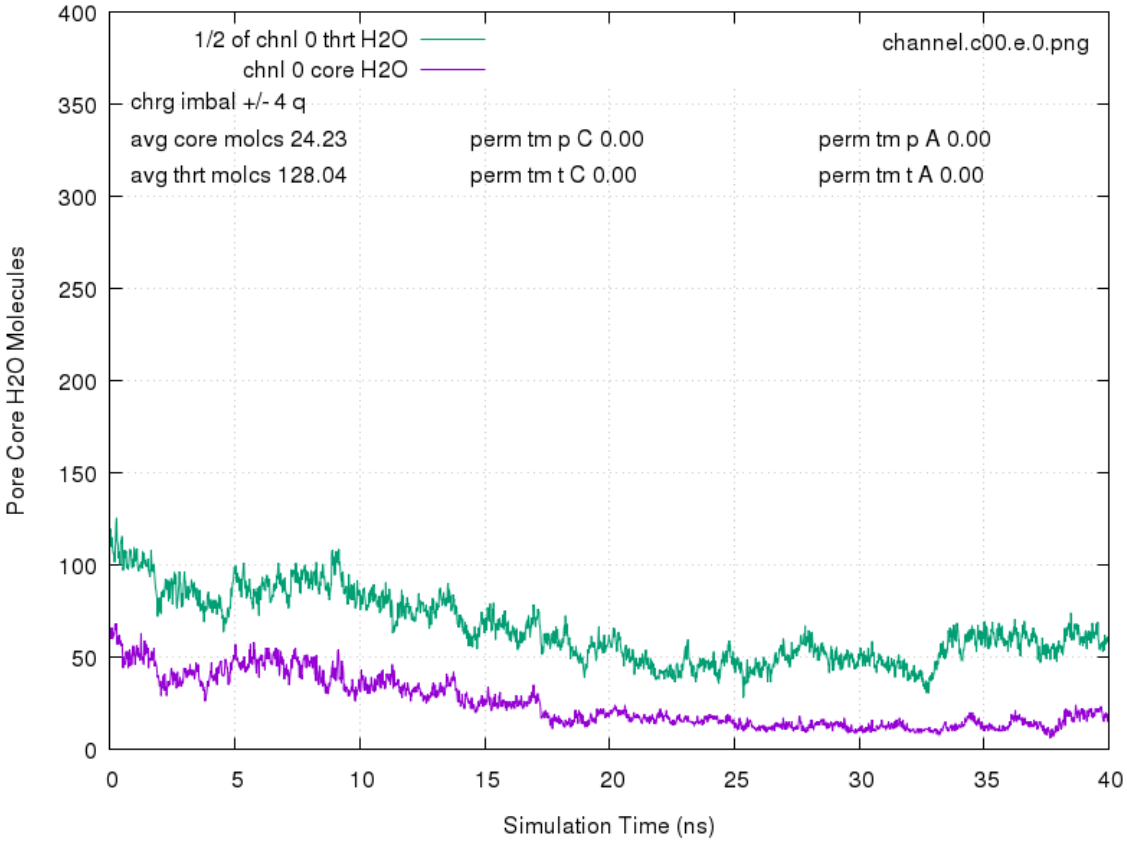

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c00.4.1.png
chrg imbal +/- 4 q
avg core molcs 53.64
avg thrt molcs 199.00
perm tm p C 1.78
perm tm t C 5.24
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

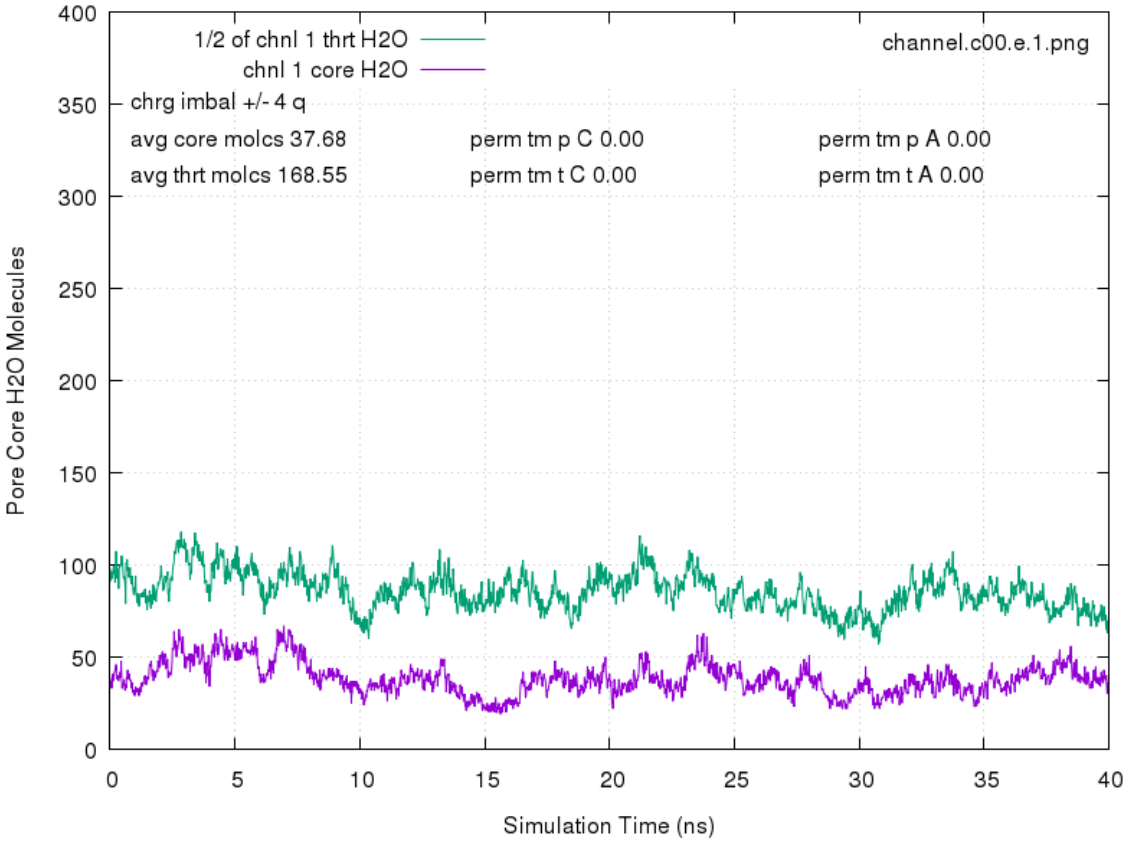

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c00.5.0.png
chrg imbal +/- 4 q
avg core molcs 60.01
avg thrt molcs 216.83
perm tm p C 0.68
perm tm t C 10.94
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

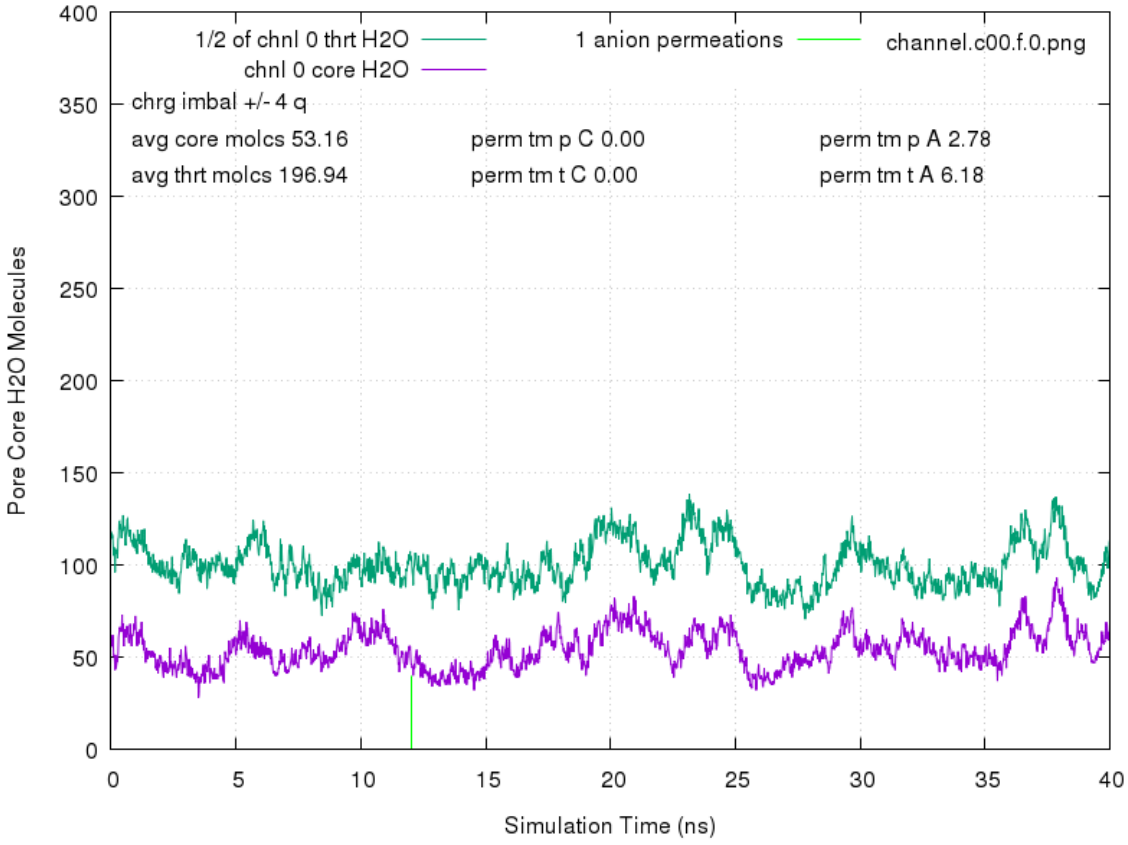

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c00.5.1.png
chrg imbal +/- 4 q
avg core molcs 31.42
avg thrt molcs 151.94
perm tm p C 4.44
perm tm t C 15.16
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

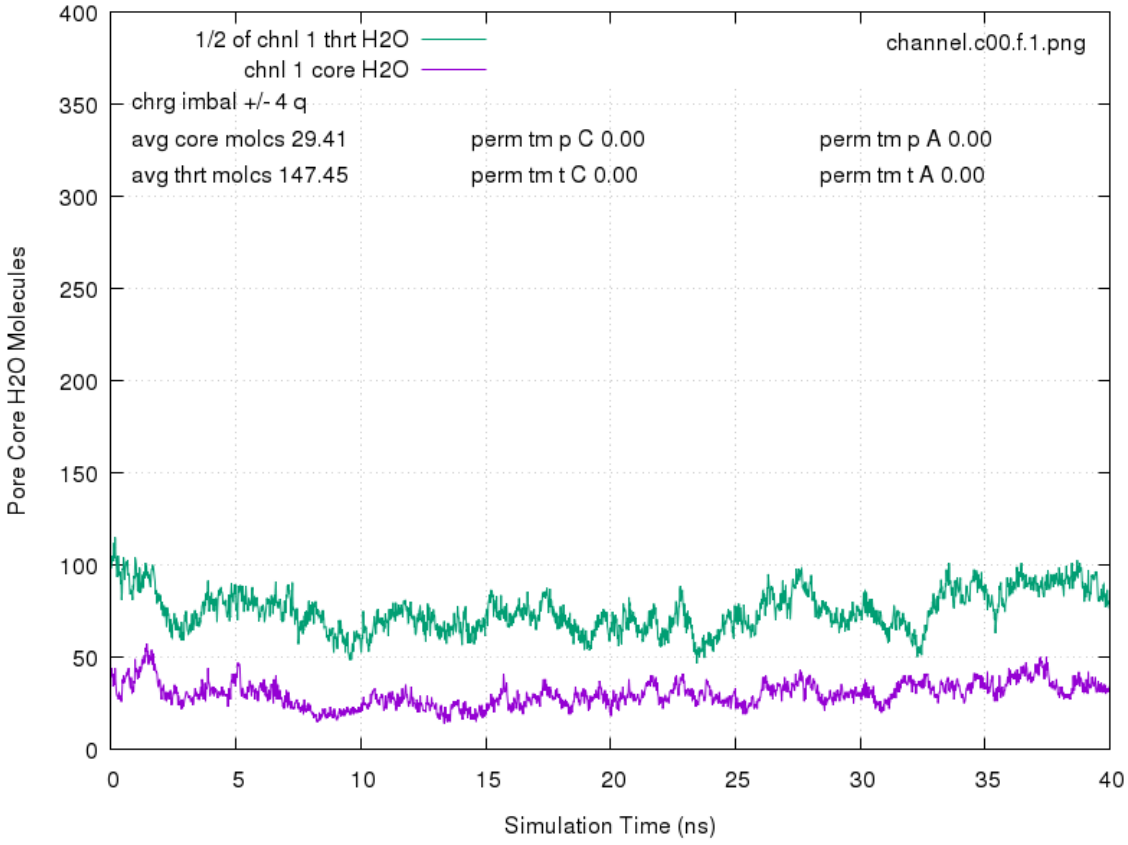

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 anion permeations
channel.c00.d.0.png
chrg imbal +/- 4 q
avg core molcs 70.28
avg thrt molcs 231.71
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.68
perm tm t A 4.16
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

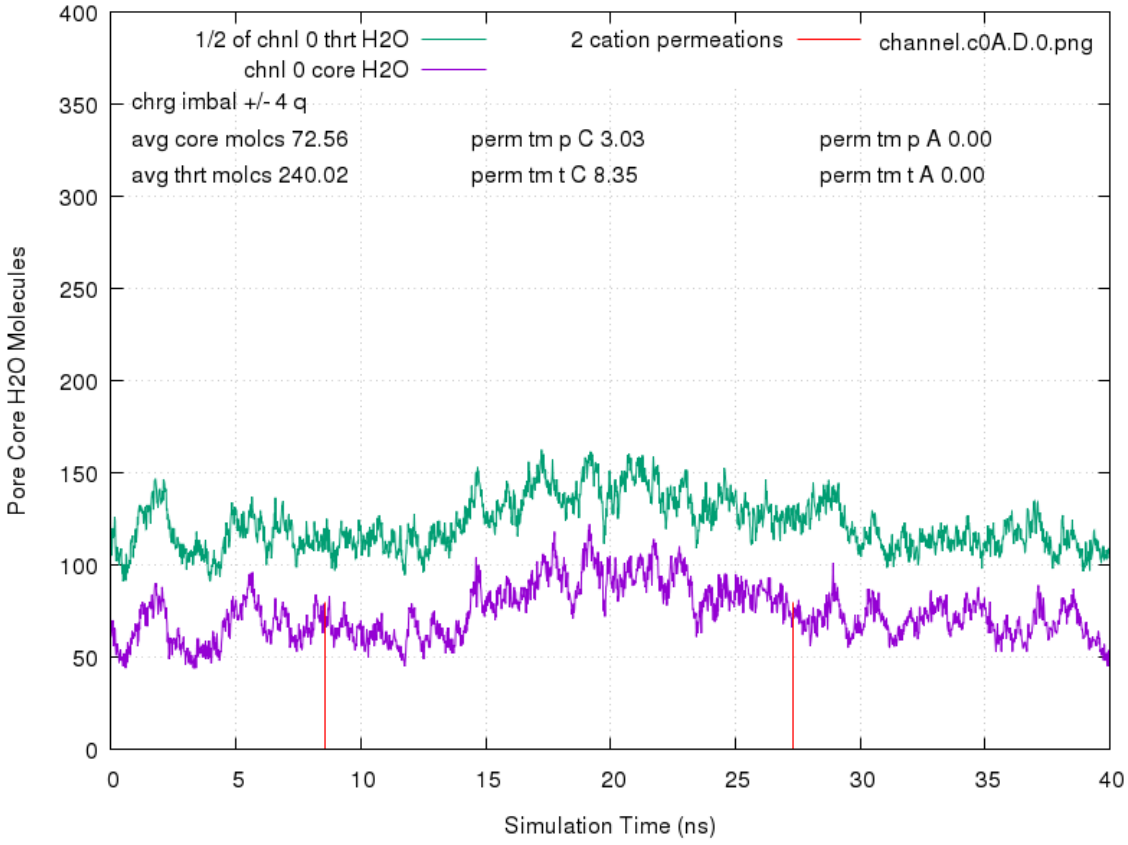

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c00.d.1.png
chrg imbal +/- 4 q
avg core molcs 27.03
avg thrt molcs 139.45
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

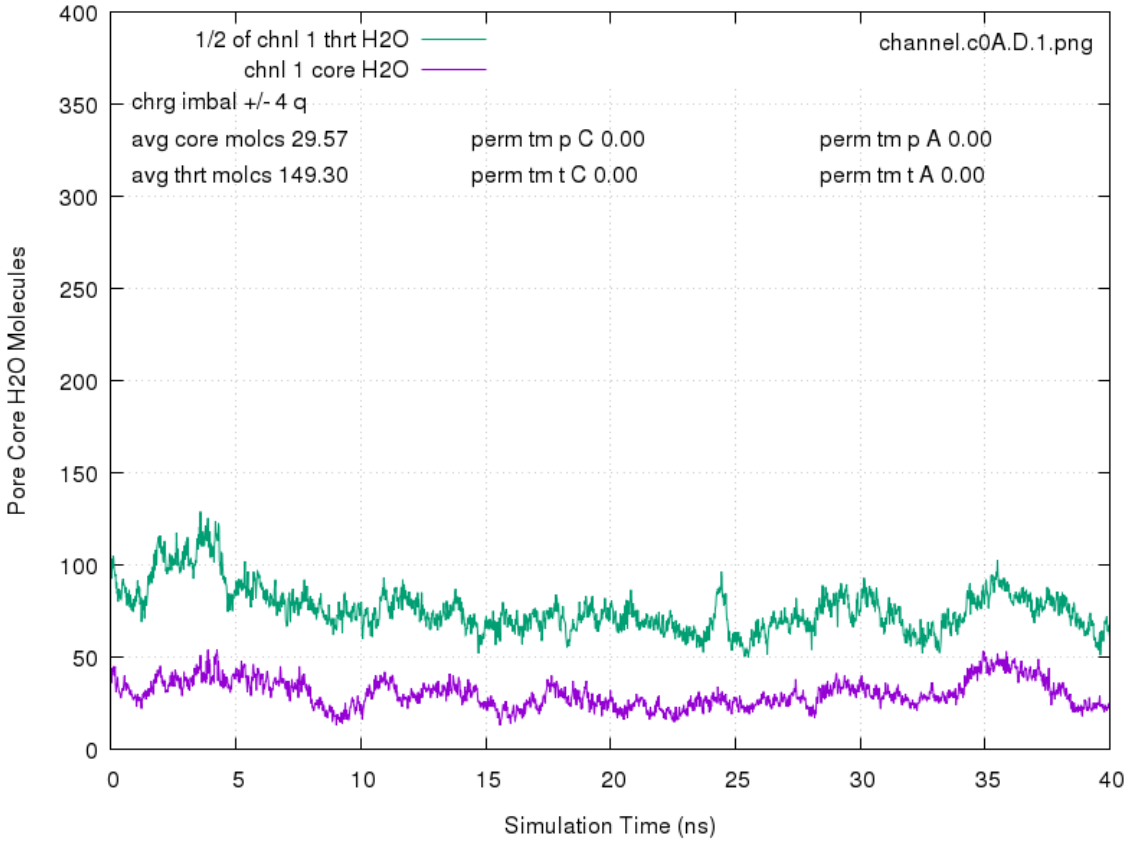

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c00.e.0.png
chrg imbal +/- 4 q
avg core molcs 24.23
avg thrt molcs 128.04
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

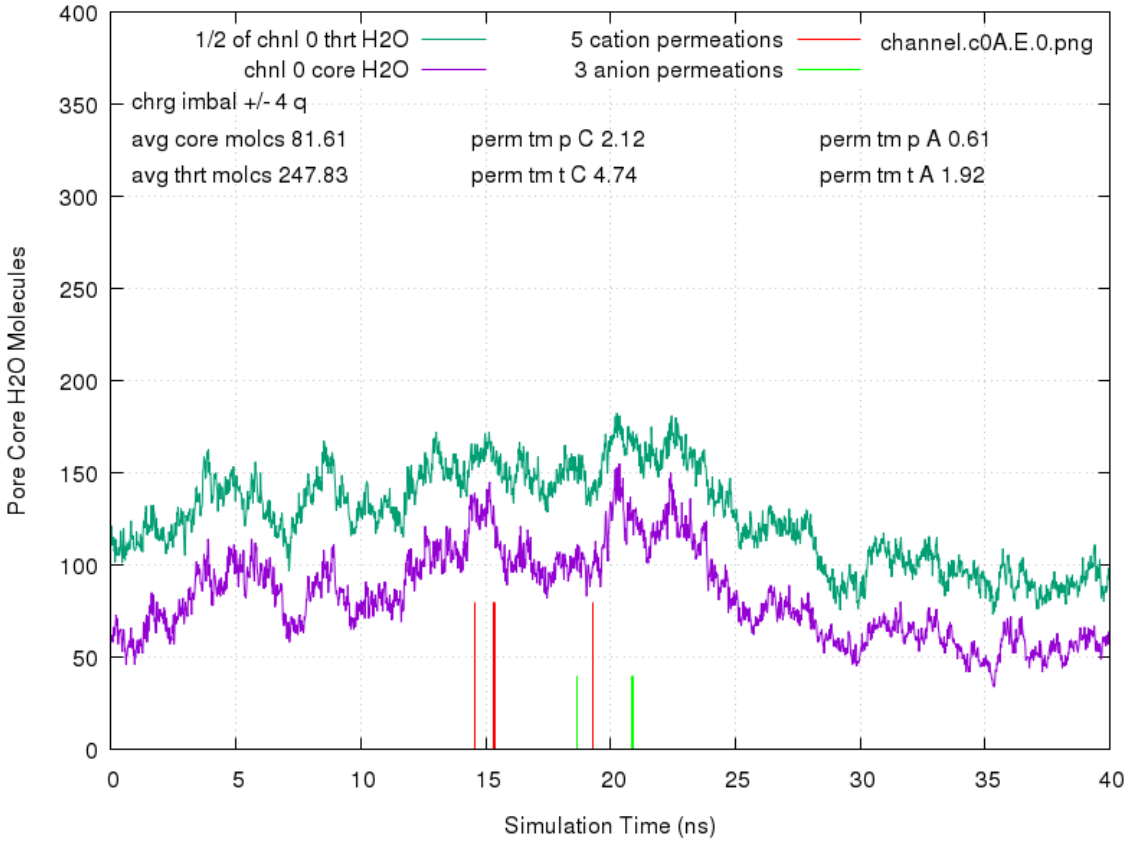

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c00.e.1.png
chrg imbal +/- 4 q
avg core molcs 37.68
avg thrt molcs 168.55
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

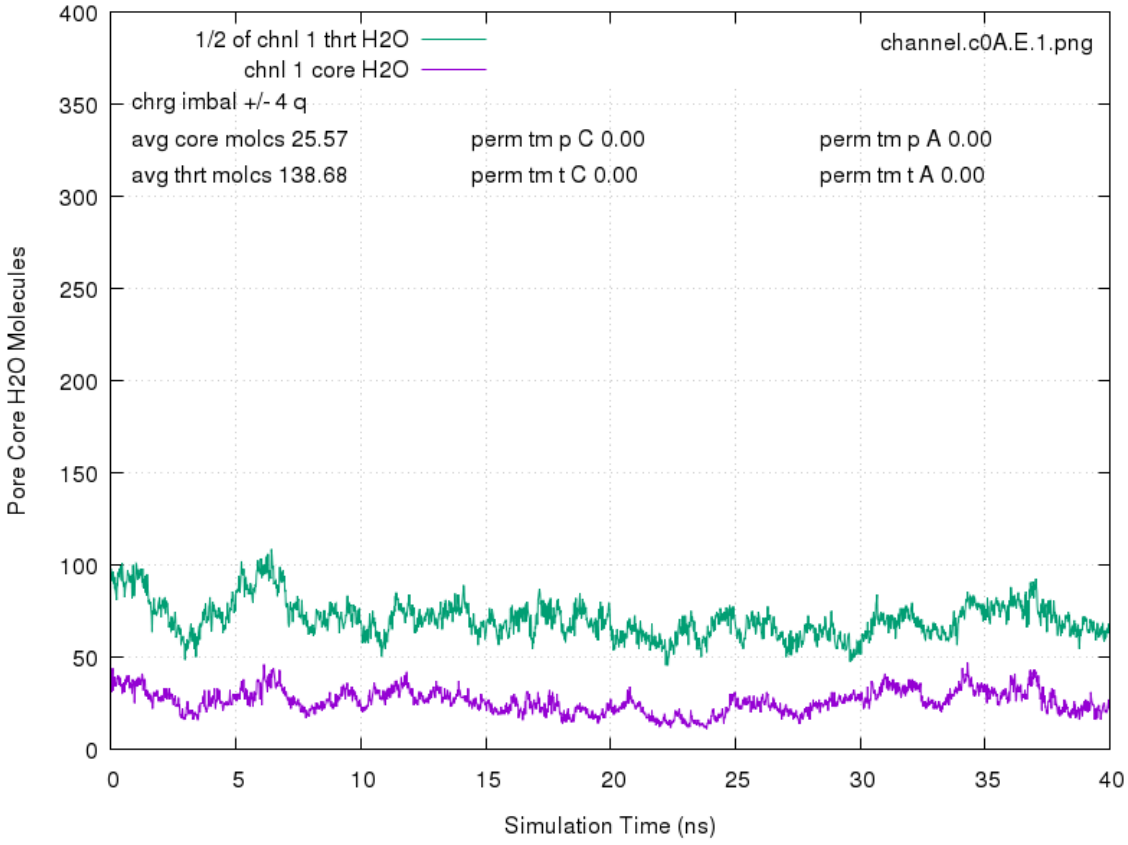

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 anion permeations
channel.c00.f.0.png
chrg imbal +/- 4 q
avg core molcs 53.16
avg thrt molcs 196.94
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 2.78
perm tm t A 6.18
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

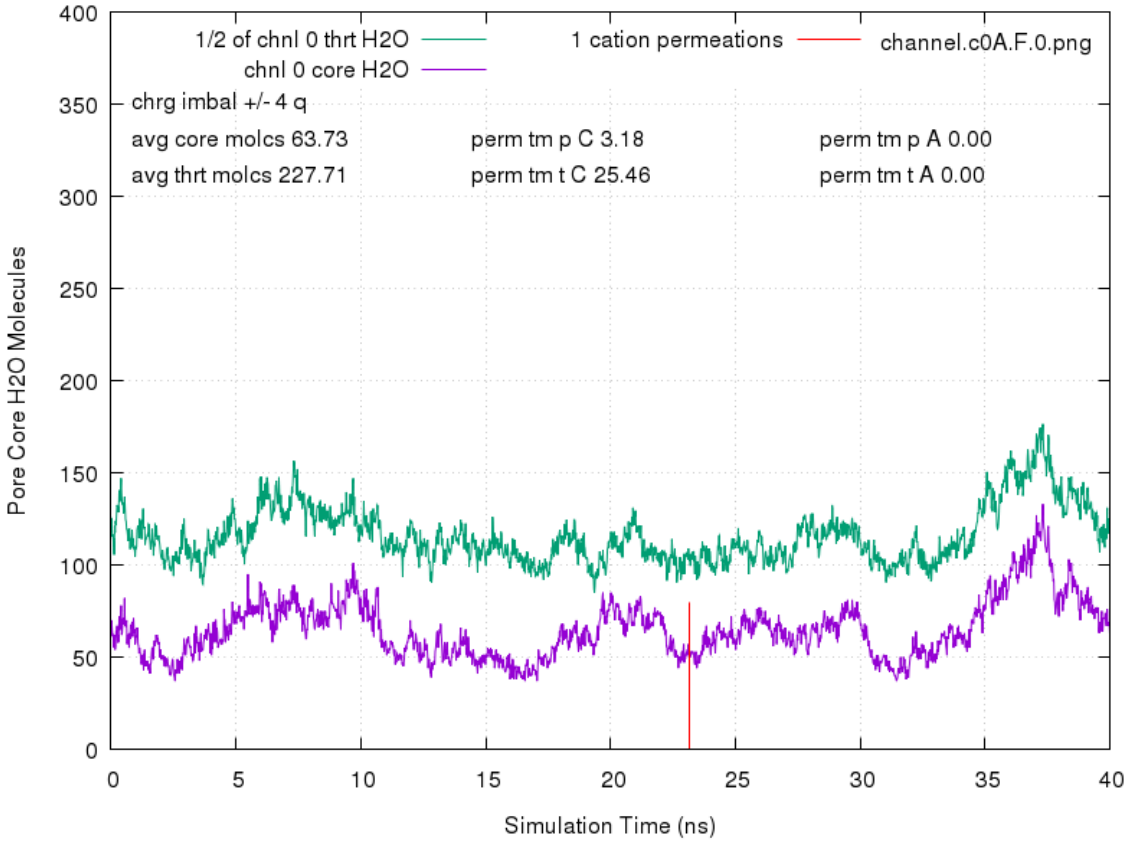

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c00.f.1.png
chrg imbal +/- 4 q
avg core molcs 29.41
avg thrt molcs 147.45
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

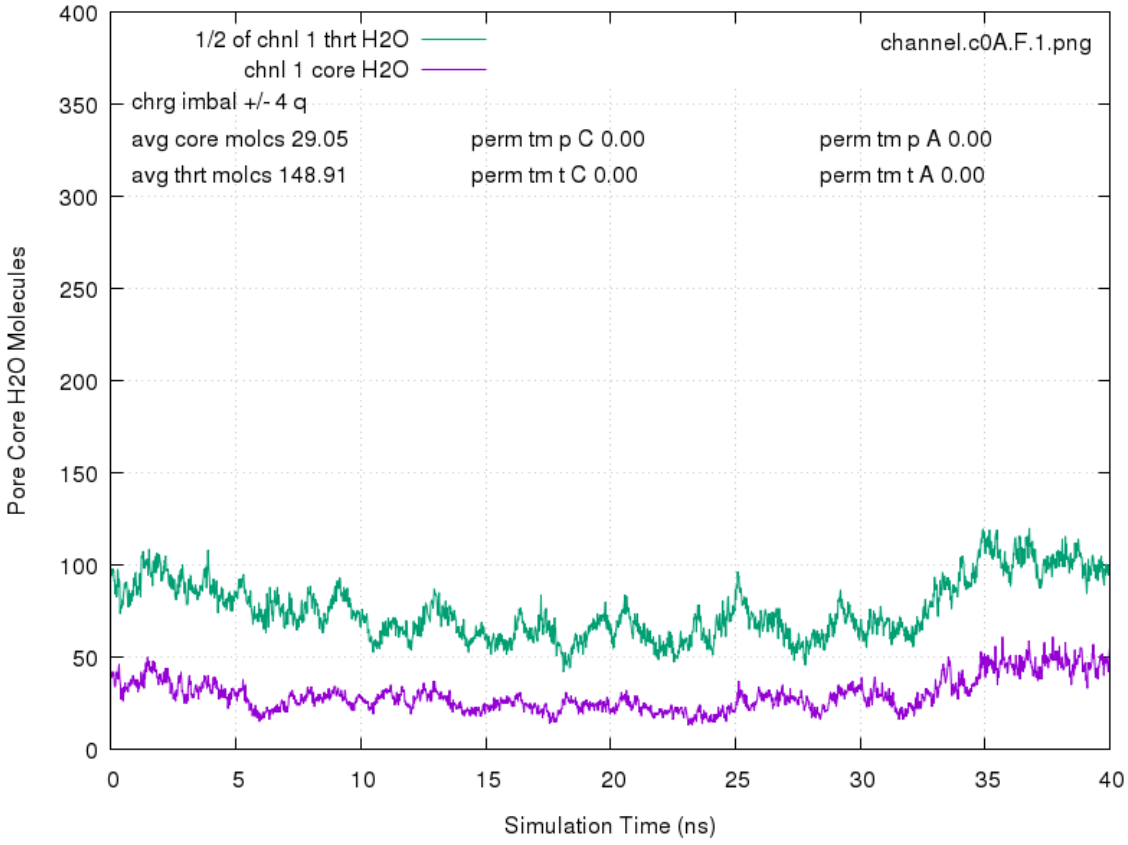

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
channel.c0A.D.0.png
chrg imbal +/- 4 q
avg core molcs 72.56
avg thrt molcs 240.02
perm tm p C 3.03
perm tm t C 8.35
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

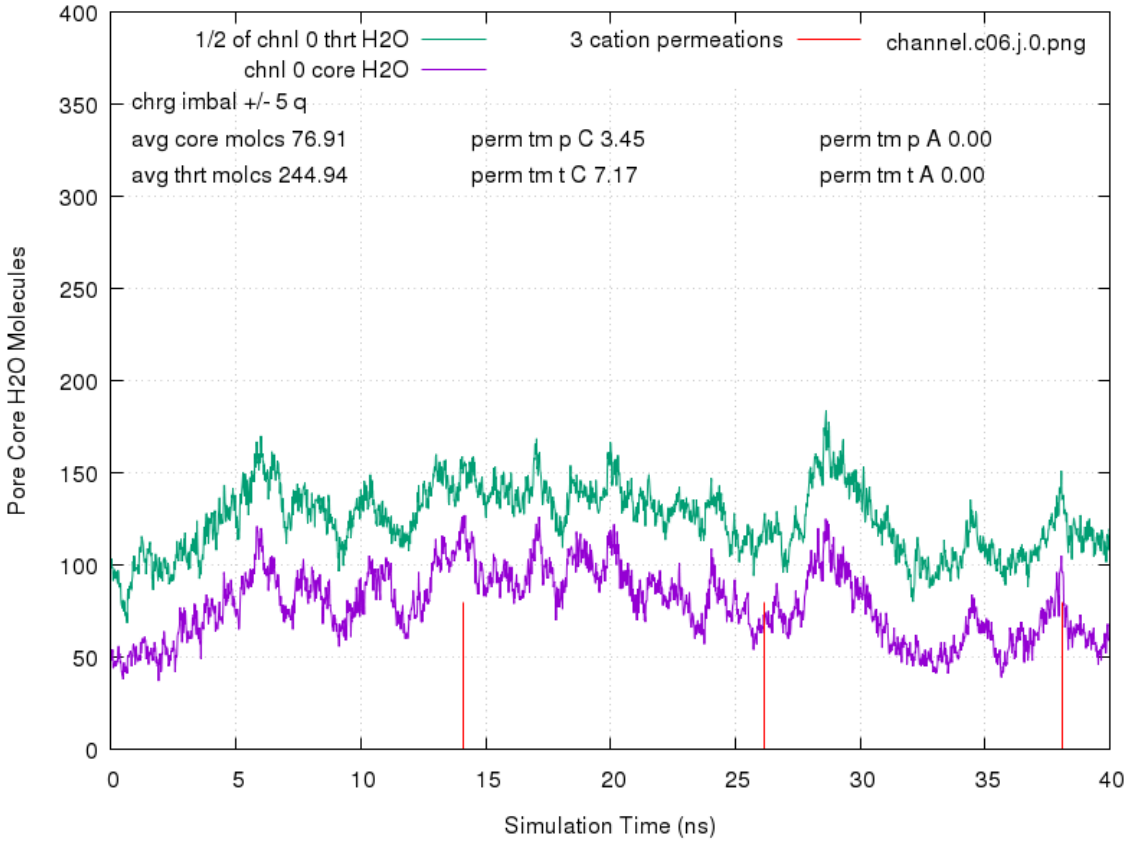

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c0A.D.1.png
chrg imbal +/- 4 q
avg core molcs 29.57
avg thrt molcs 149.30
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

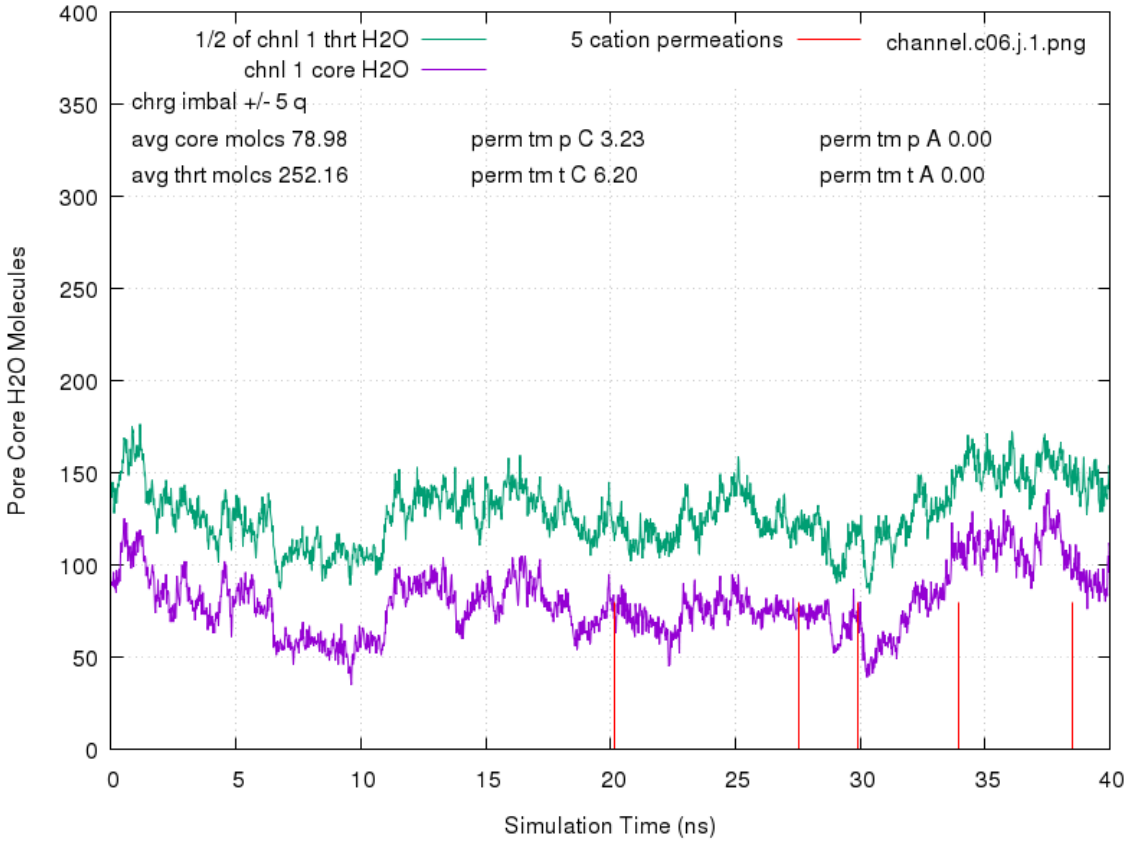

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
3 anion permeations
channel.c0A.E.0.png
chrg imbal +/- 4 q
avg core molcs 81.61
avg thrt molcs 247.83
perm tm p C 2.12
perm tm t C 4.74
perm tm p A 0.61
perm tm t A 1.92
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

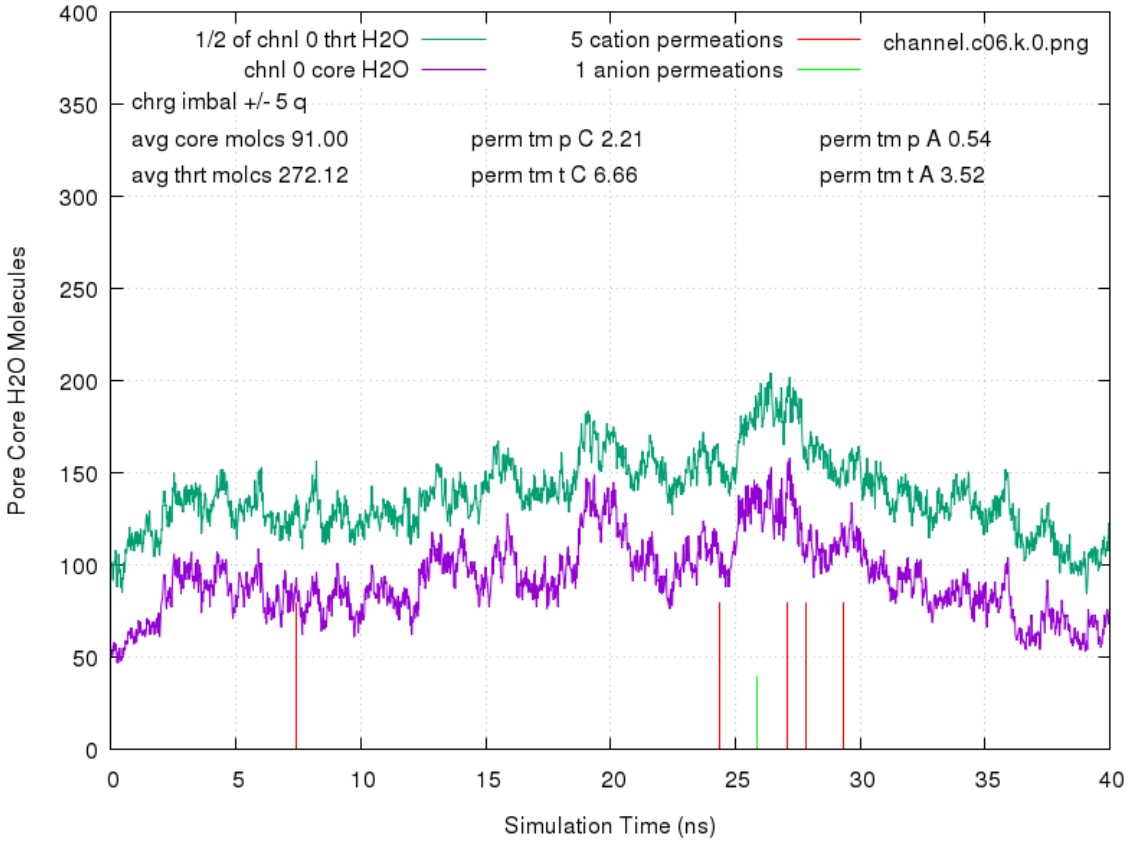

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c0A.E.1.png
chrg imbal +/- 4 q
avg core molcs 25.57
avg thrt molcs 138.68
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

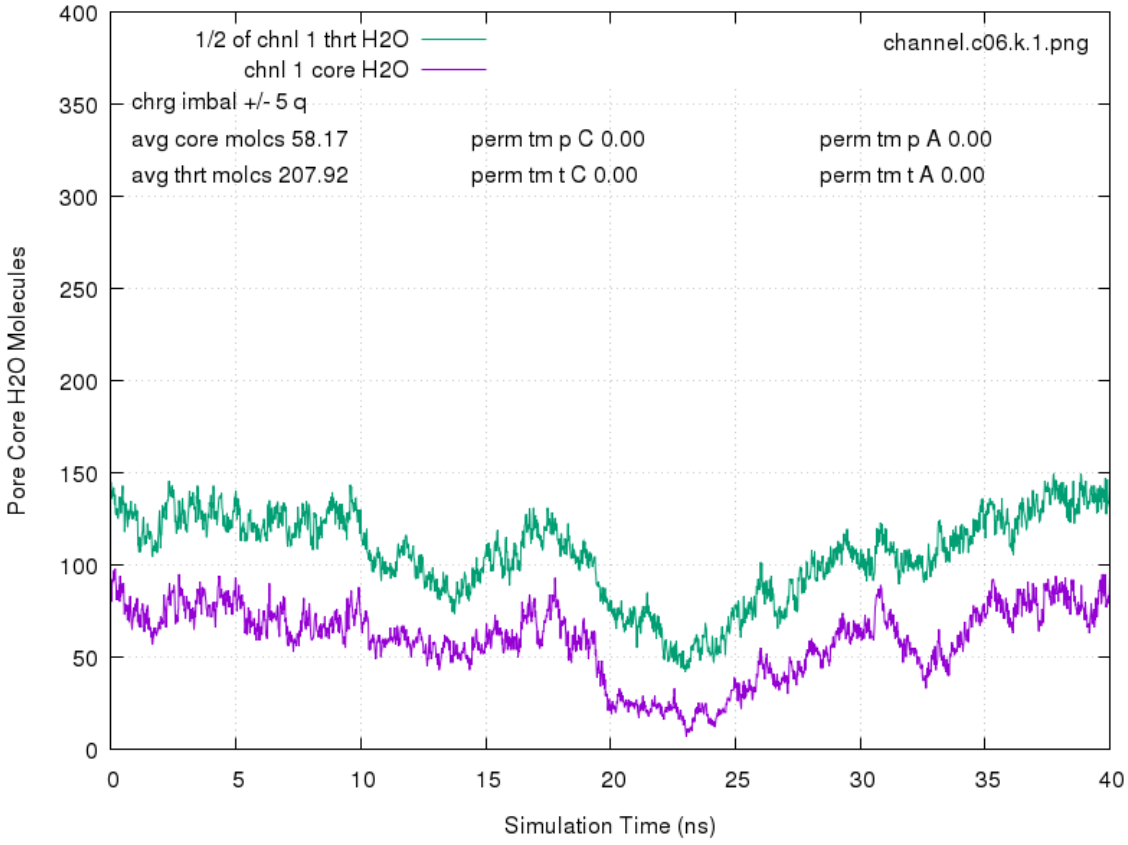

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c0A.F.0.png
chrg imbal +/- 4 q
avg core molcs 63.73
avg thrt molcs 227.71
perm tm p C 3.18
perm tm t C 25.46
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

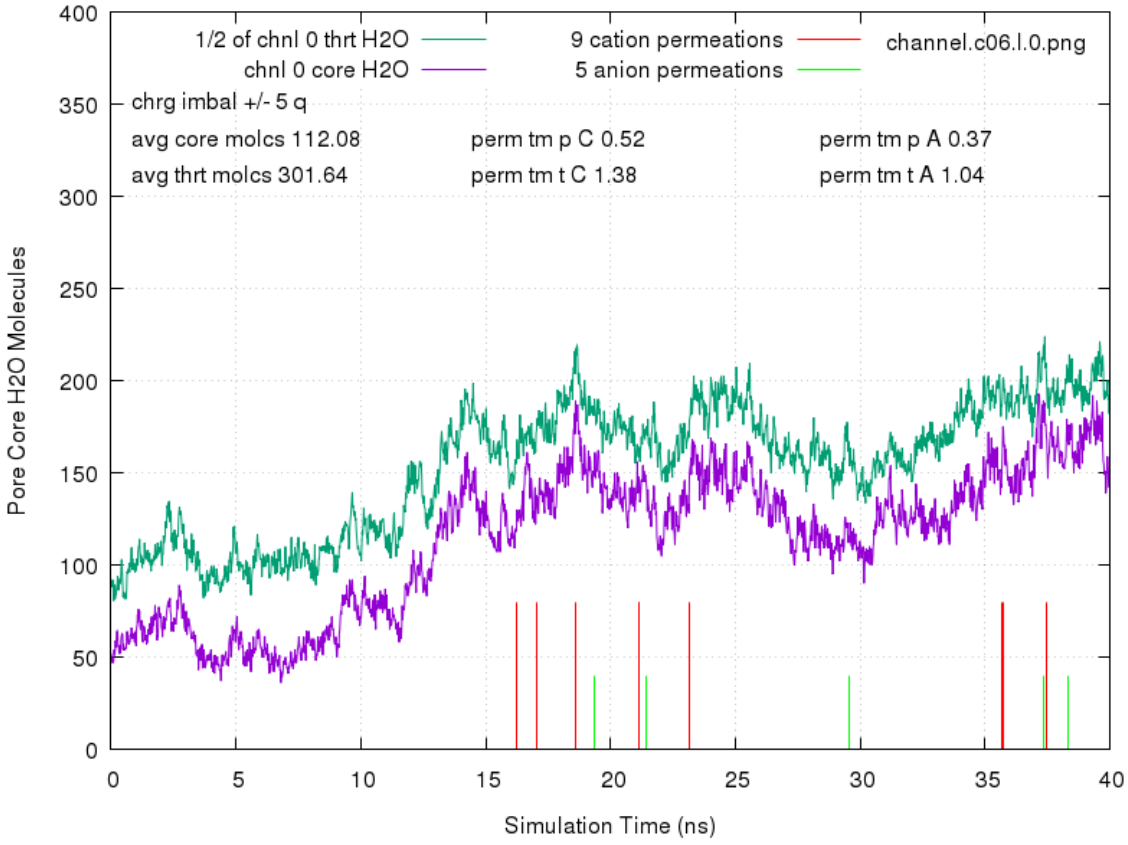

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c0A.F.1.png
chrg imbal +/- 4 q
avg core molcs 29.05
avg thrt molcs 148.91
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

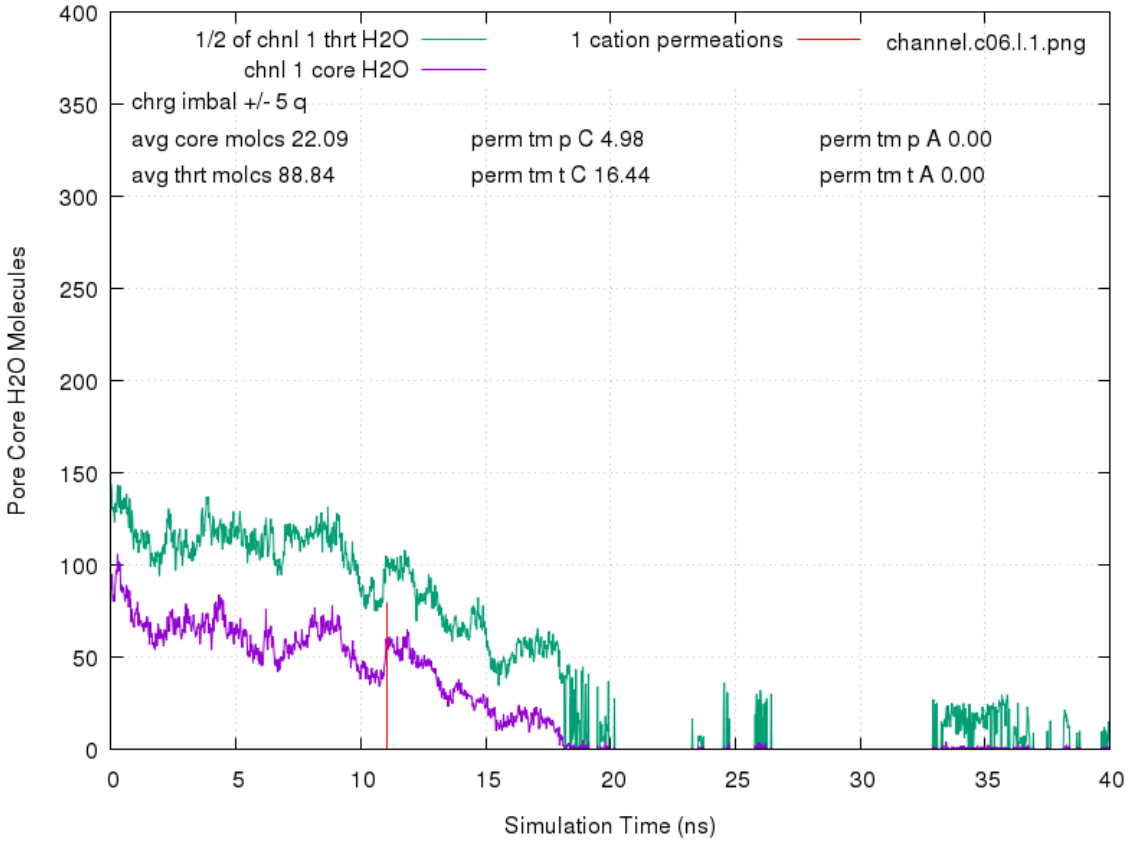

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
channel.c06.j.0.png
chrg imbal +/- 5 q
avg core molcs 76.91
avg thrt molcs 244.94
perm tm p C 3.45
perm tm t C 7.17
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

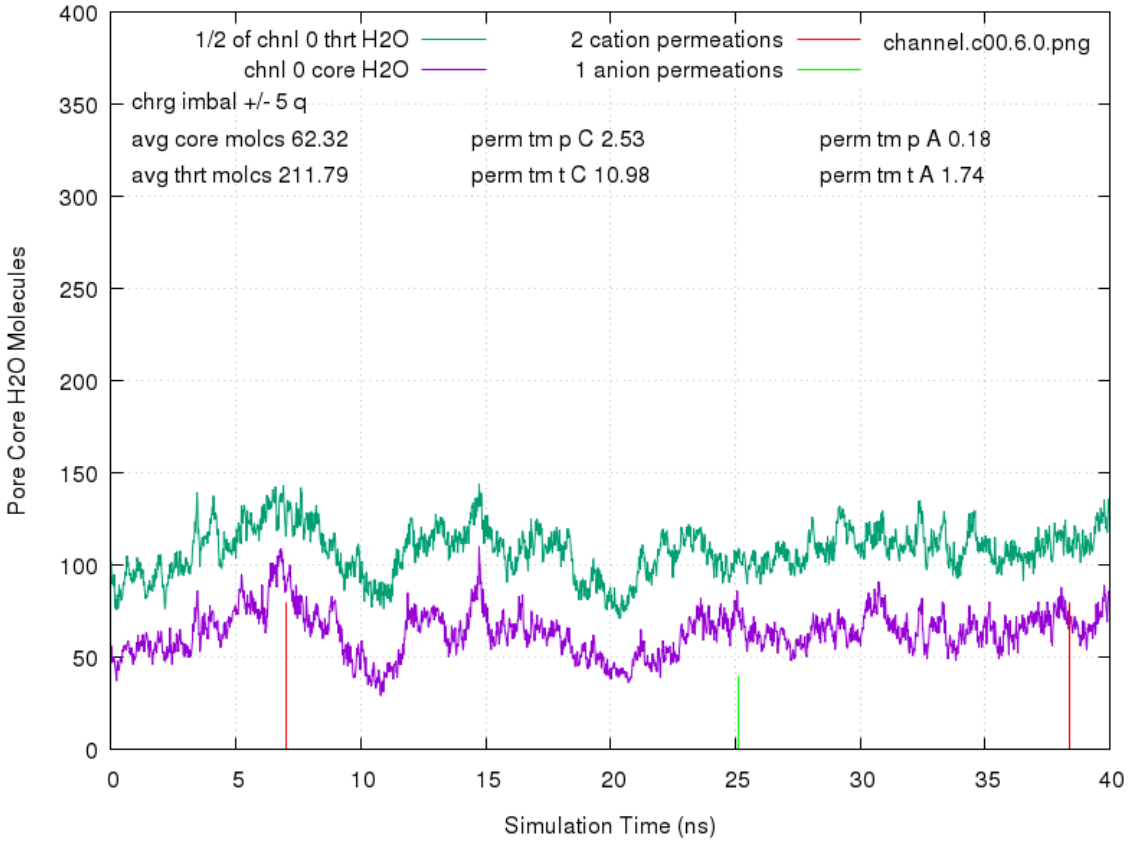

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
5 cation permeations
channel.c06.j.1.png
chrg imbal +/- 5 q
avg core molcs 78.98
avg thrt molcs 252.16
perm tm p C 3.23
perm tm t C 6.20
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

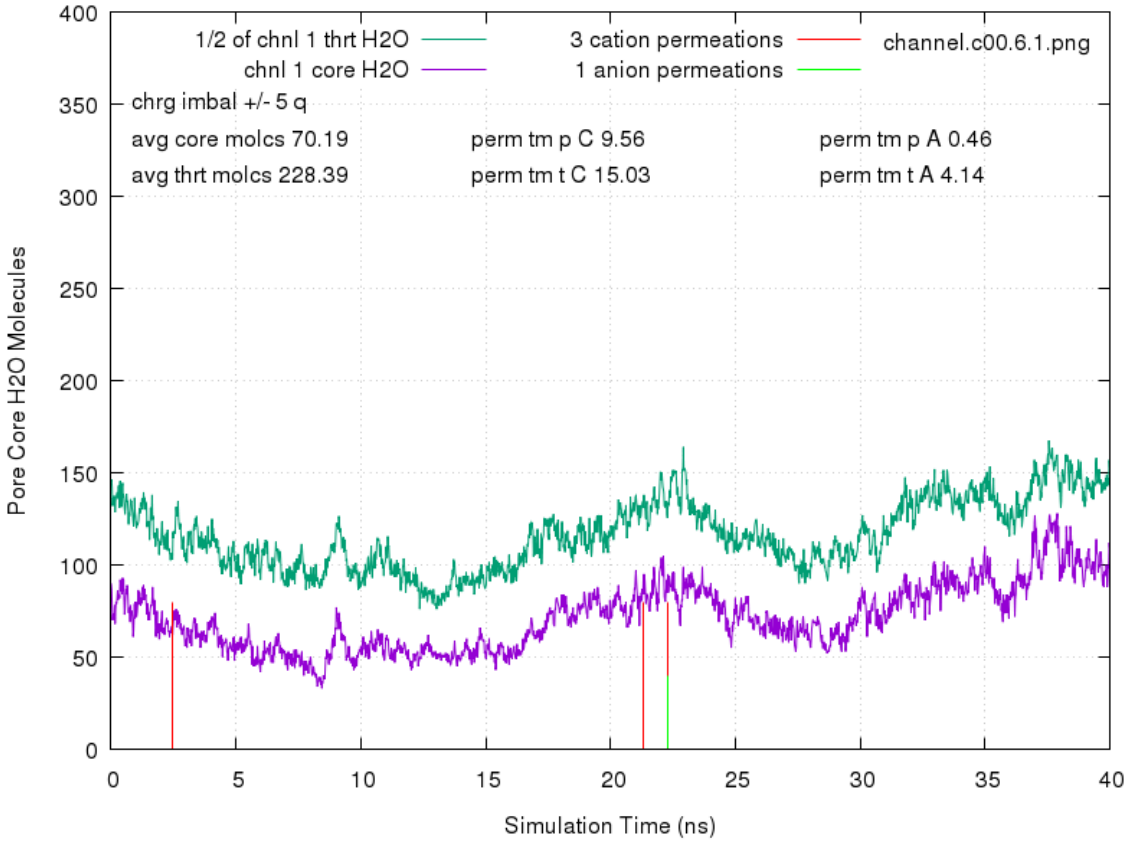

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
1 anion permeations
channel.c06.k.0.png
chrg imbal +/- 5 q
avg core molcs 91.00
avg thrt molcs 272.12
perm tm p C 2.21
perm tm t C 6.66
perm tm p A 0.54
perm tm t A 3.52
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

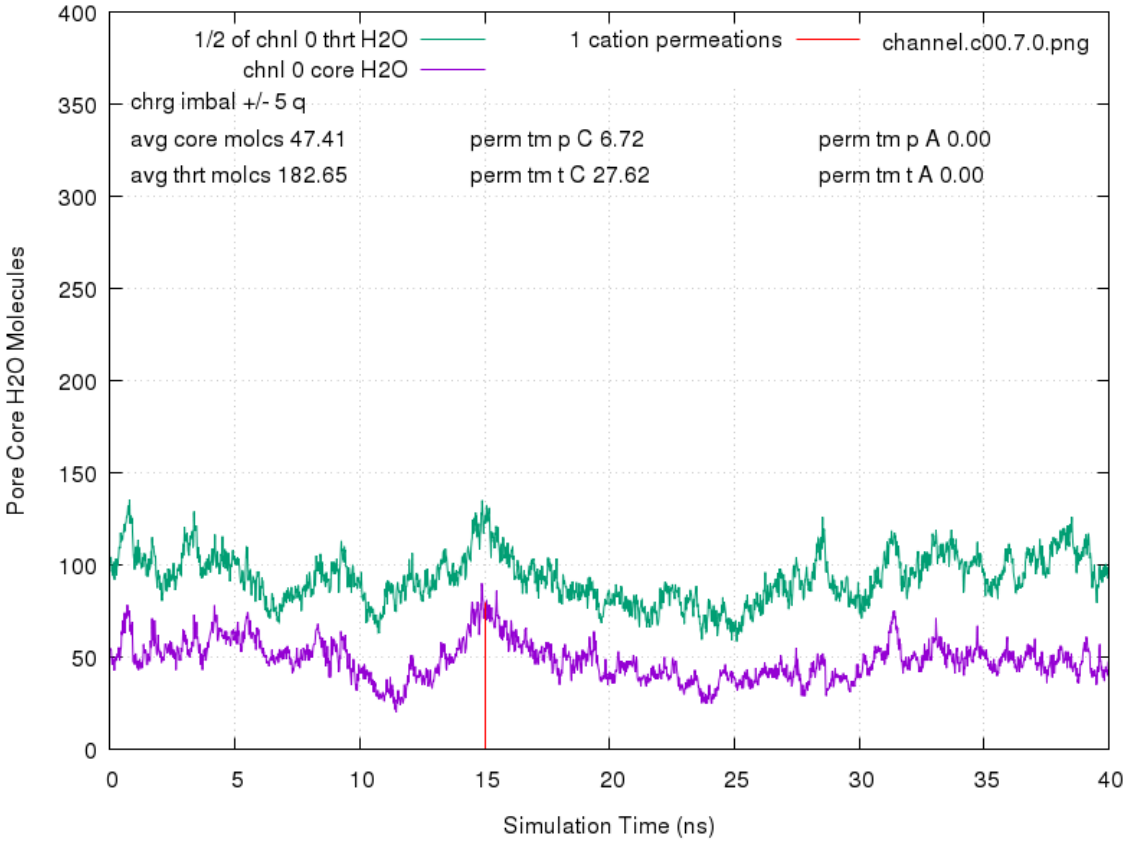

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c06.k.1.png
chrg imbal +/- 5 q
avg core molcs 58.17
avg thrt molcs 207.92
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

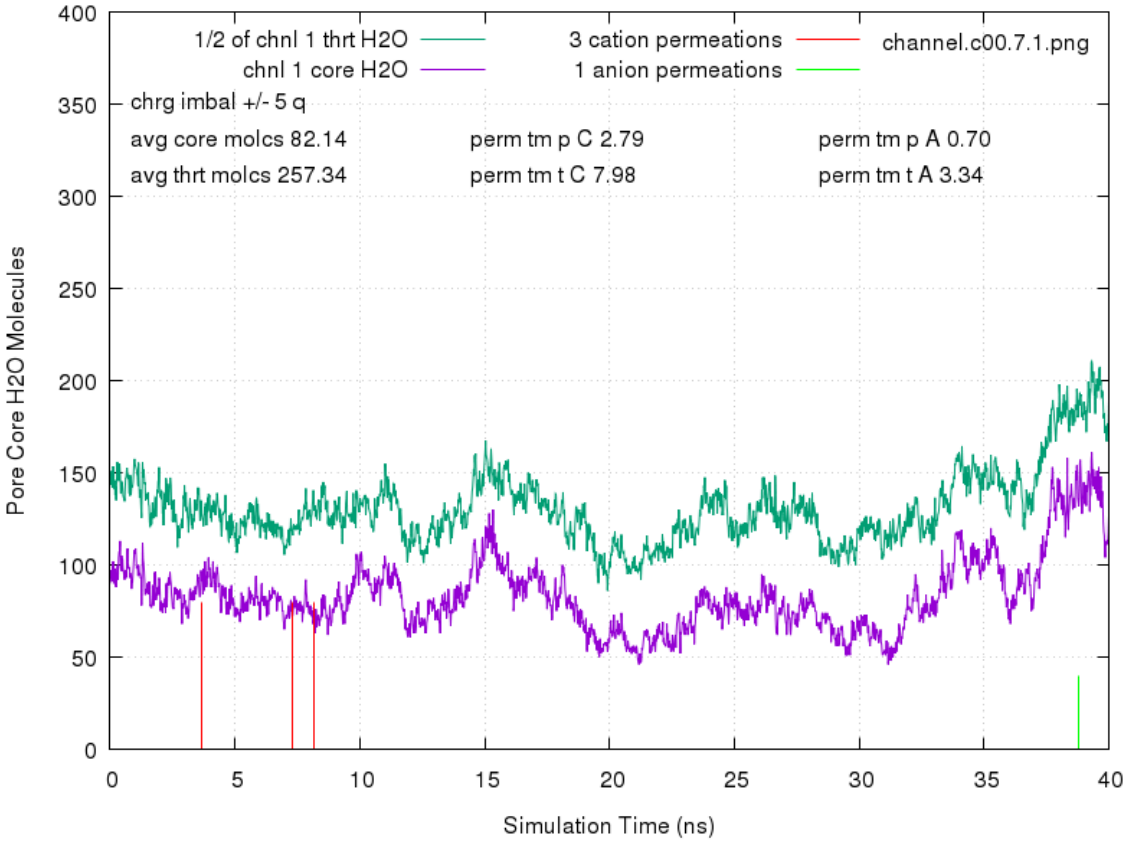

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
9 cation permeations
5 anion permeations
channel.c06.l.0.png
chrg imbal +/- 5 q
avg core molcs 112.08
avg thrt molcs 301.64
perm tm p C 0.52
perm tm t C 1.38
perm tm p A 0.37
perm tm t A 1.04
Pore Core H2O Molecules
Simulation Time (ns)

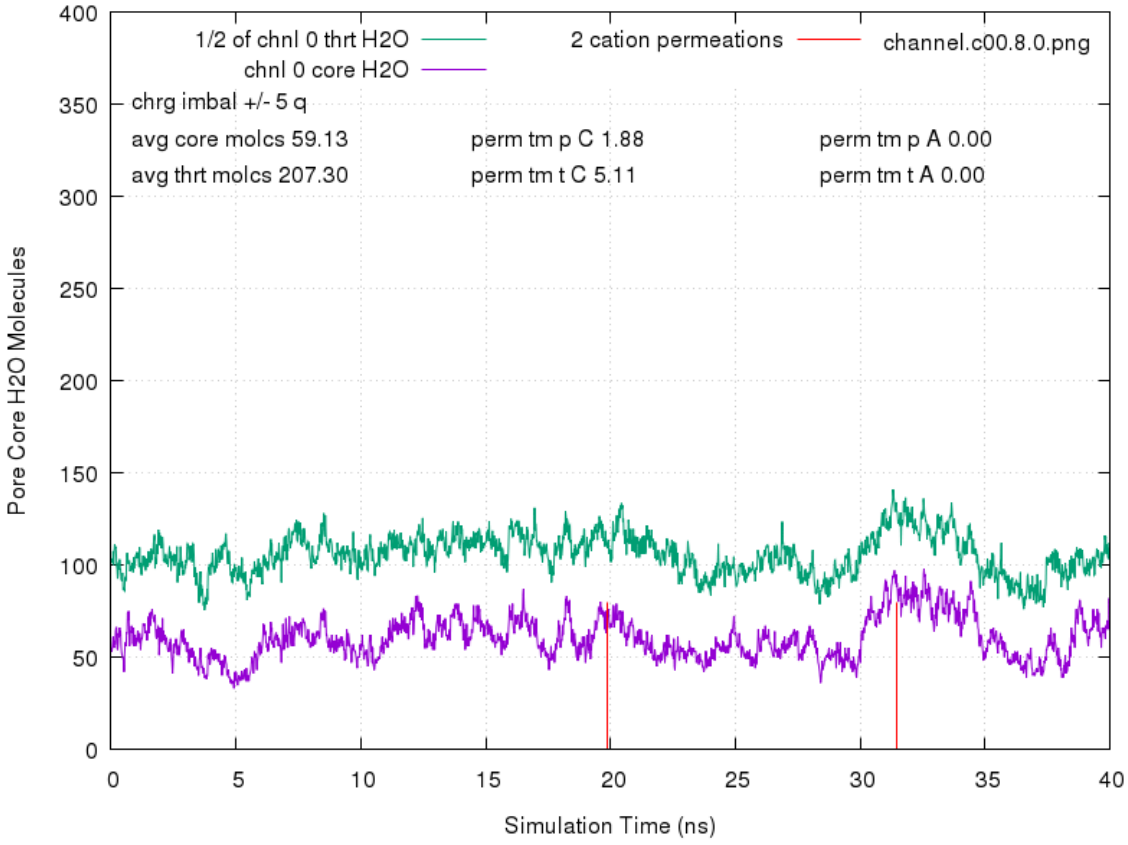

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c06.l.1.png
chrg imbal +/- 5 q
avg core molcs 22.09
avg thrt molcs 88.84
perm tm p C 4.98
perm tm t C 16.44
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

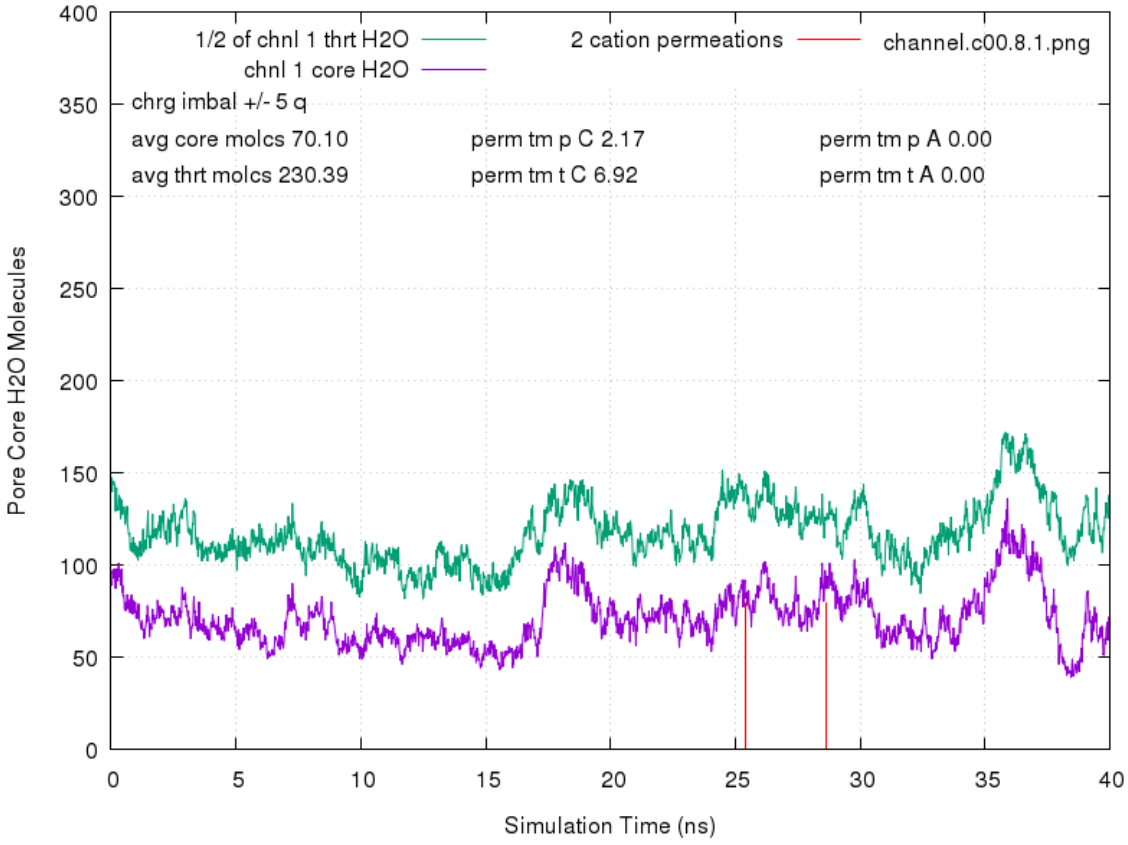

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
1 anion permeations
channel.c00.6.0.png
chrg imbal +/- 5 q
avg core molcs 62.32
avg thrt molcs 211.79
perm tm p C 2.53
perm tm t C 10.98
perm tm p A 0.18
perm tm t A 1.74
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

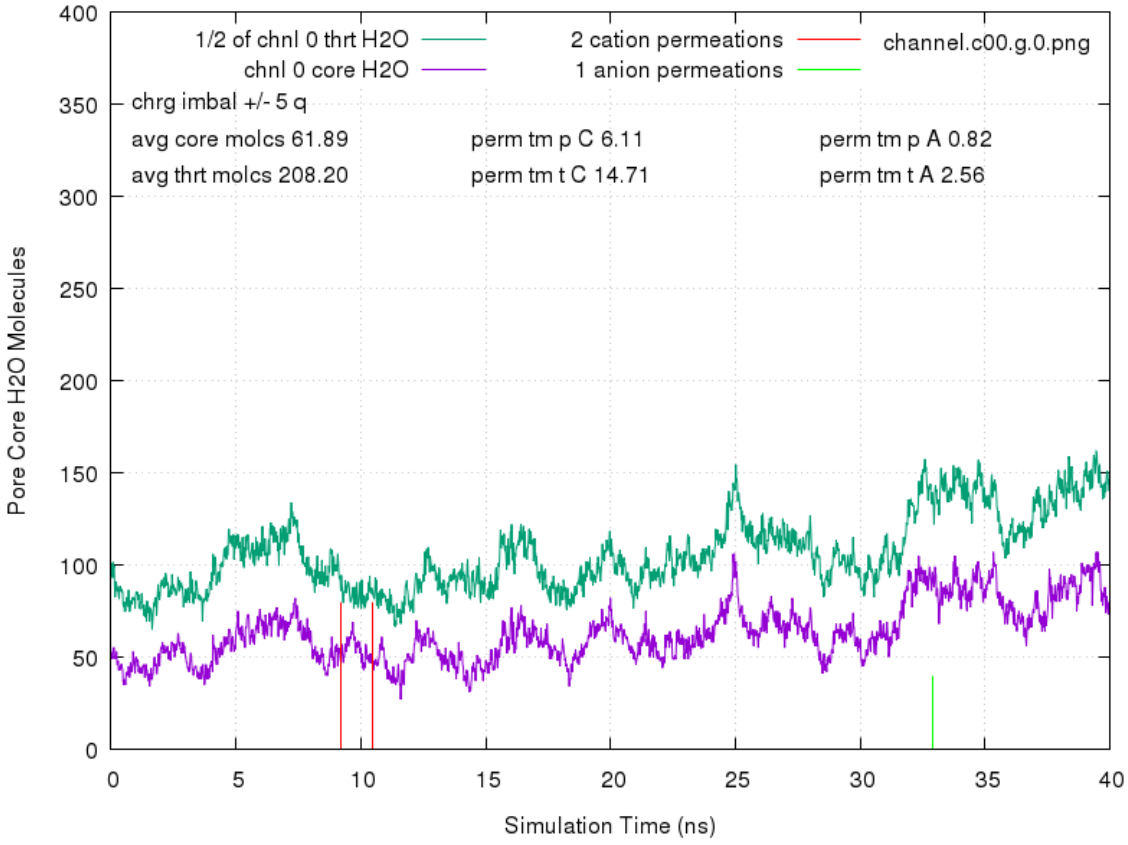

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
1 anion permeations
channel.c00.6.1.png
chrg imbal +/- 5 q
avg core molcs 70.19
avg thrt molcs 228.39
perm tm p C 9.56
perm tm t C 15.03
perm tm p A 0.46
perm tm t A 4.14
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

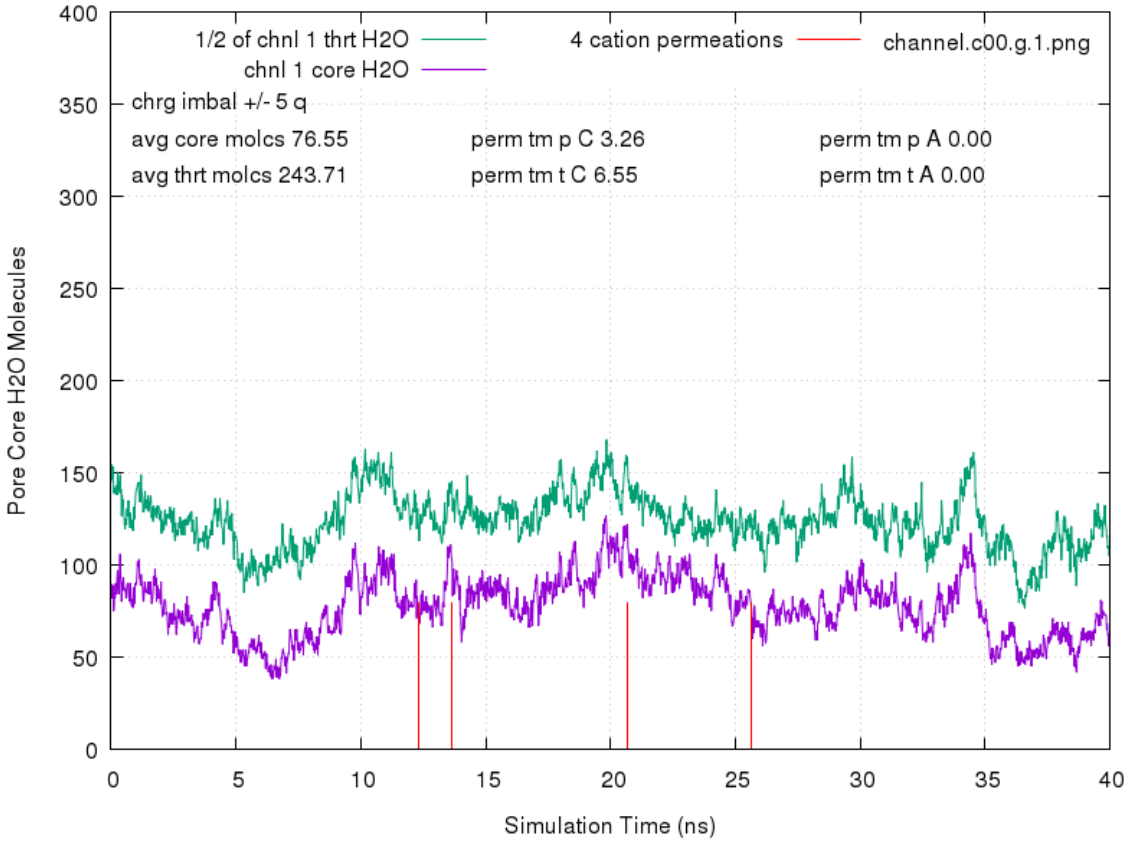

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c00.7.0.png
chrg imbal +/- 5 q
avg core molcs 47.41
avg thrt molcs 182.65
perm tm p C 6.72
perm tm t C 27.62
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

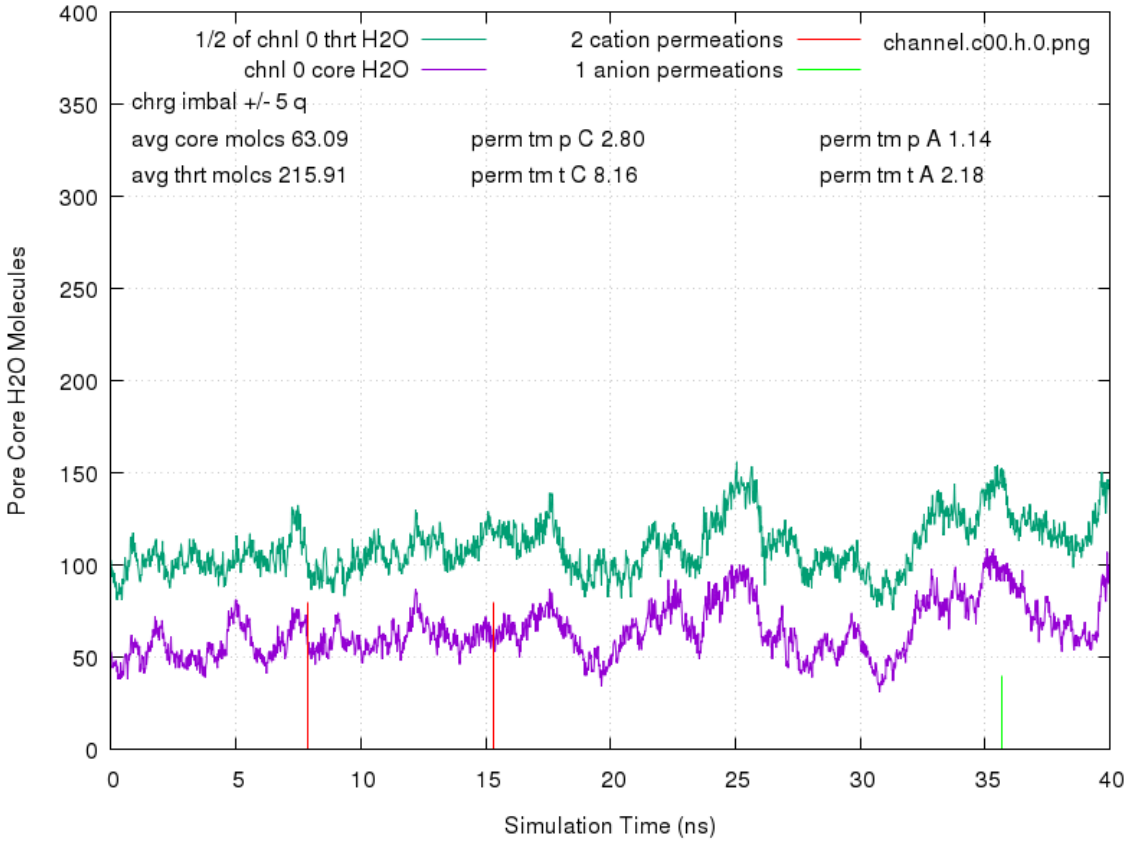

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
1 anion permeations
channel.c00.7.1.png
chrg imbal +/- 5 q
avg core molcs 82.14
avg thrt molcs 257.34
perm tm p C 2.79
perm tm t C 7.98
perm tm p A 0.70
perm tm t A 3.34
Pore Core H2O Molecules
Simulation Time (ns)

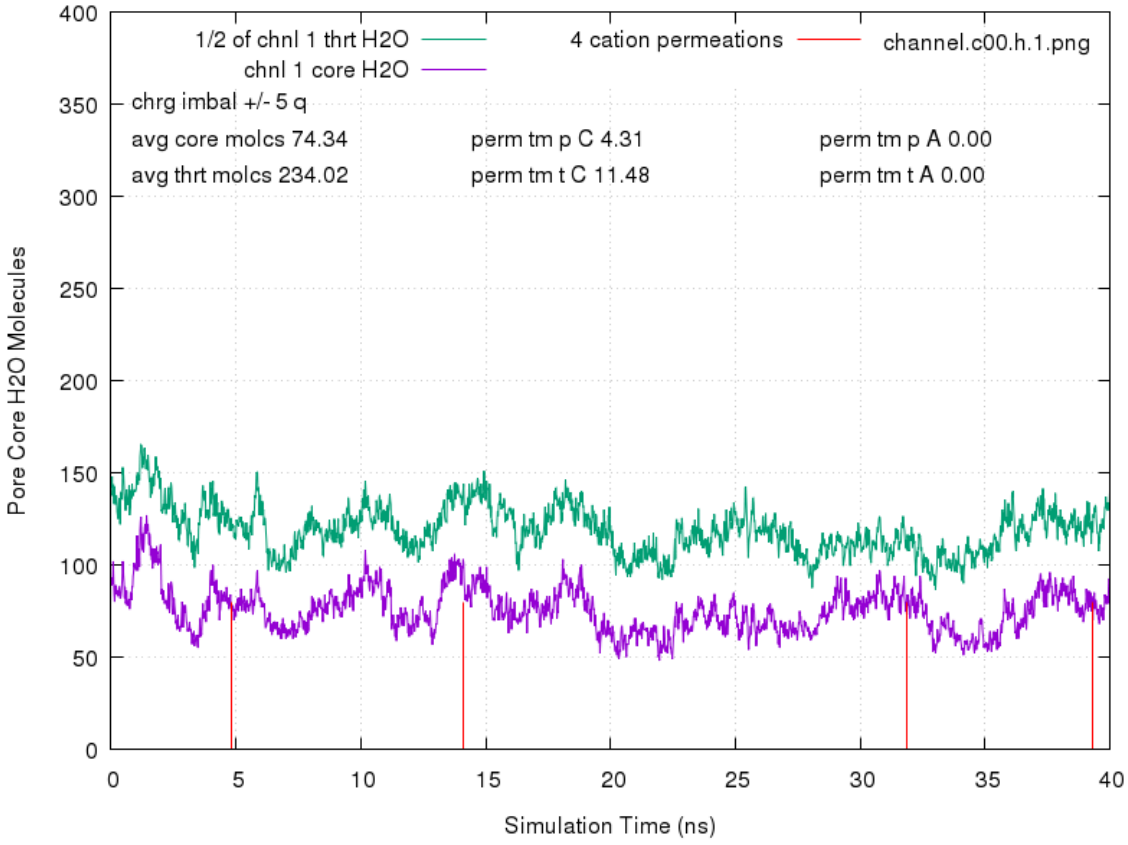

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
channel.c00.8.0.png
chrg imbal +/- 5 q
avg core molcs 59.13
avg thrt molcs 207.30
perm tm p C 1.88
perm tm t C 5.11
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

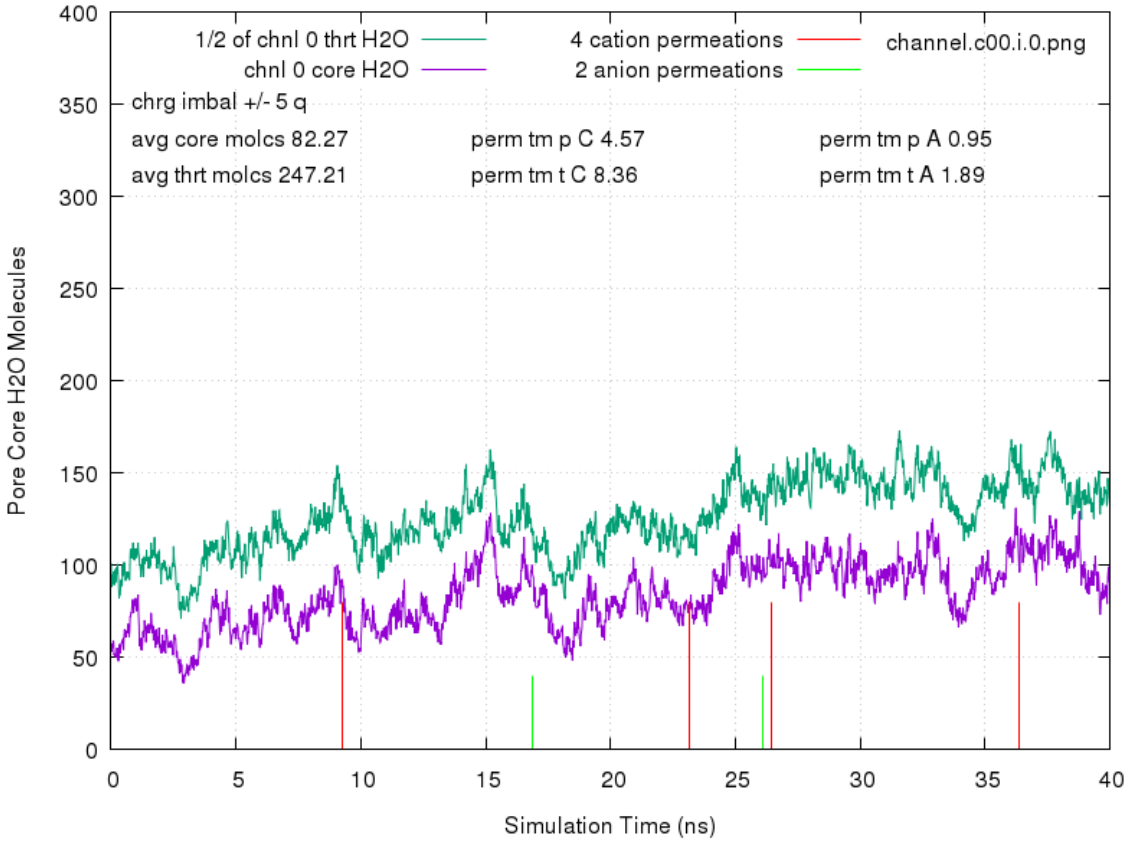

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
channel.c00.8.1.png
chrg imbal +/- 5 q
avg core molcs 70.10
avg thrt molcs 230.39
perm tm p C 2.17
perm tm t C 6.92
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

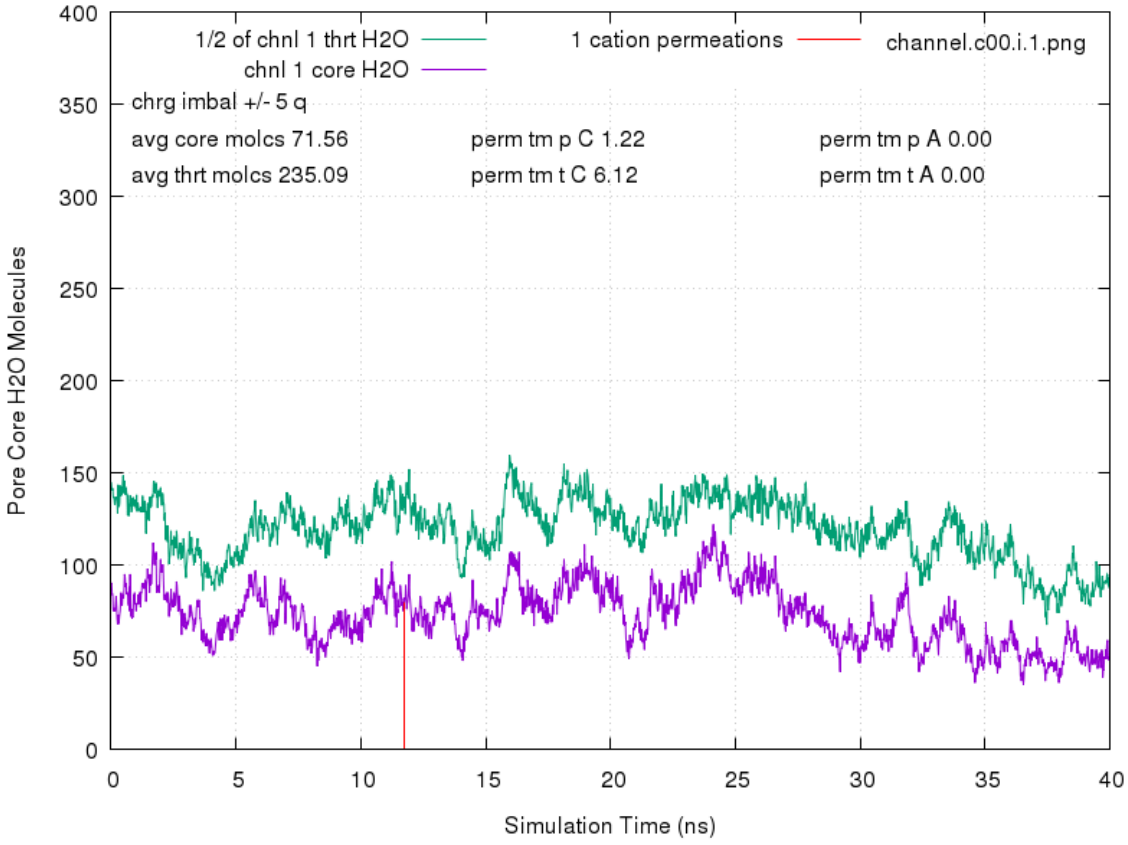

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
1 anion permeations
channel.c00.g.0.png
chrg imbal +/- 5 q
avg core molcs 61.89
avg thrt molcs 208.20
perm tm p C 6.11
perm tm t C 14.71
perm tm p A 0.82
perm tm t A 2.56
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

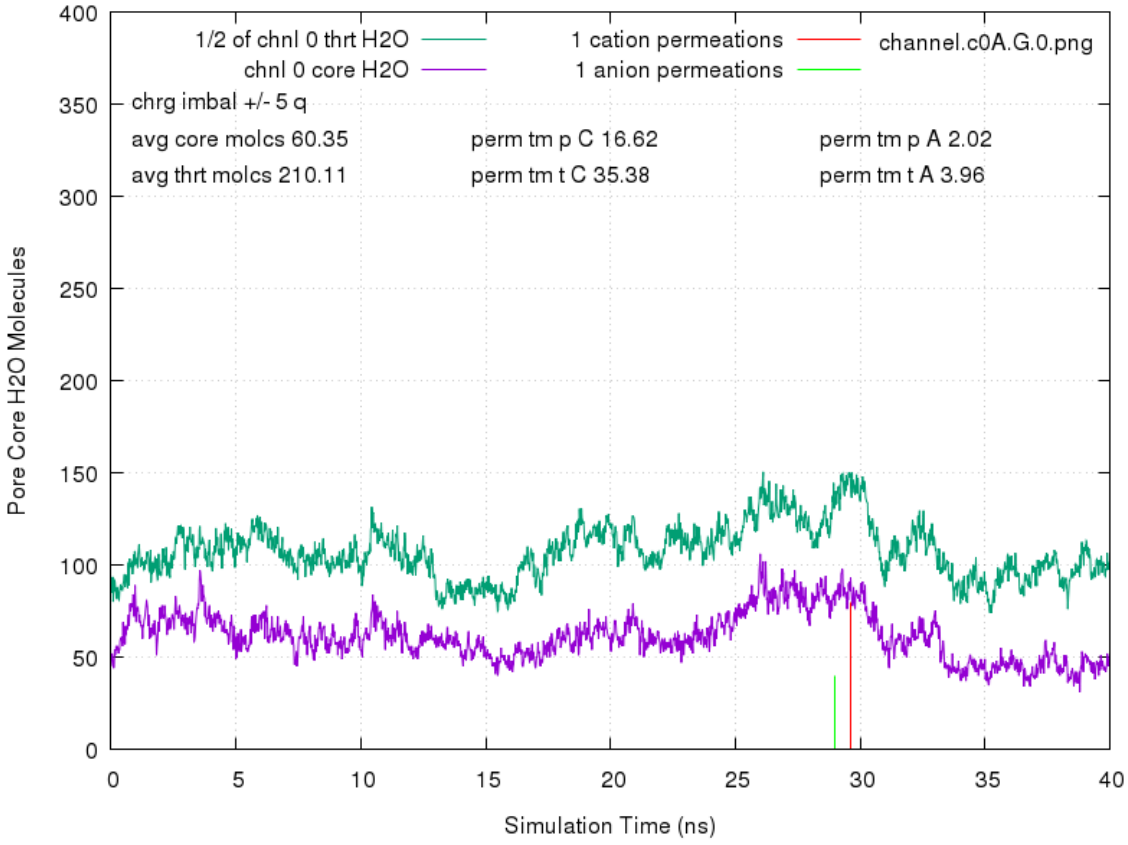

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
channel.c00.g.1.png
chrg imbal +/- 5 q
avg core molcs 76.55
avg thrt molcs 243.71
perm tm p C 3.26
perm tm t C 6.55
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

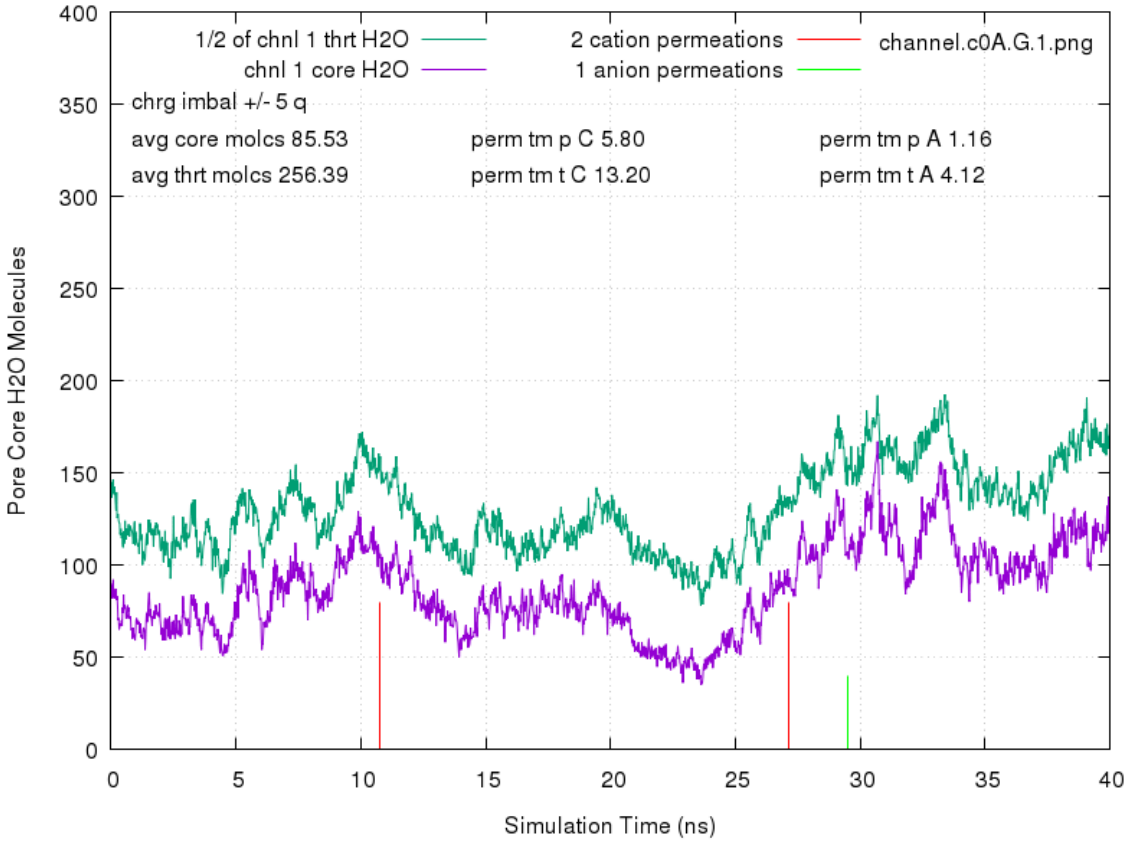

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
1 anion permeations
channel.c00.h.0.png
chrg imbal +/- 5 q
avg core molcs 63.09
avg thrt molcs 215.91
perm tm p C 2.80
perm tm t C 8.16
perm tm p A 1.14
perm tm t A 2.18
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

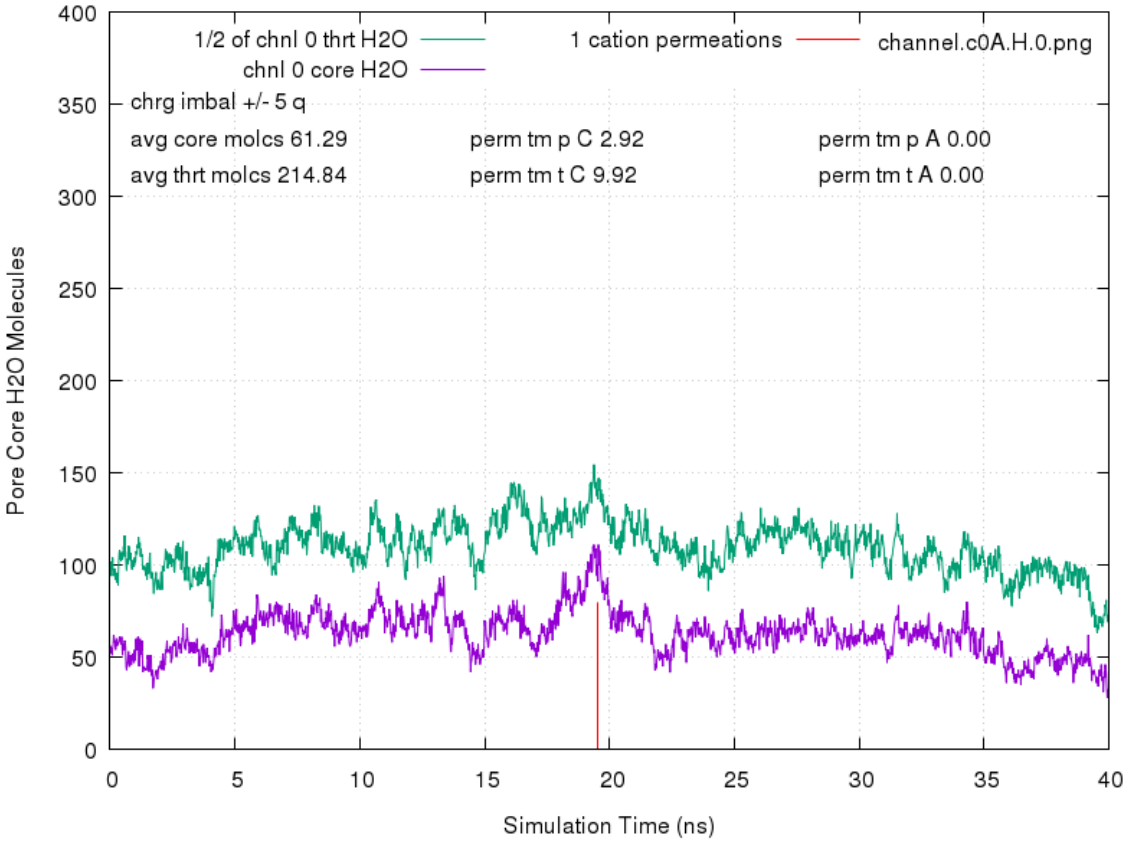

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
channel.c00.h.1.png
chrg imbal +/- 5 q
avg core molcs 74.34
avg thrt molcs 234.02
perm tm p C 4.31
perm tm t C 11.48
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

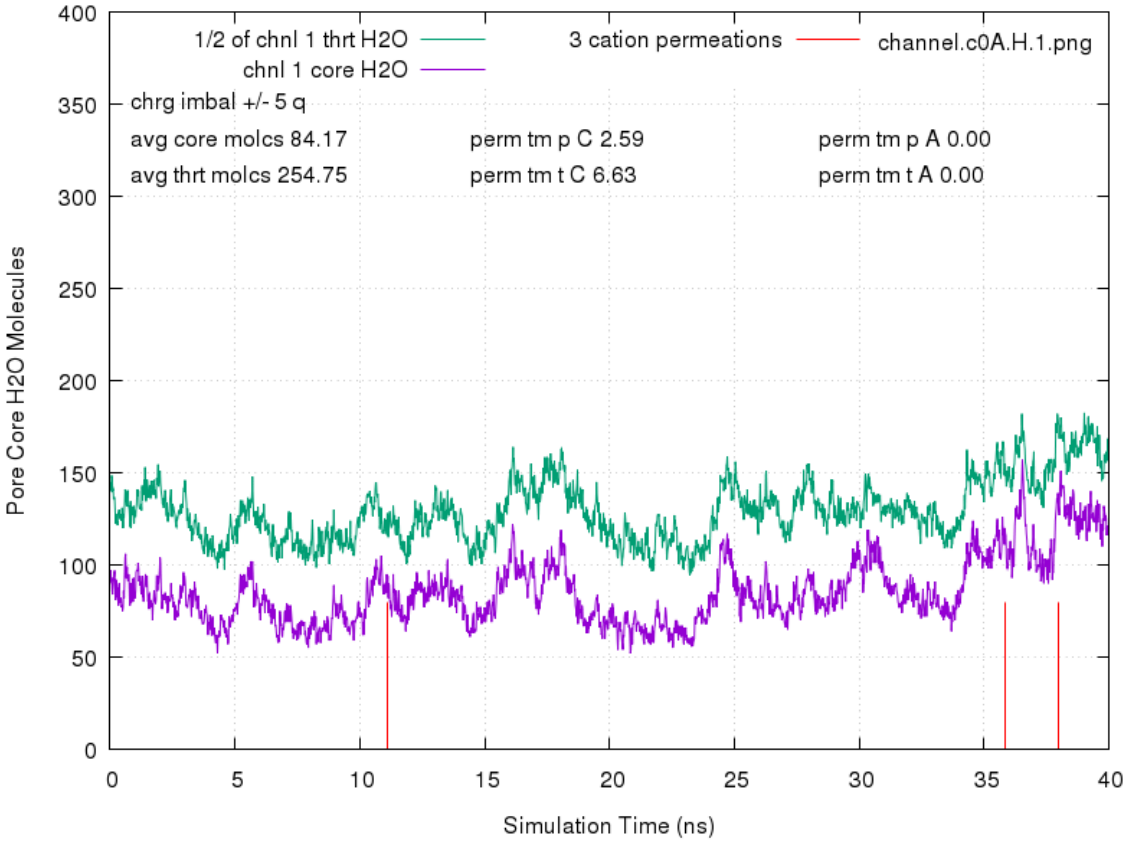

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
4 cation permeations
2 anion permeations
channel.c00.i.0.png
chrg imbal +/- 5 q
avg core molcs 82.27
avg thrt molcs 247.21
perm tm p C 4.57
perm tm t C 8.36
perm tm p A 0.95
perm tm t A 1.89
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

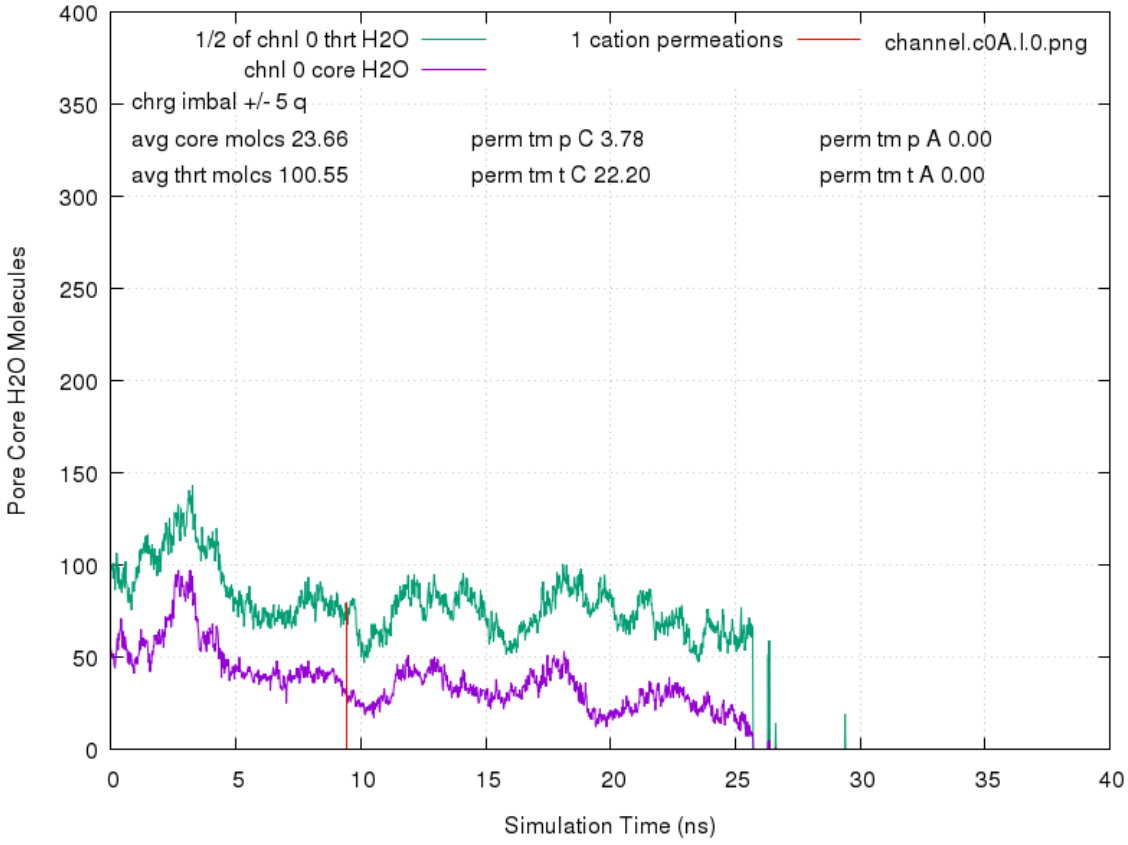

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c00.i.1.png
chrg imbal +/- 5 q
avg core molcs 71.56
avg thrt molcs 235.09
perm tm p C 1.22
perm tm t C 6.12
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

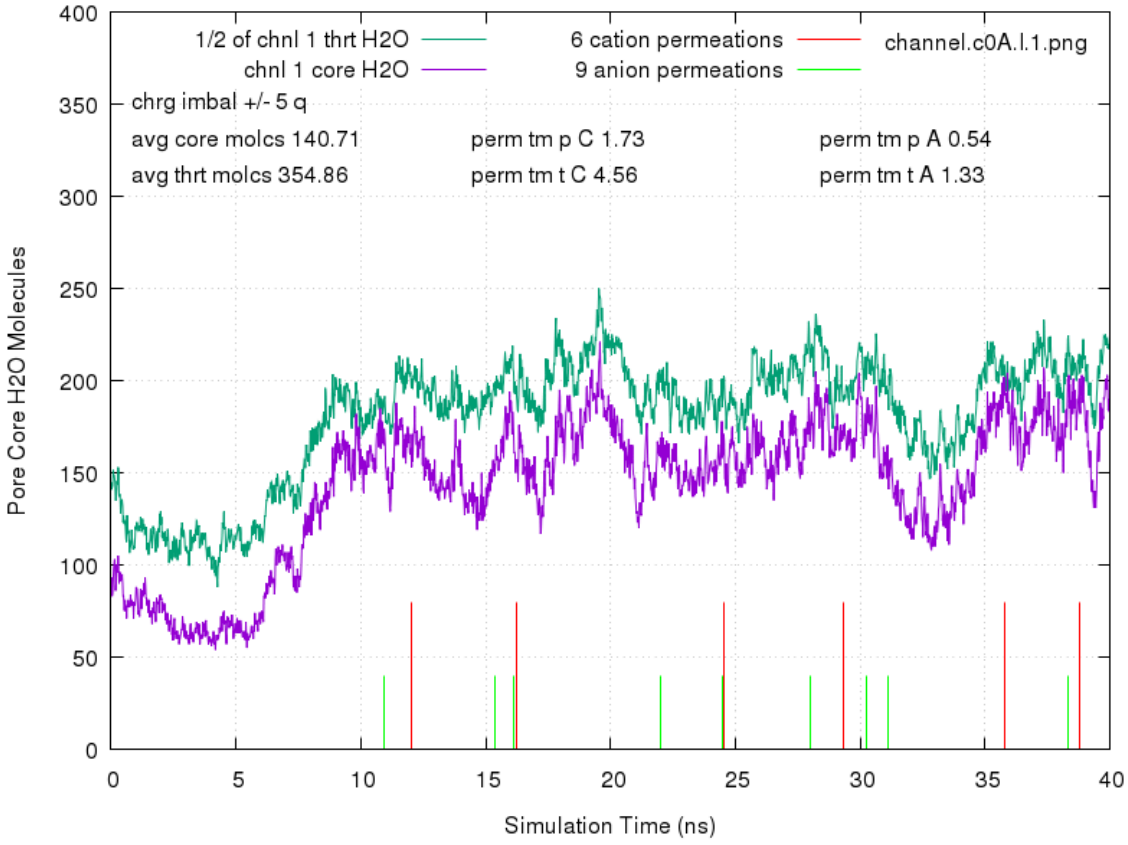

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
1 anion permeations
channel.c0A.G.0.png
chrg imbal +/- 5 q
avg core molcs 60.35
avg thrt molcs 210.11
perm tm p C 16.62
perm tm t C 35.38
perm tm p A 2.02
perm tm t A 3.96
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

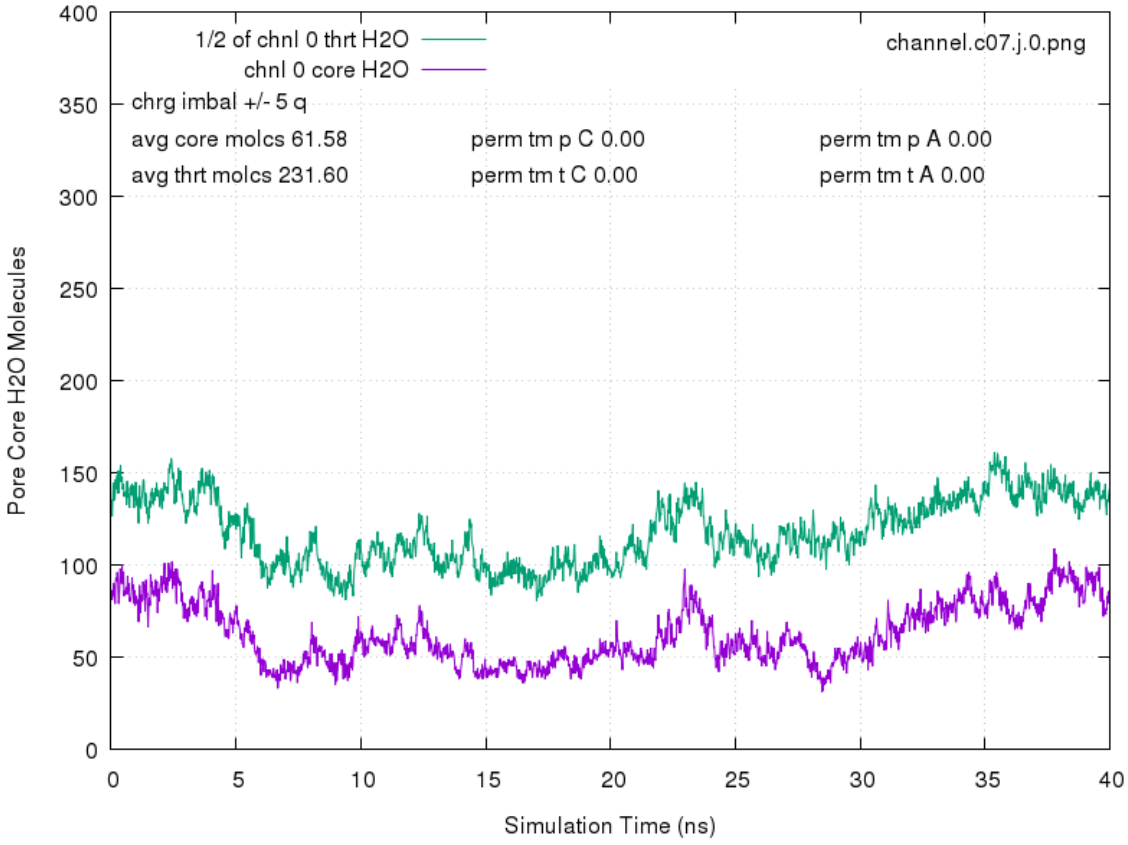

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
1 anion permeations
channel.c0A.G.1.png
chrg imbal +/- 5 q
avg core molcs 85.53
avg thrt molcs 256.39
perm tm p C 5.80
perm tm t C 13.20
perm tm p A 1.16
perm tm t A 4.12
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

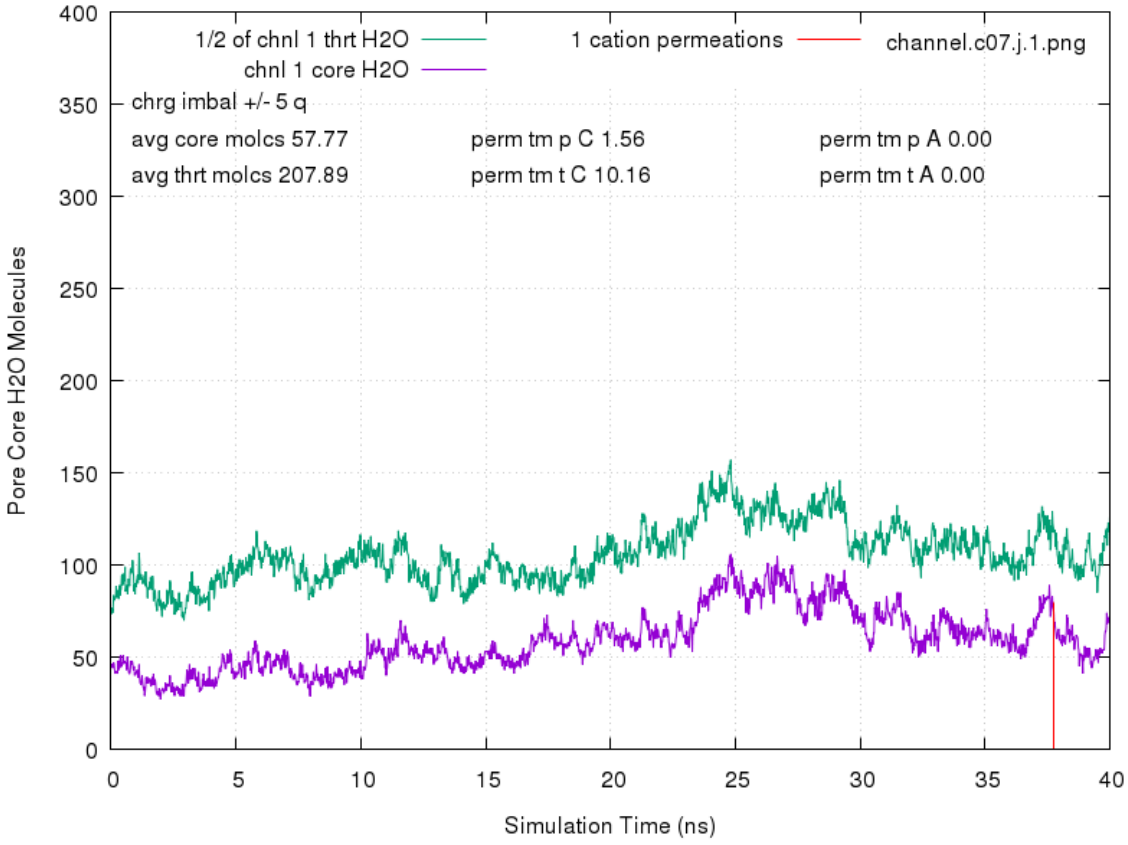

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c0A.H.0.png
chrg imbal +/- 5 q
avg core molcs 61.29
avg thrt molcs 214.84
perm tm p C 2.92
perm tm t C 9.92
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

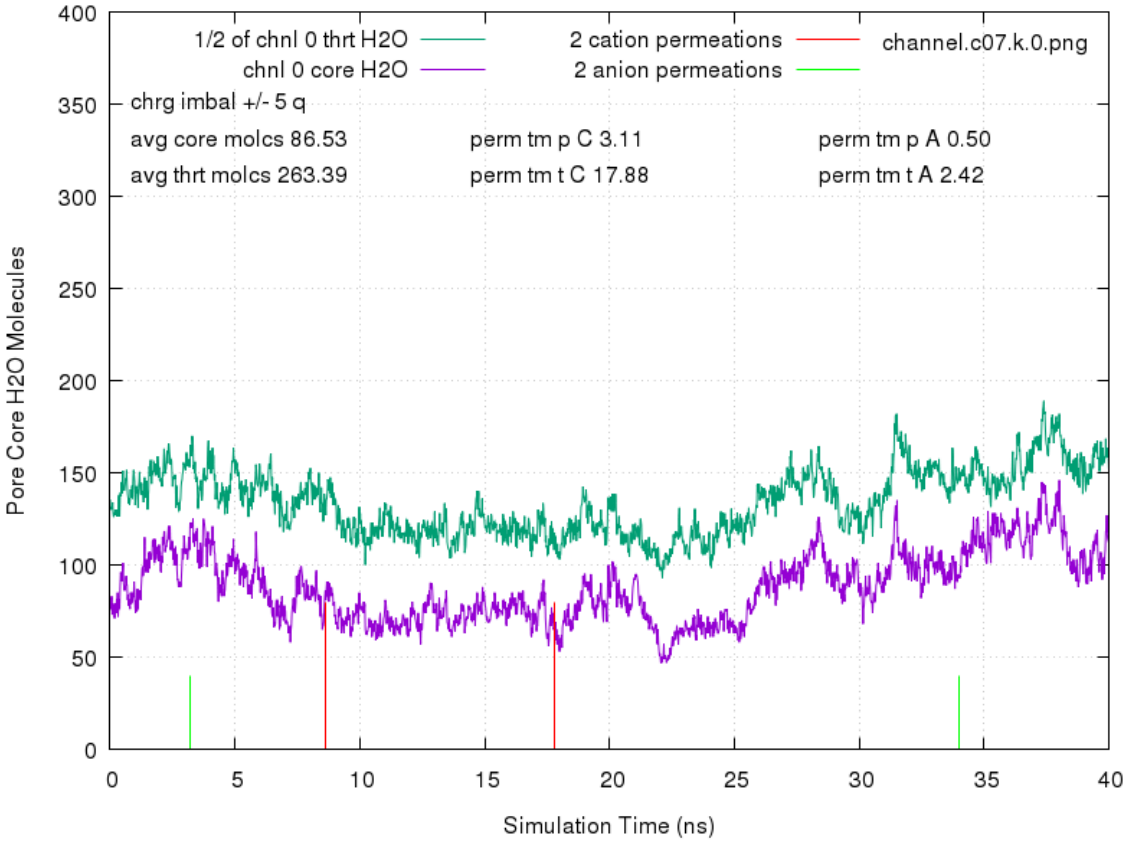

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
channel.c0A.H.1.png
chrg imbal +/- 5 q
avg core molcs 84.17
avg thrt molcs 254.75
perm tm p C 2.59
perm tm t C 6.63
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

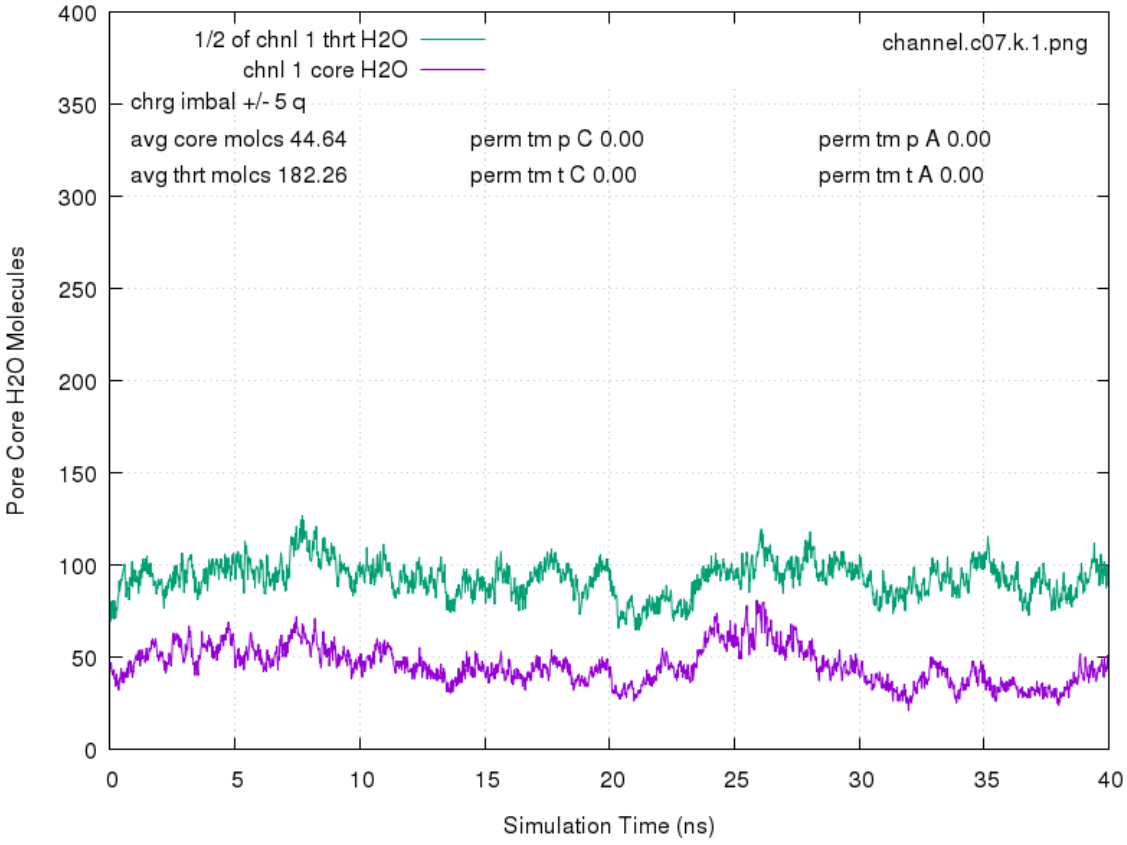

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c0A.l.0.png
chrg imbal +/- 5 q
avg core molcs 23.66
avg thrt molcs 100.55
perm tm p C 3.78
perm tm t C 22.20
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

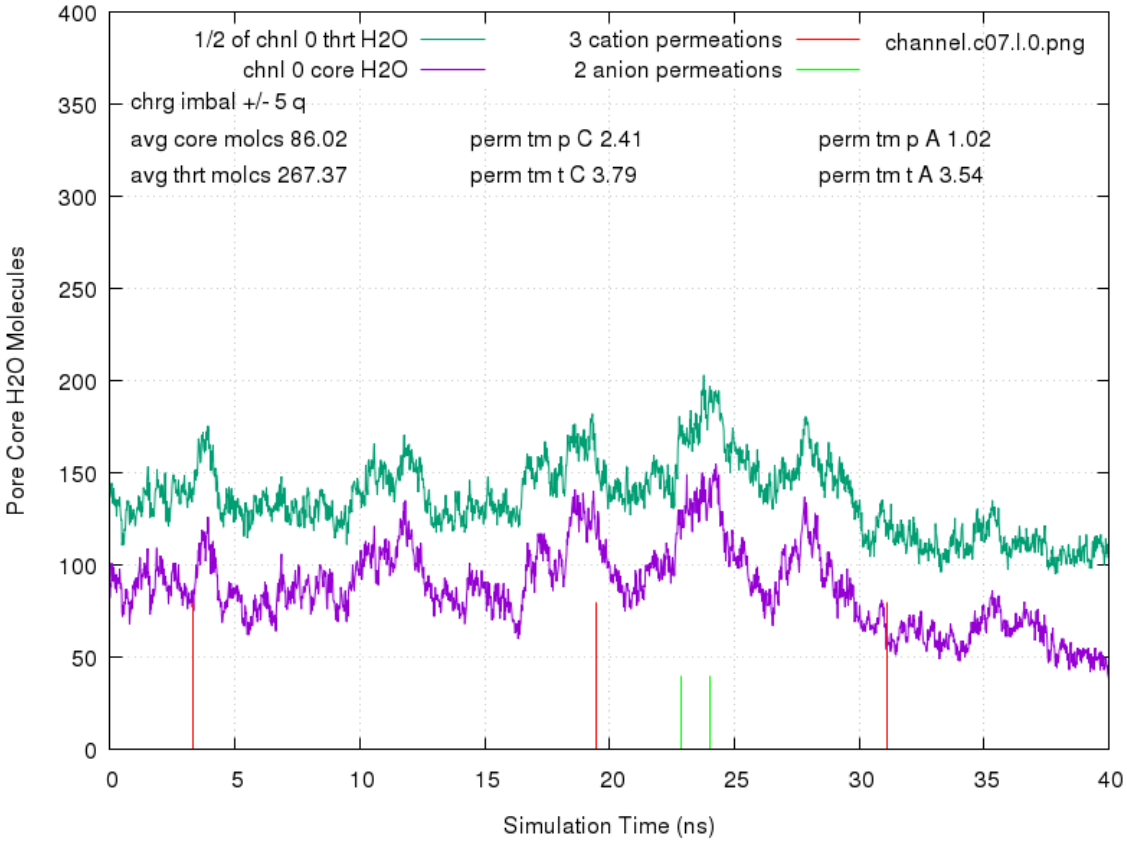

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
6 cation permeations
9 anion permeations
channel.c0A.l.1.png
chrg imbal +/- 5 q
avg core molcs 140.71
avg thrt molcs 354.86
perm tm p C 1.73
perm tm t C 4.56
perm tm p A 0.54
perm tm t A 1.33
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

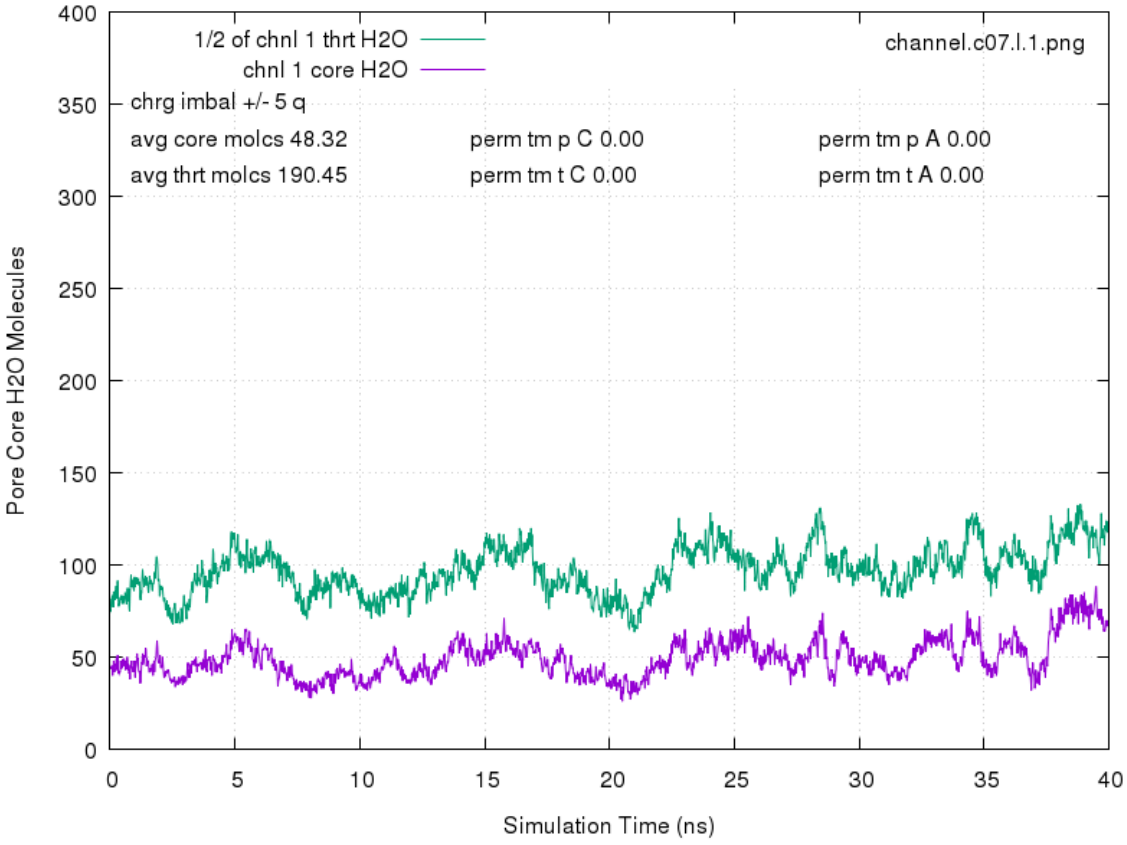

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c07.j.0.png
chrg imbal +/- 5 q
avg core molcs 61.58
avg thrt molcs 231.60
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

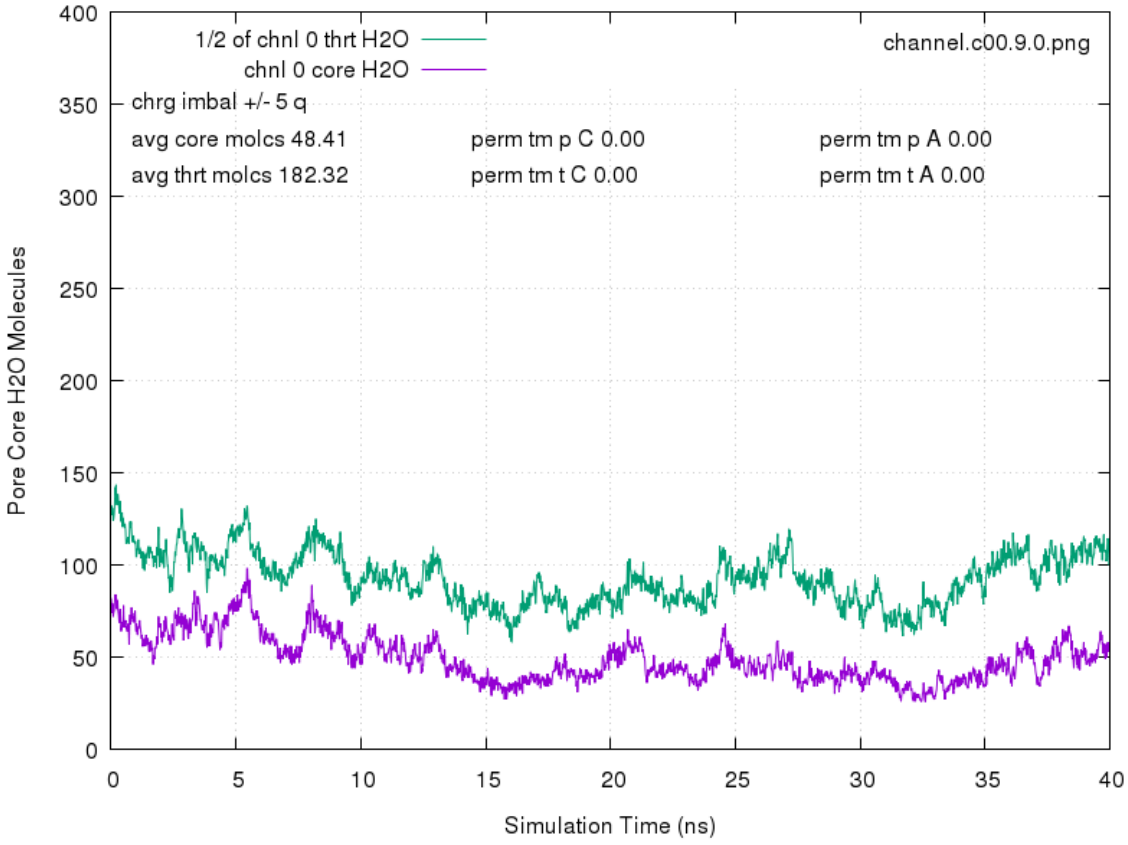

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c07.j.1.png
chrg imbal +/- 5 q
avg core molcs 57.77
avg thrt molcs 207.89
perm tm p C 1.56
perm tm t C 10.16
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

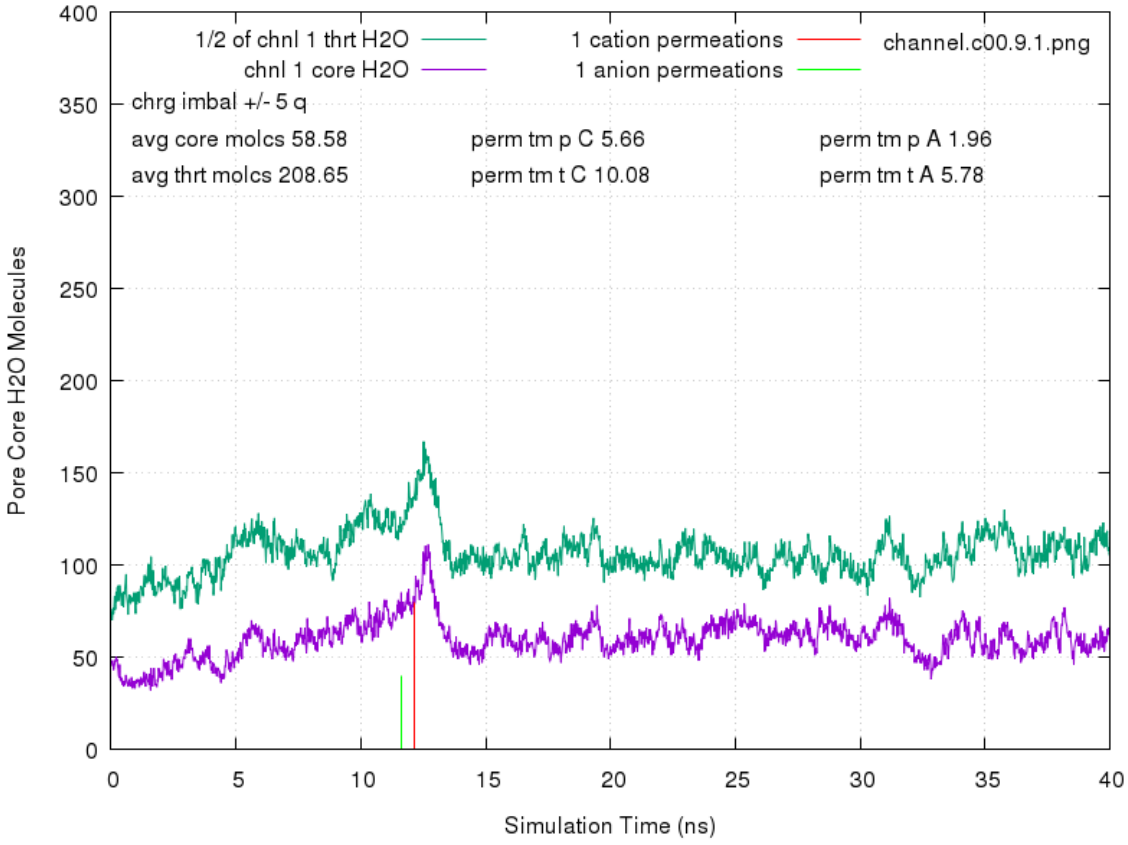

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
2 anion permeations
channel.c07.k.0.png
chrg imbal +/- 5 q
avg core molcs 86.53
avg thrt molcs 263.39
perm tm p C 3.11
perm tm t C 17.88
perm tm p A 0.50
perm tm t A 2.42
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

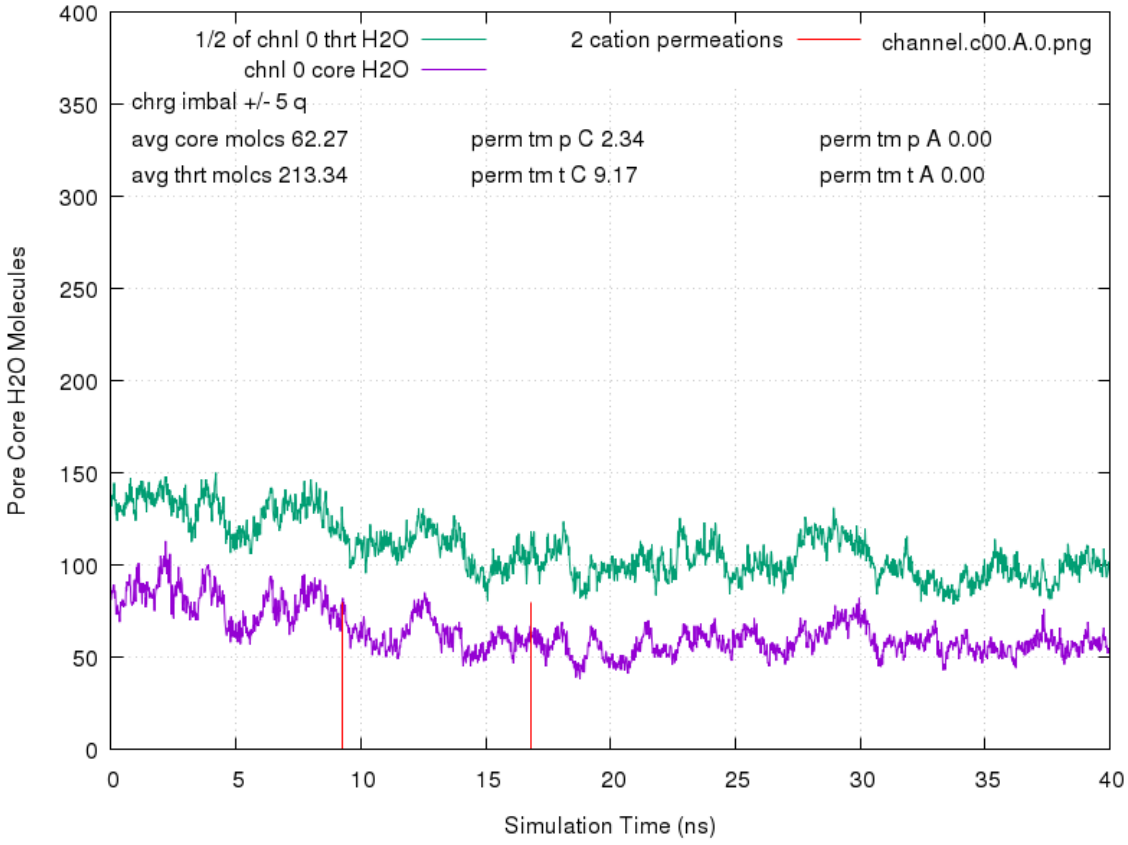

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c07.k.1.png
chrg imbal +/- 5 q
avg core molcs 44.64
avg thrt molcs 182.26
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

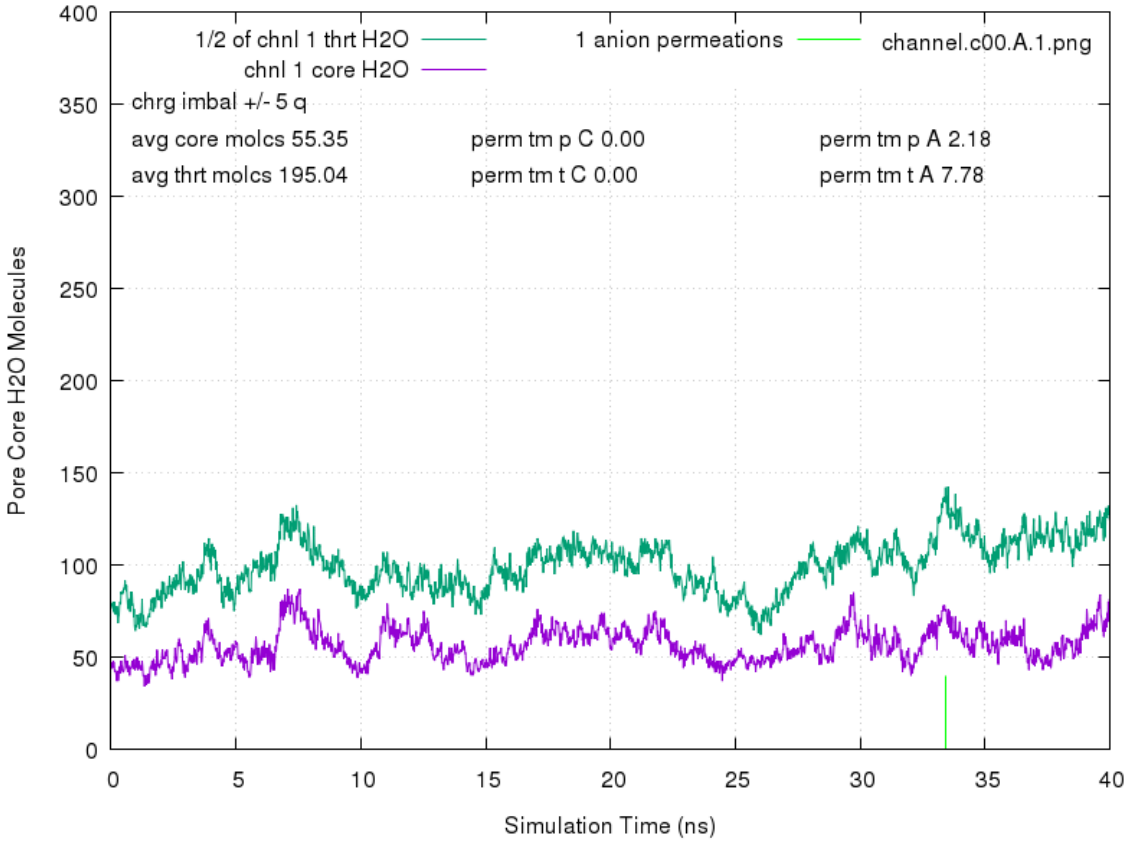

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
2 anion permeations
channel.c07.l.0.png
chrg imbal +/- 5 q
avg core molcs 86.02
avg thrt molcs 267.37
perm tm p C 2.41
perm tm t C 3.79
perm tm p A 1.02
perm tm t A 3.54
Pore Core H2O Molecules
Simulation Time (ns)

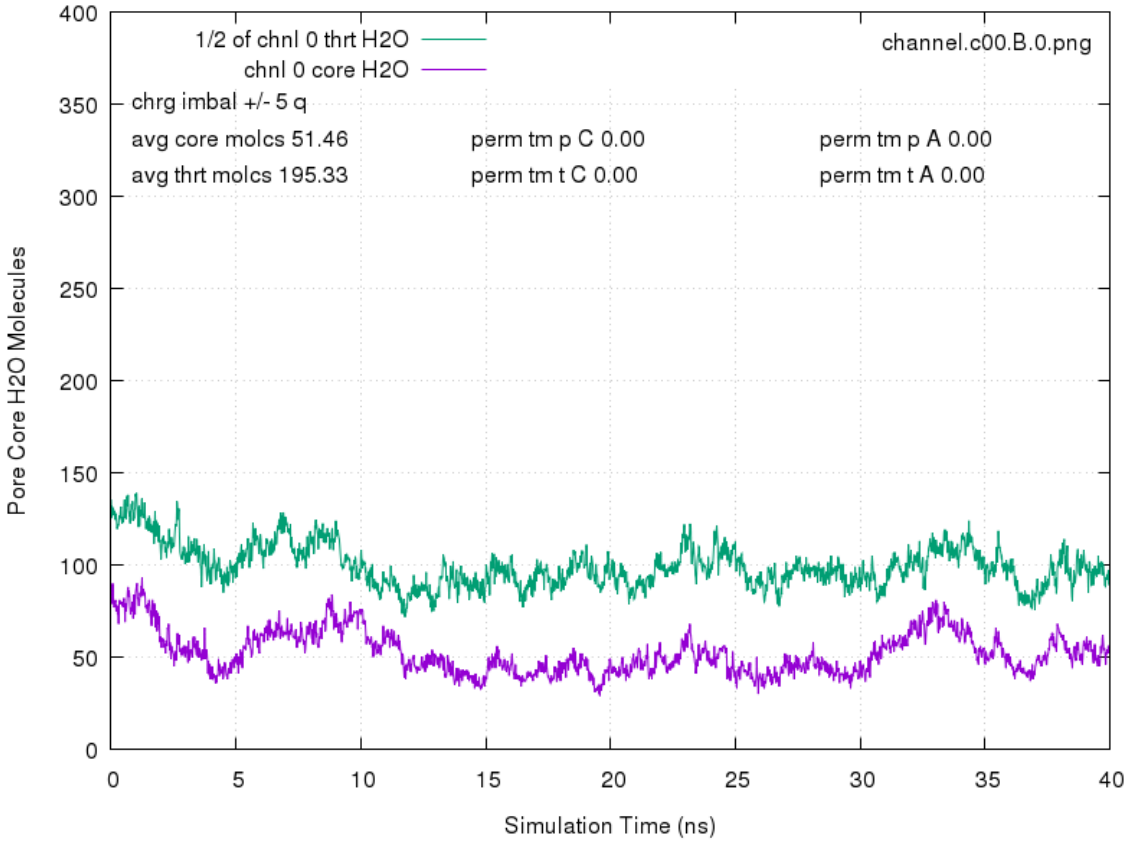

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c07.l.1.png
chrg imbal +/- 5 q
avg core molcs 48.32
avg thrt molcs 190.45
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

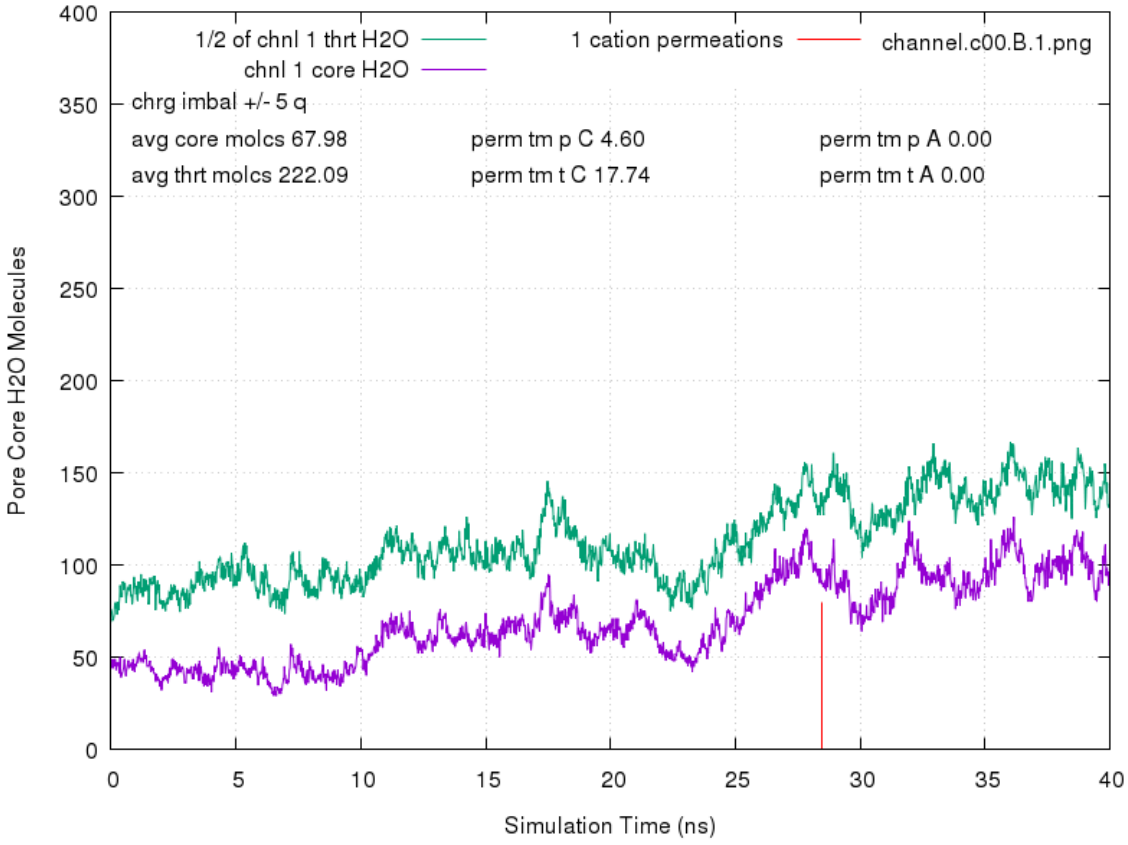

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c00.9.0.png
chrg imbal +/- 5 q
avg core molcs 48.41
avg thrt molcs 182.32
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

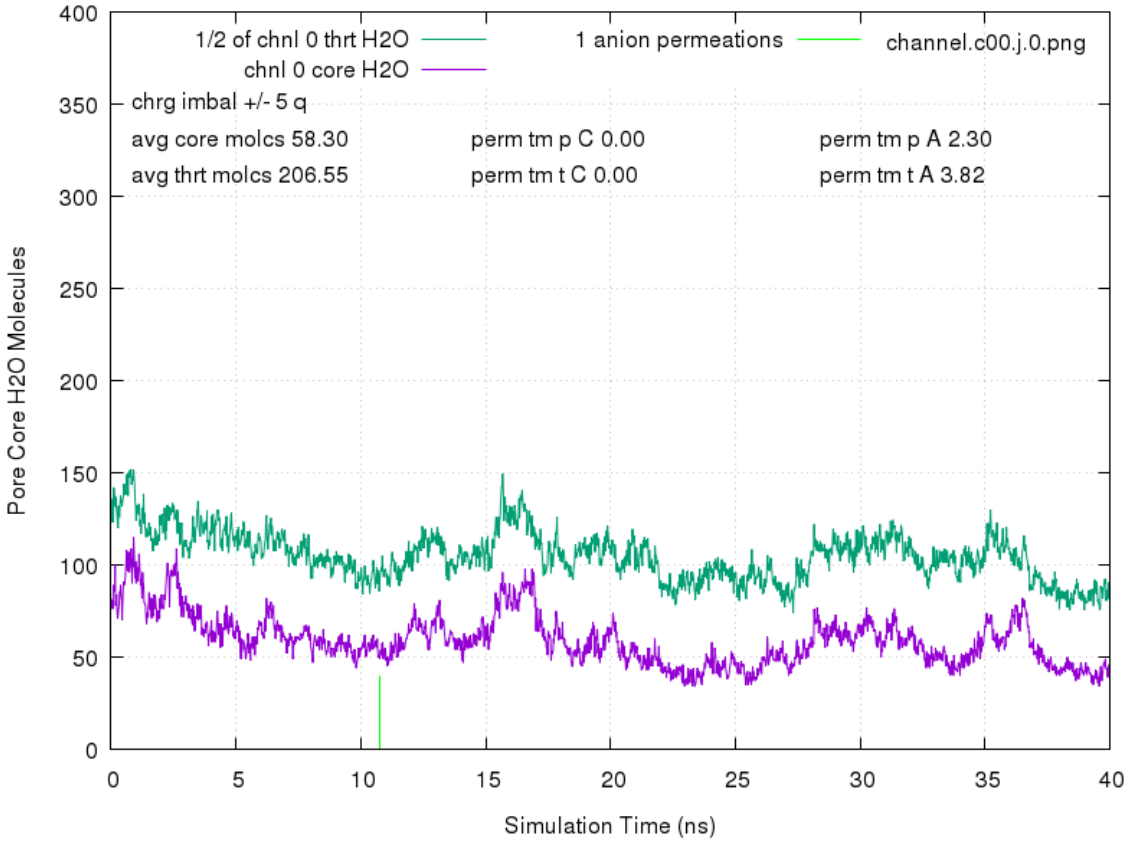

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
1 anion permeations
channel.c00.9.1.png
chrg imbal +/- 5 q
avg core molcs 58.58
avg thrt molcs 208.65
perm tm p C 5.66
perm tm t C 10.08
perm tm p A 1.96
perm tm t A 5.78
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

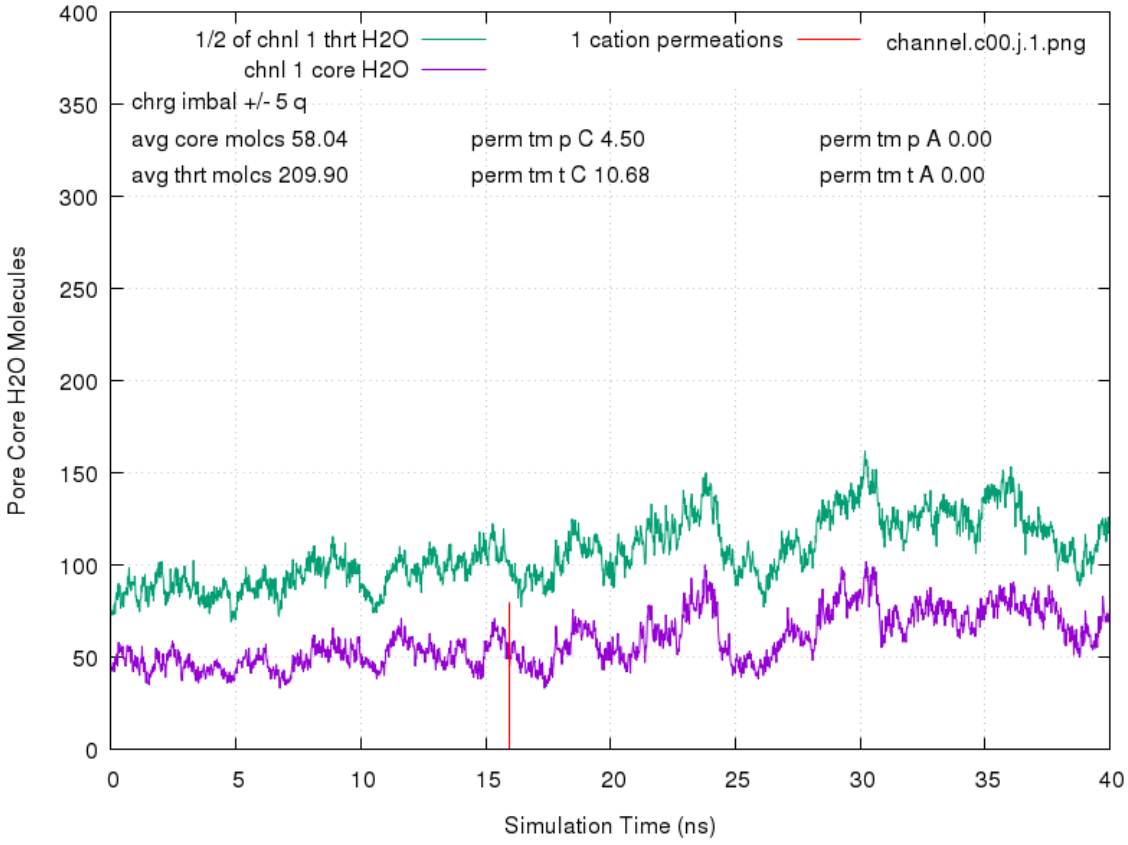

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
channel.c00.A.0.png
chrg imbal +/- 5 q
avg core molcs 62.27
avg thrt molcs 213.34
perm tm p C 2.34
perm tm t C 9.17
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

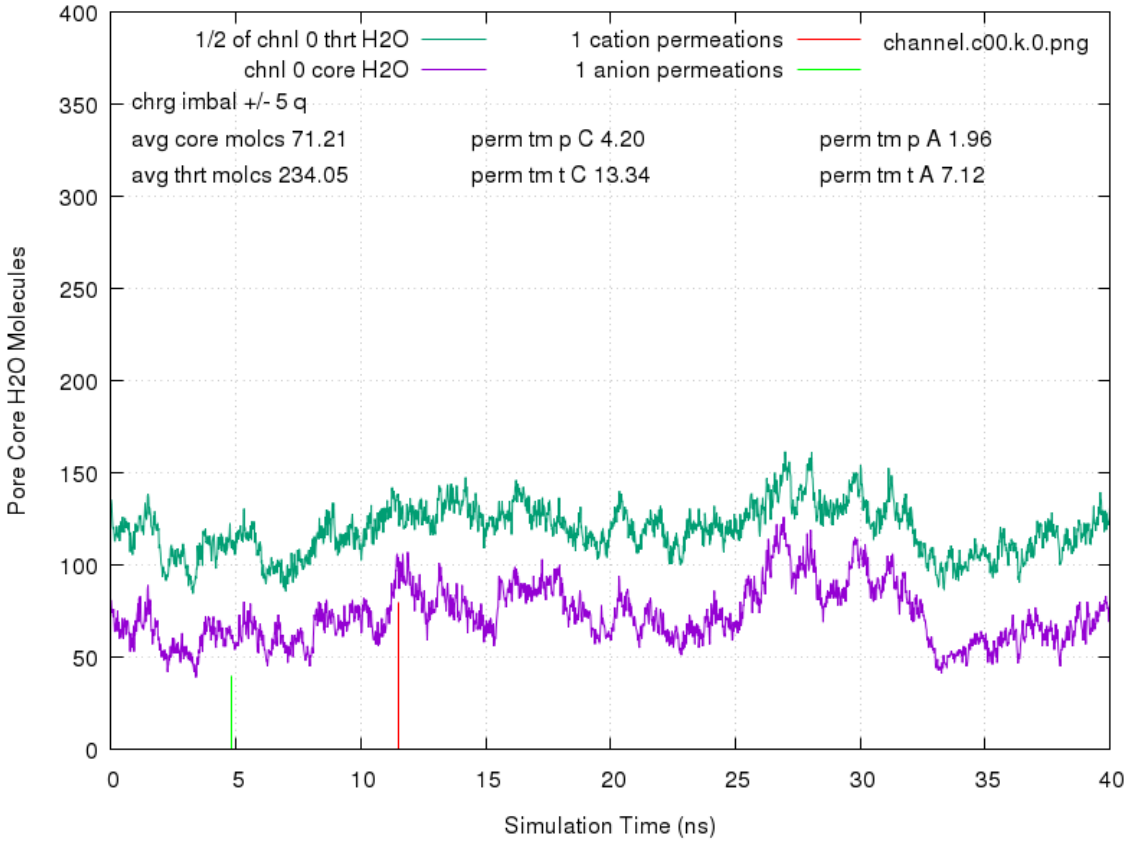

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 anion permeations
channel.c00.A.1.png
chrg imbal +/- 5 q
avg core molcs 55.35
avg thrt molcs 195.04
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 2.18
perm tm t A 7.78
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

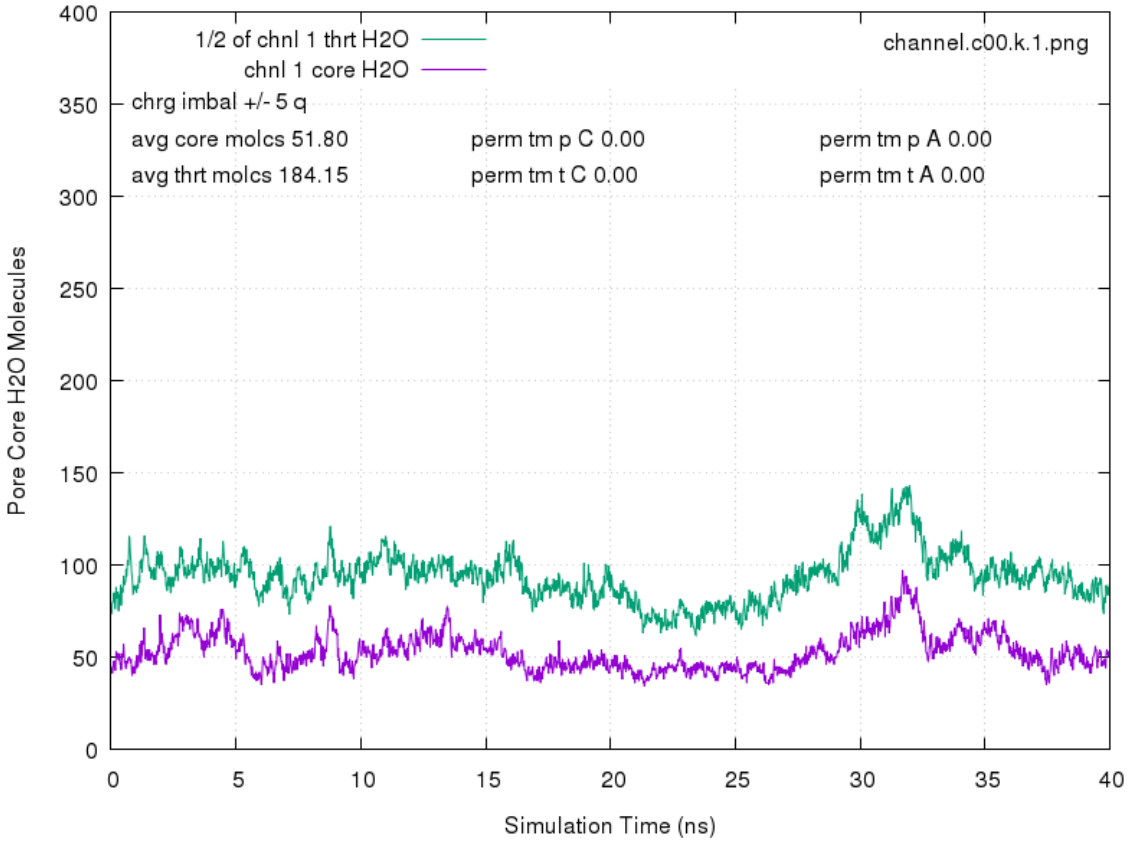

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c00.B.0.png
chrg imbal +/- 5 q
avg core molcs 51.46
avg thrt molcs 195.33
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

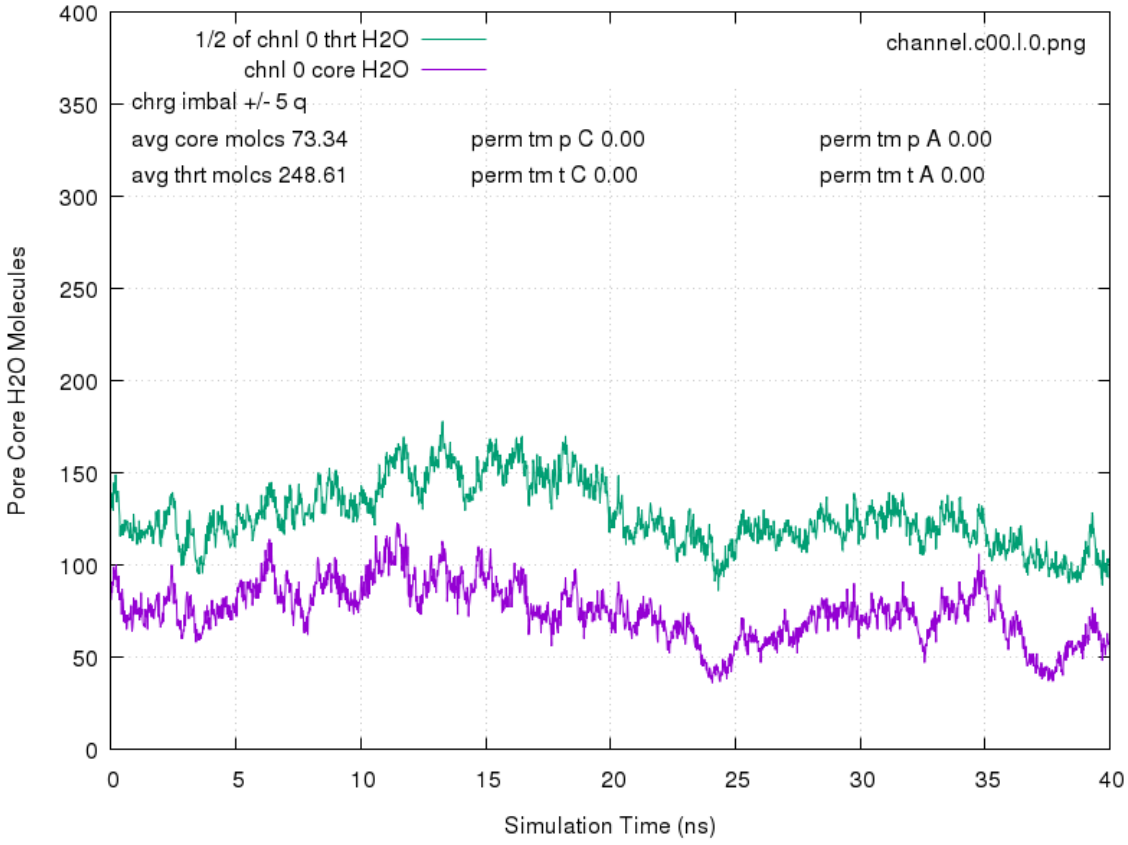

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c00.B.1.png
chrg imbal +/- 5 q
avg core molcs 67.98
avg thrt molcs 222.09
perm tm p C 4.60
perm tm t C 17.74
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

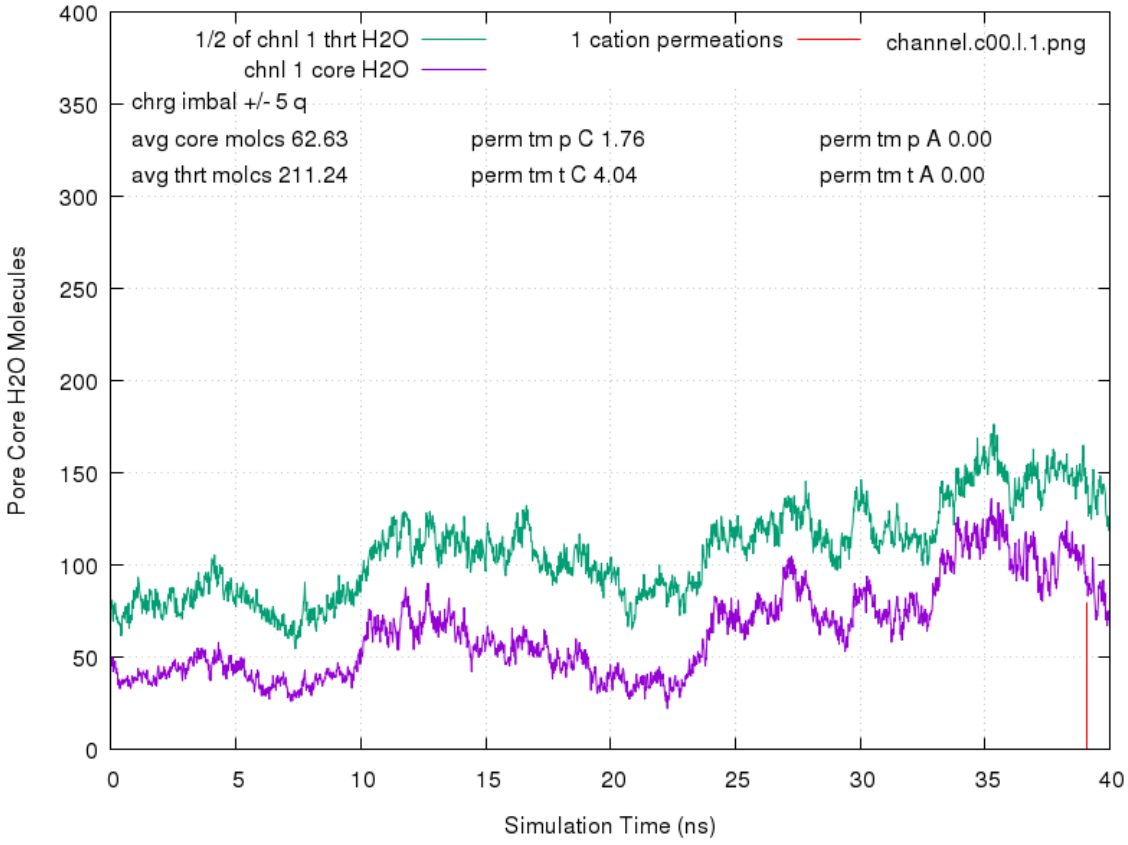

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 anion permeations
channel.c00.j.0.png
chrg imbal +/- 5 q
avg core molcs 58.30
avg thrt molcs 206.55
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 2.30
perm tm t A 3.82
Pore Core H2O Molecules
Simulation Time (ns)

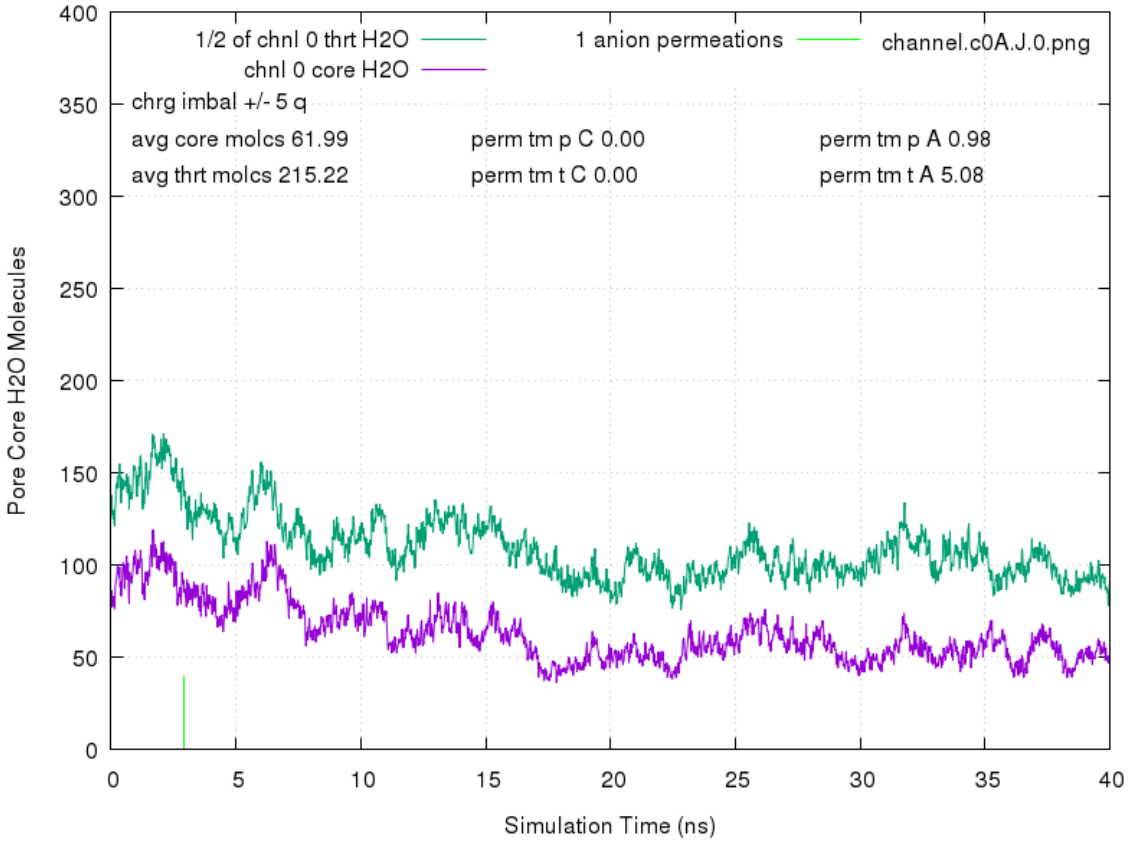

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c00.j.1.png
chrg imbal +/- 5 q
avg core molcs 58.04
avg thrt molcs 209.90
perm tm p C 4.50
perm tm t C 10.68
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

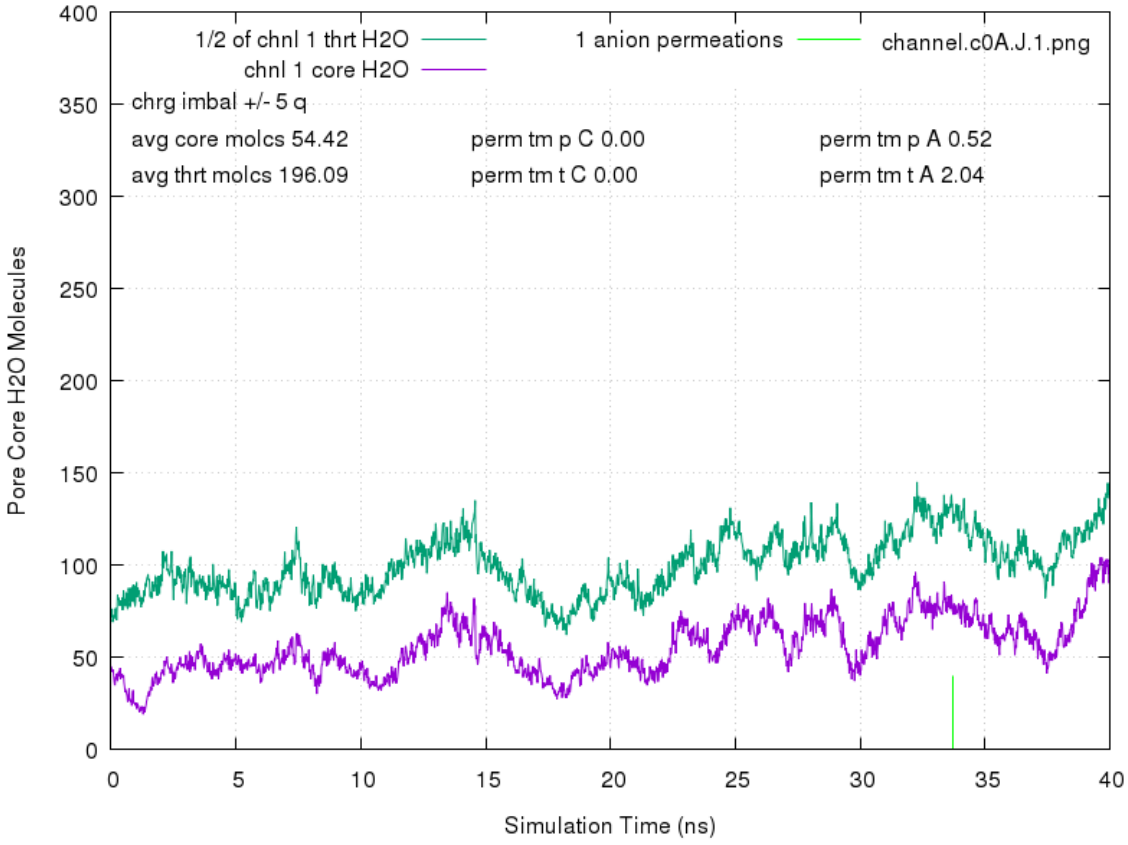

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
1 anion permeations
channel.c00.k.0.png
chrg imbal +/- 5 q
avg core molcs 71.21
avg thrt molcs 234.05
perm tm p C 4.20
perm tm t C 13.34
perm tm p A 1.96
perm tm t A 7.12
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

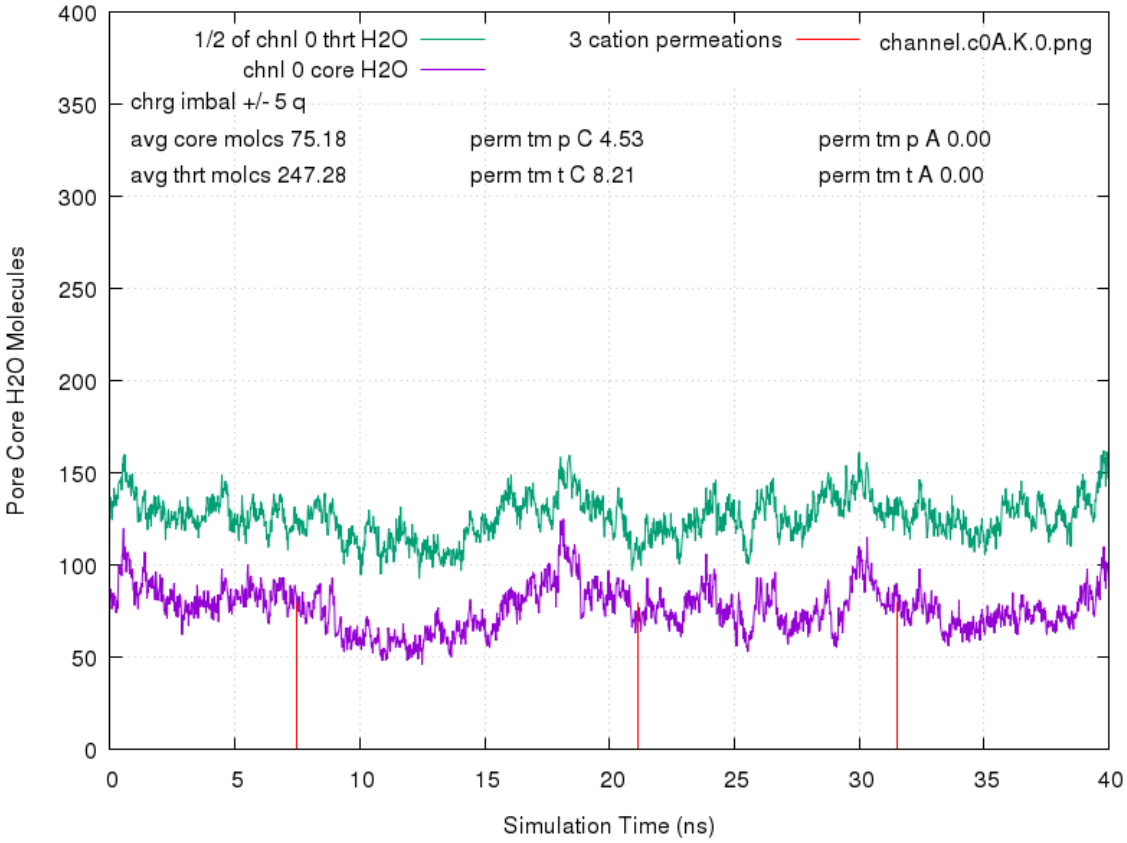

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c00.k.1.png
chrg imbal +/- 5 q
avg core molcs 51.80
avg thrt molcs 184.15
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

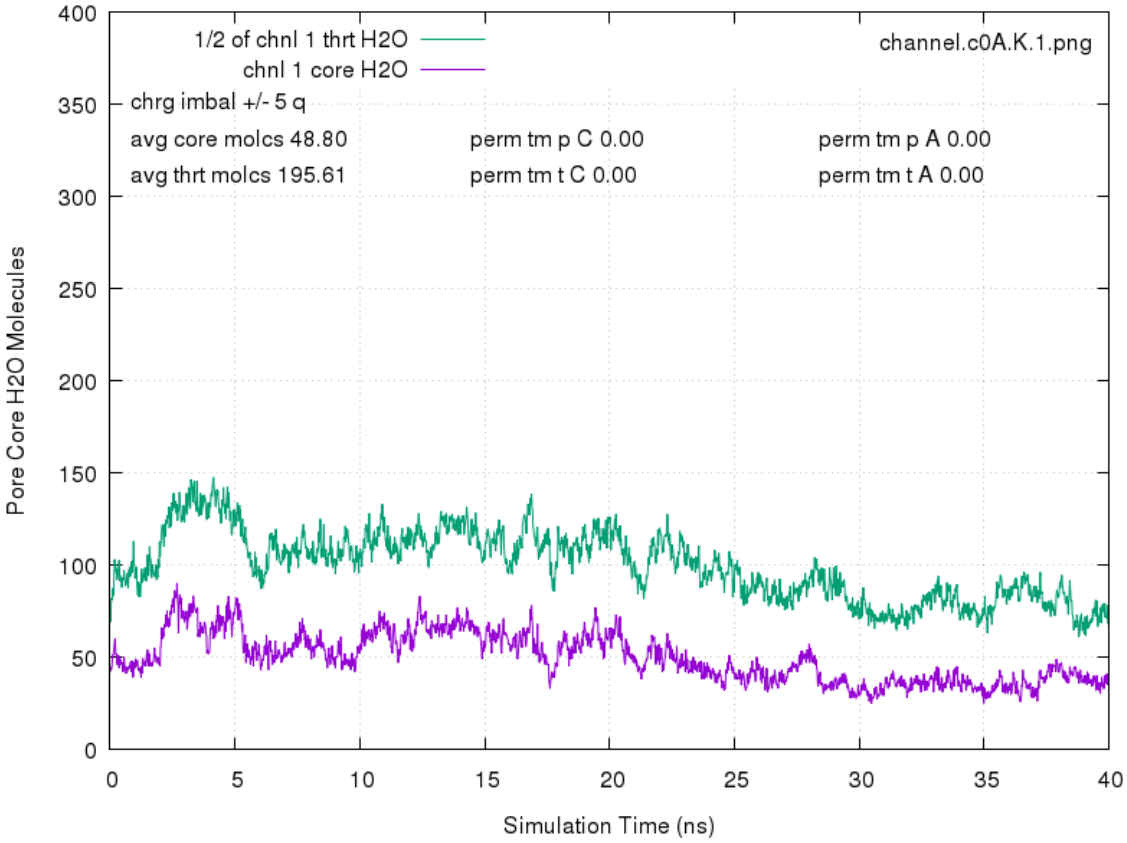

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c00.l.0.png
chrg imbal +/- 5 q
avg core molcs 73.34
avg thrt molcs 248.61
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

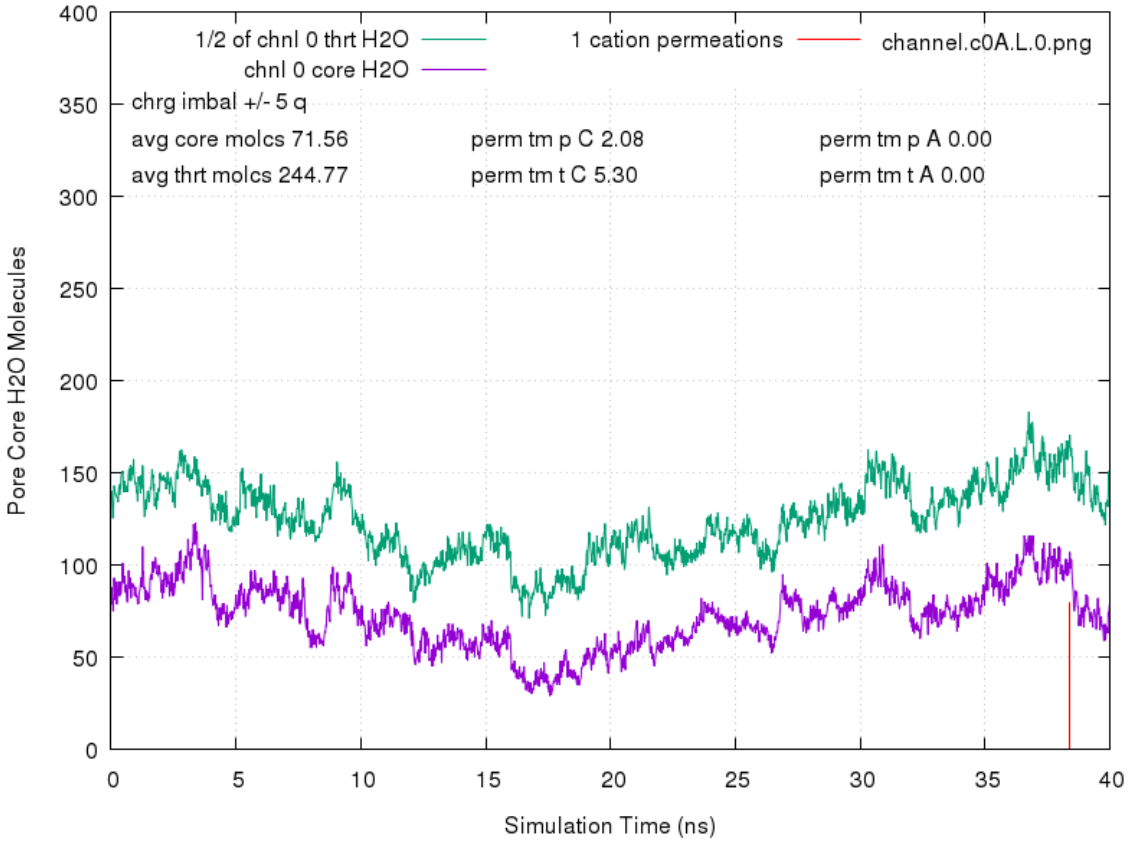

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c00.l.1.png
chrg imbal +/- 5 q
avg core molcs 62.63
avg thrt molcs 211.24
perm tm p C 1.76
perm tm t C 4.04
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

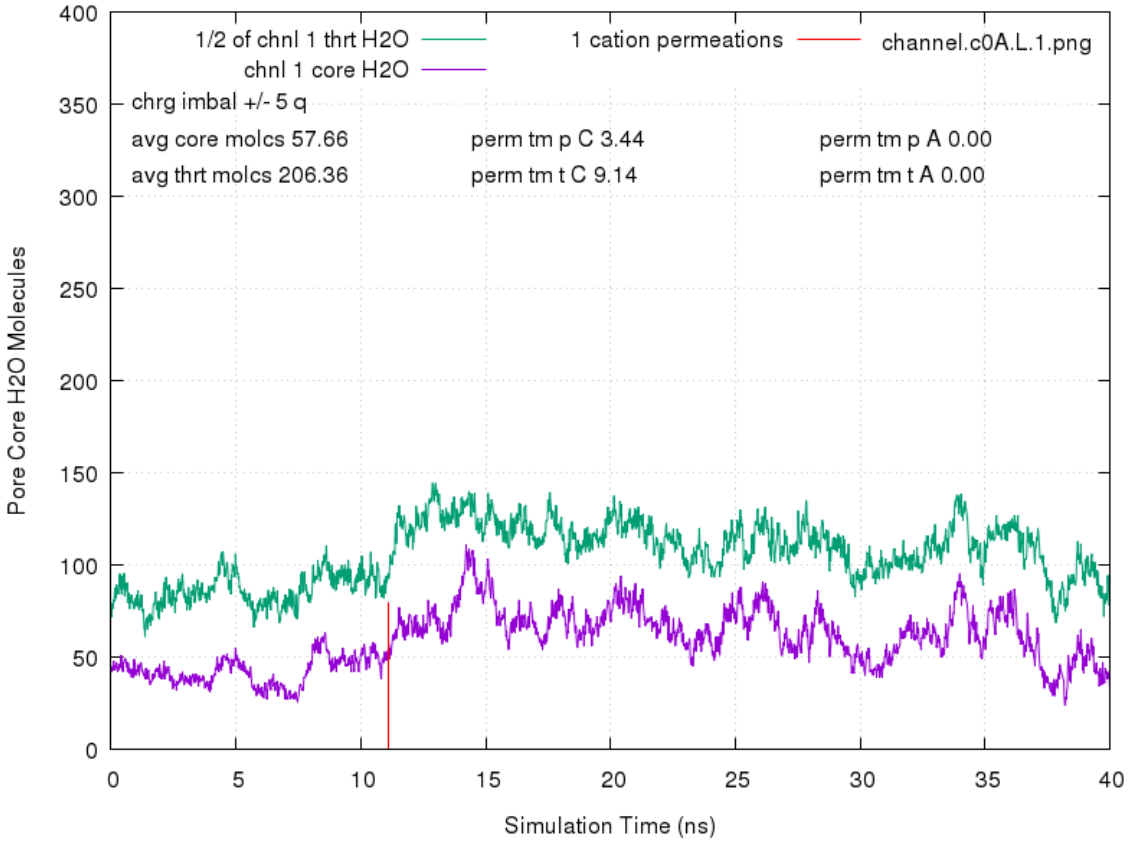

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 anion permeations
channel.c0A.J.0.png
chrg imbal +/- 5 q
avg core molcs 61.99
avg thrt molcs 215.22
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.98
perm tm t A 5.08
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

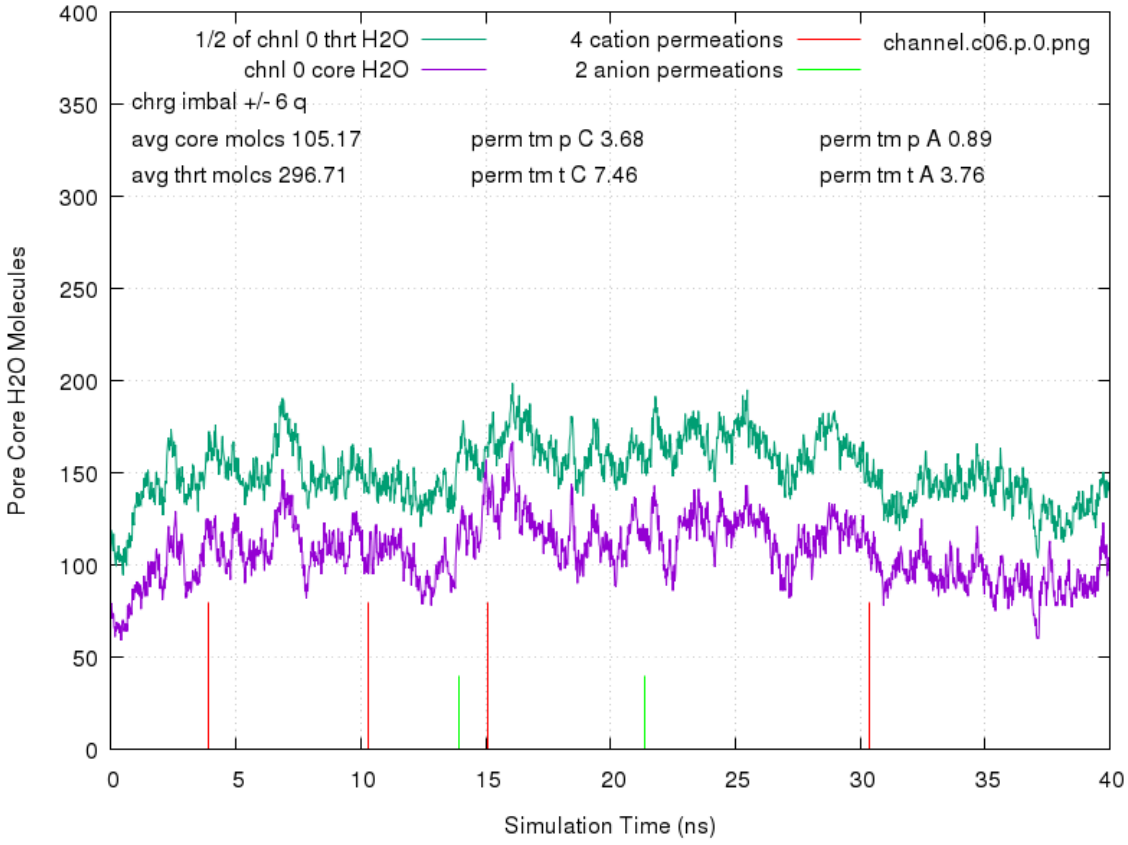

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 anion permeations
channel.c0A.J.1.png
chrg imbal +/- 5 q
avg core molcs 54.42
avg thrt molcs 196.09
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.52
perm tm t A 2.04
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

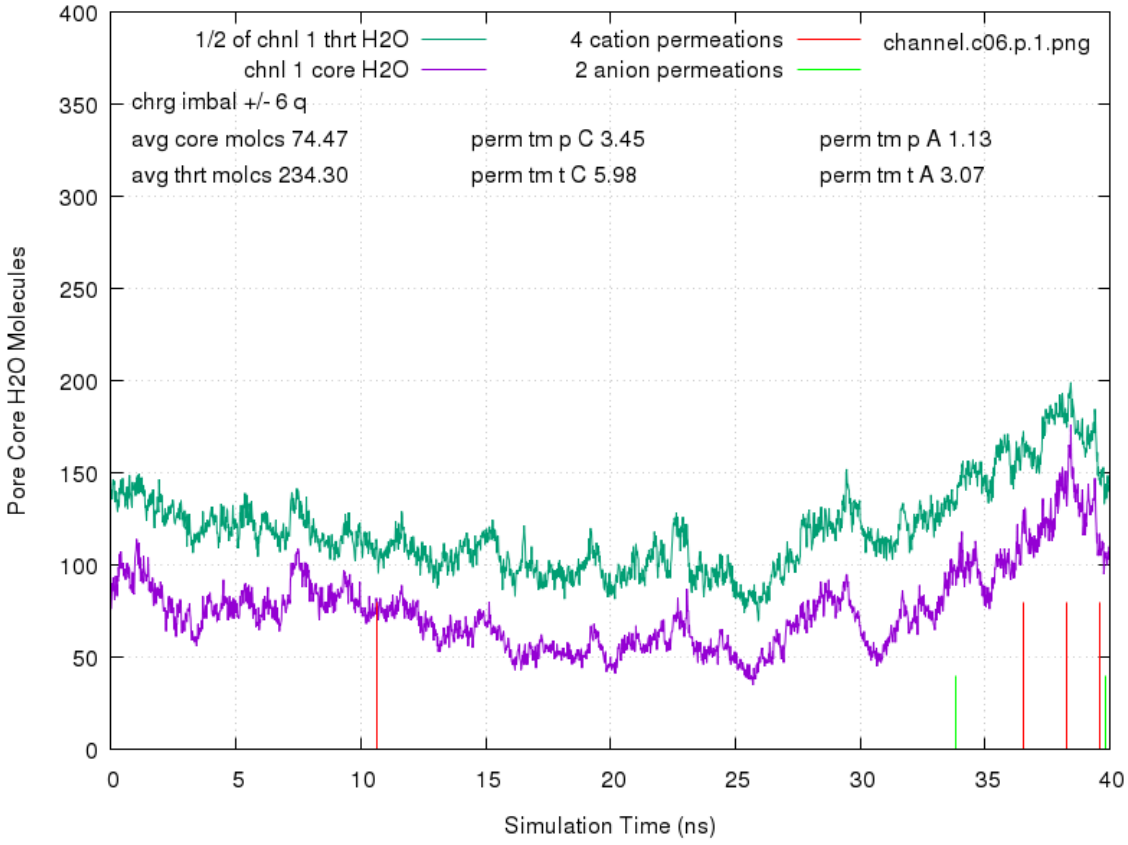

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
channel.c0A.K.0.png
chrg imbal +/- 5 q
avg core molcs 75.18
avg thrt molcs 247.28
perm tm p C 4.53
perm tm t C 8.21
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

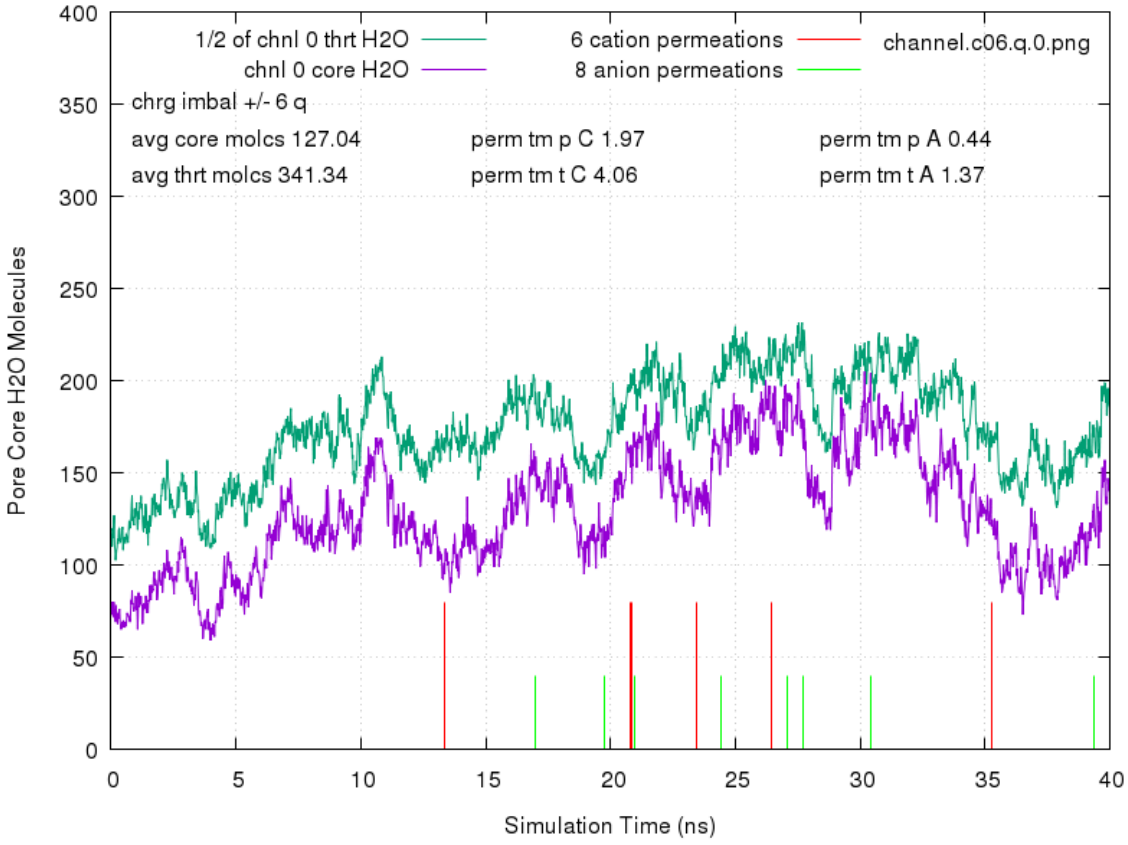

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
channel.c0A.K.1.png
chrg imbal +/- 5 q
avg core molcs 48.80
avg thrt molcs 195.61
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

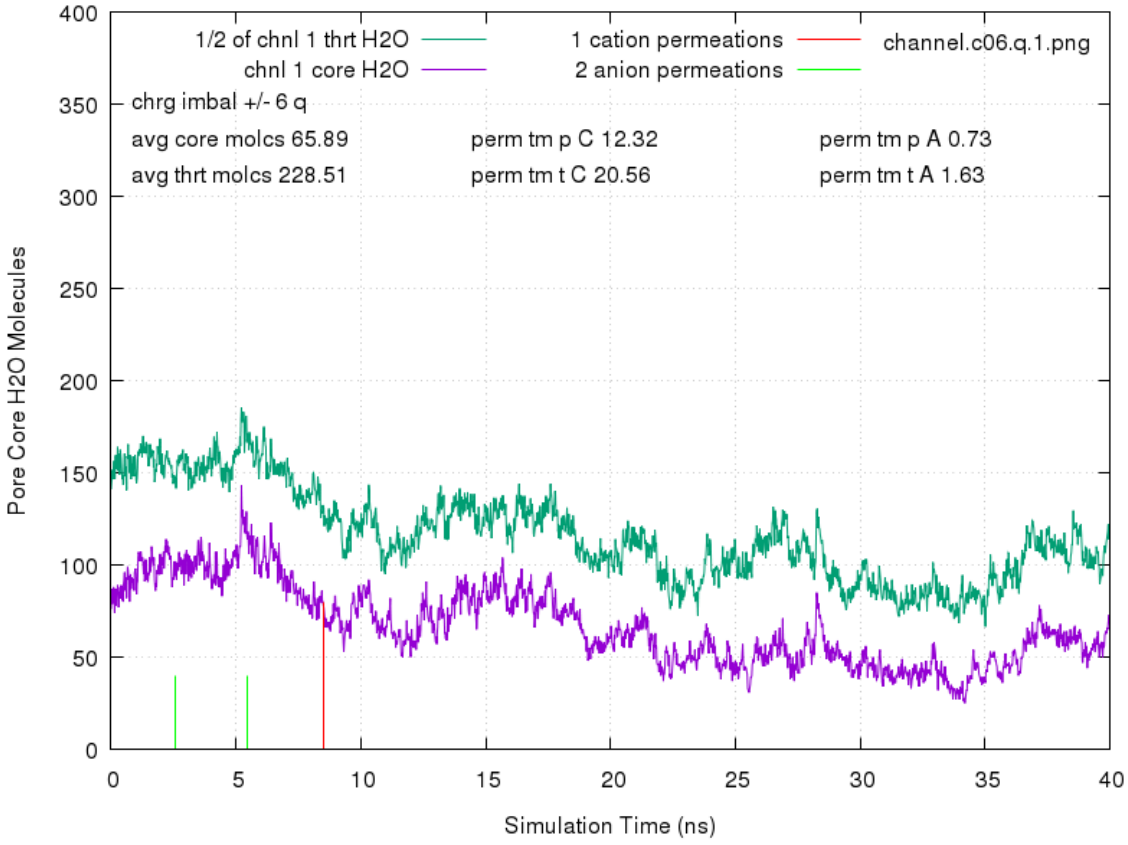

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c0A.L.0.png
chrg imbal +/- 5 q
avg core molcs 71.56
avg thrt molcs 244.77
perm tm p C 2.08
perm tm t C 5.30
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

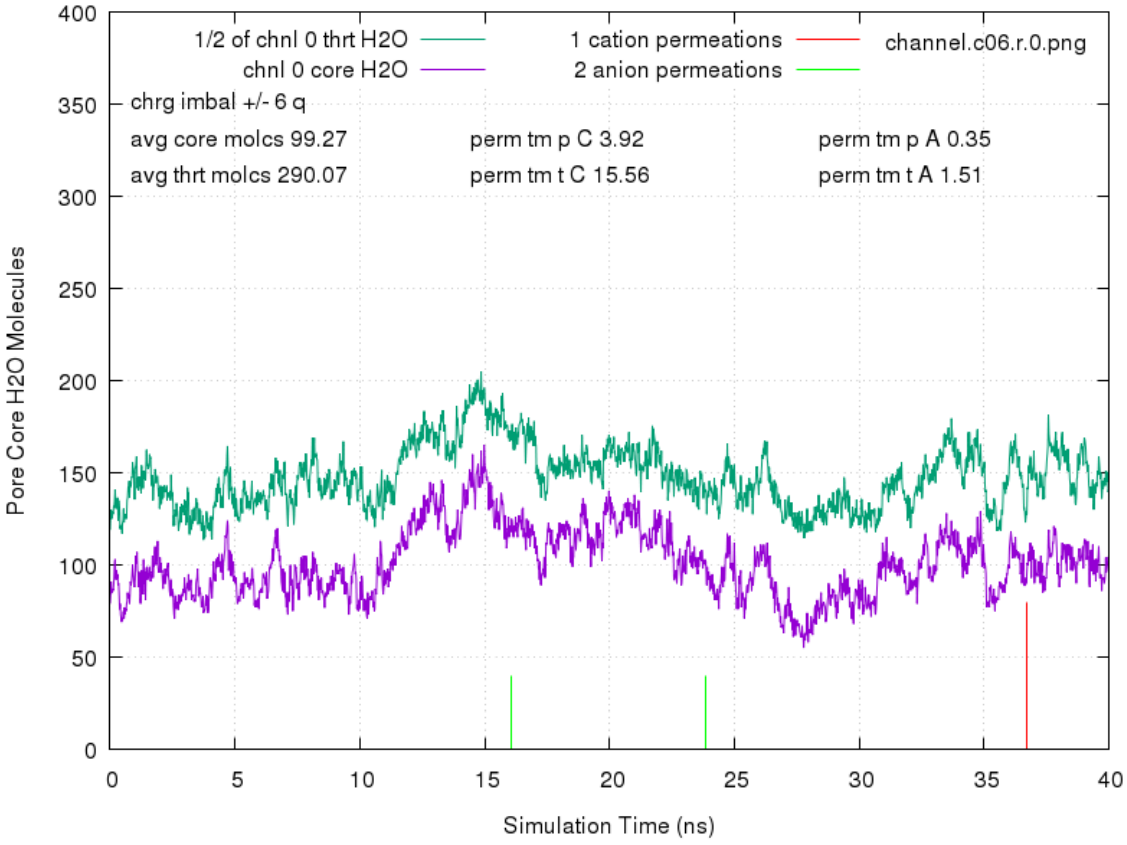

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c0A.L.1.png
chrg imbal +/- 5 q
avg core molcs 57.66
avg thrt molcs 206.36
perm tm p C 3.44
perm tm t C 9.14
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

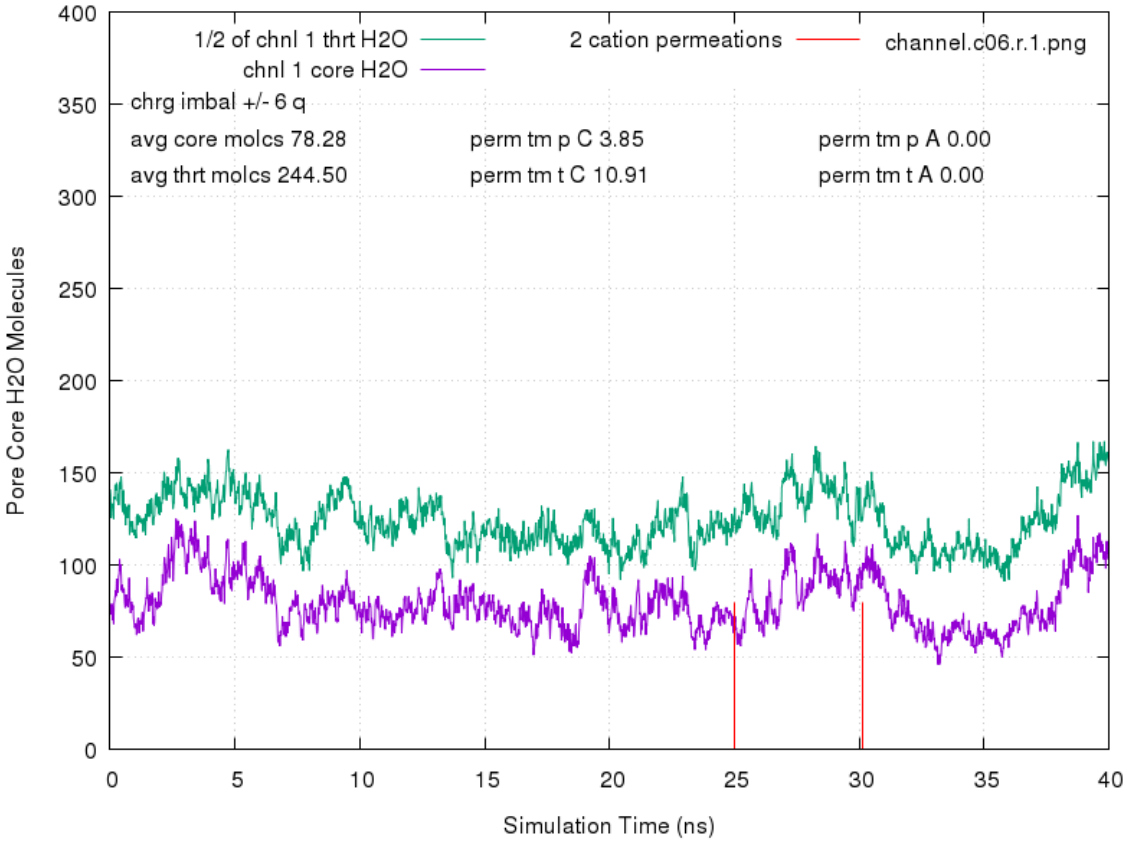

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
4 cation permeations
2 anion permeations
channel.c06.p.0.png
chrg imbal +/- 6 q
avg core molcs 105.17
avg thrt molcs 296.71
perm tm p C 3.68
perm tm t C 7.46
perm tm p A 0.89
perm tm t A 3.76
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

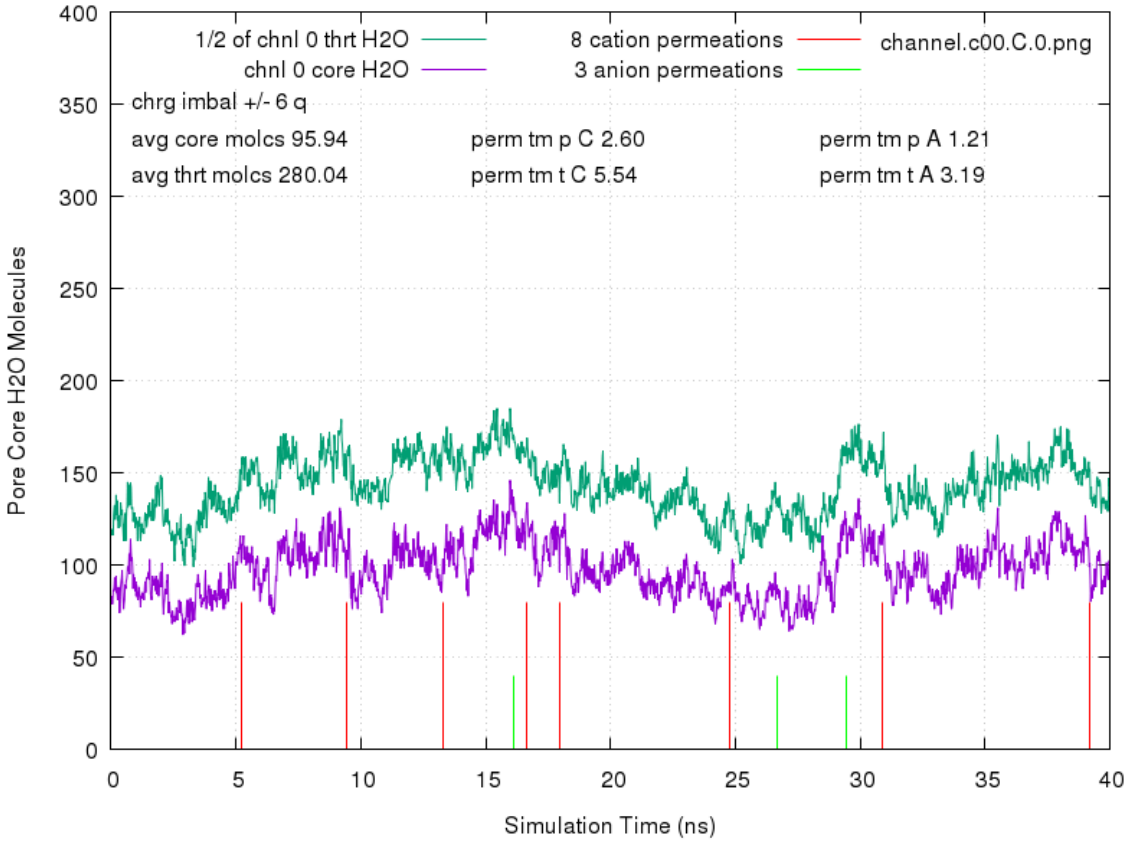

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
2 anion permeations
channel.c06.p.1.png
chrg imbal +/- 6 q
avg core molcs 74.47
avg thrt molcs 234.30
perm tm p C 3.45
perm tm t C 5.98
perm tm p A 1.13
perm tm t A 3.07
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

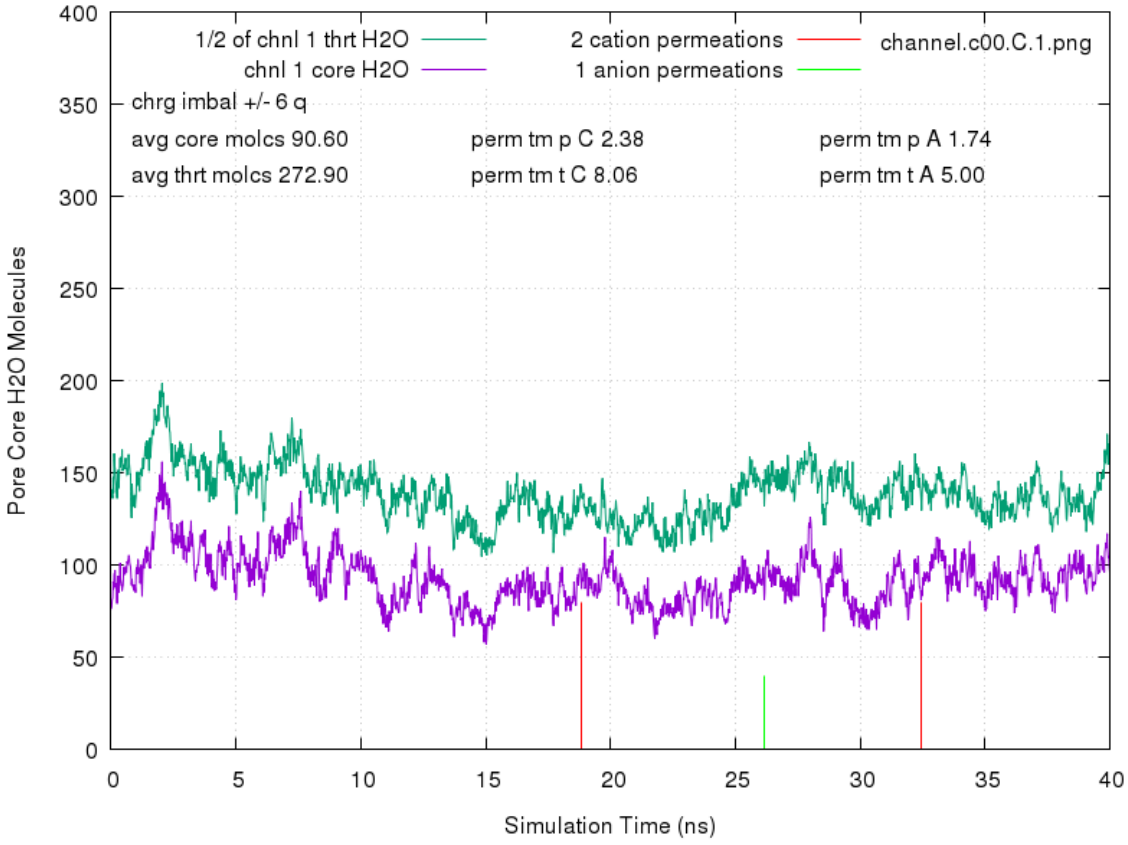

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
6 cation permeations
8 anion permeations
channel.c06.q.0.png
chrg imbal +/- 6 q
avg core molcs 127.04
avg thrt molcs 341.34
perm tm p C 1.97
perm tm t C 4.06
perm tm p A 0.44
perm tm t A 1.37
Pore Core H2O Molecules
Simulation Time (ns)

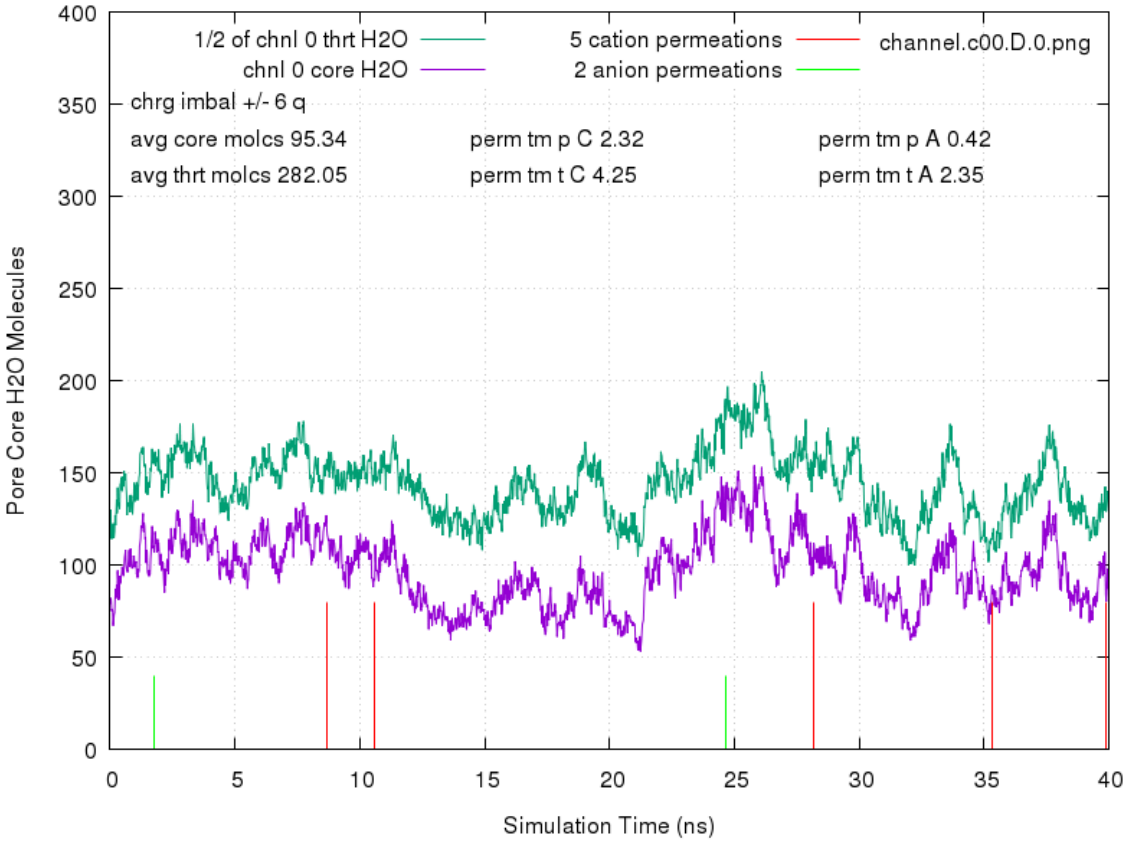

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
2 anion permeations
channel.c06.q.1.png
chrg imbal +/- 6 q
avg core molcs 65.89
avg thrt molcs 228.51
perm tm p C 12.32
perm tm t C 20.56
perm tm p A 0.73
perm tm t A 1.63
Pore Core H2O Molecules
Simulation Time (ns)

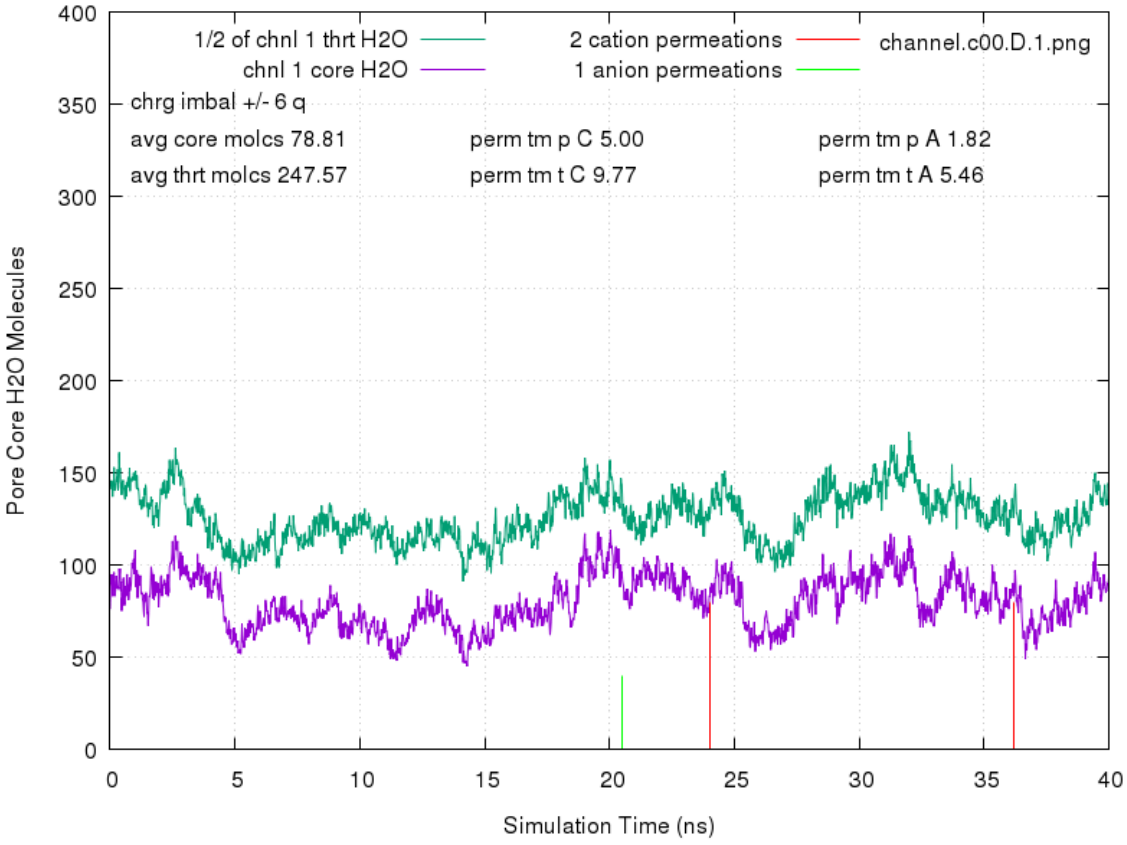

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
2 anion permeations
channel.c06.r.0.png
chrg imbal +/- 6 q
avg core molcs 99.27
avg thrt molcs 290.07
perm tm p C 3.92
perm tm t C 15.56
perm tm p A 0.35
perm tm t A 1.51
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

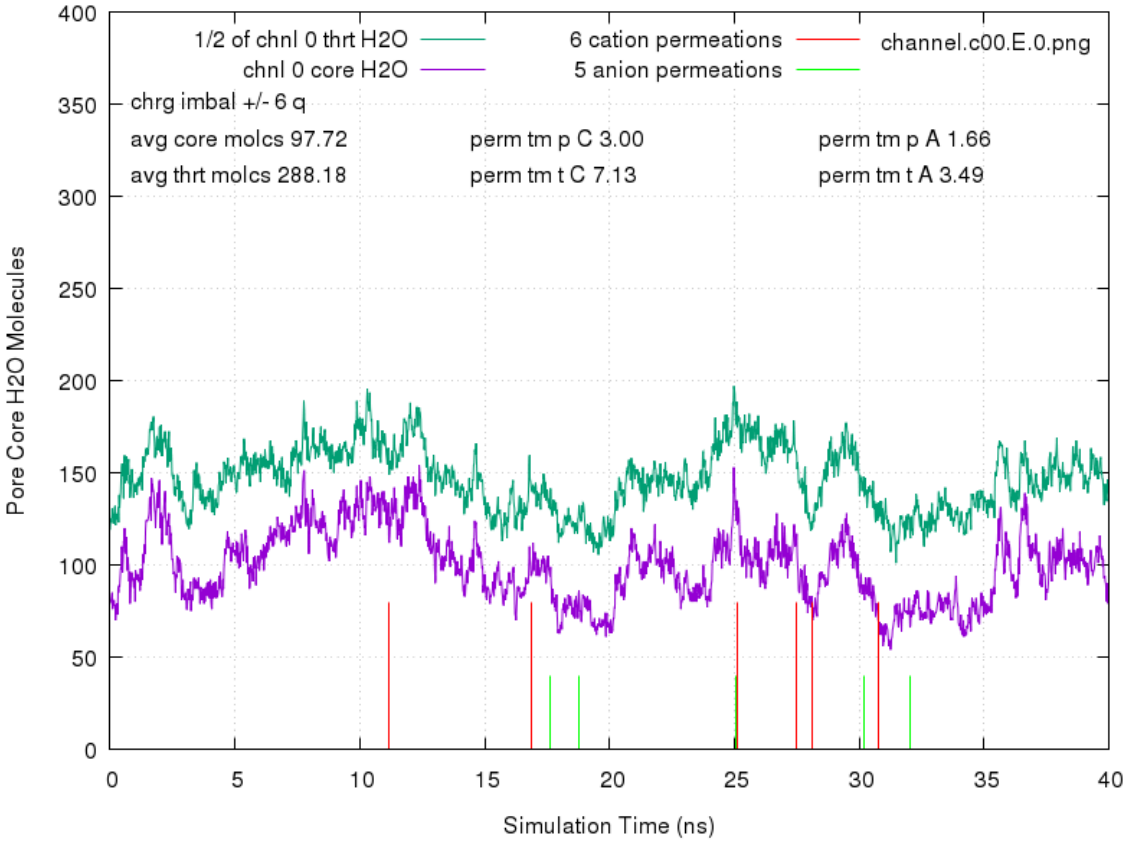

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
channel.c06.r.1.png
chrg imbal +/- 6 q
avg core molcs 78.28
avg thrt molcs 244.50
perm tm p C 3.85
perm tm t C 10.91
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

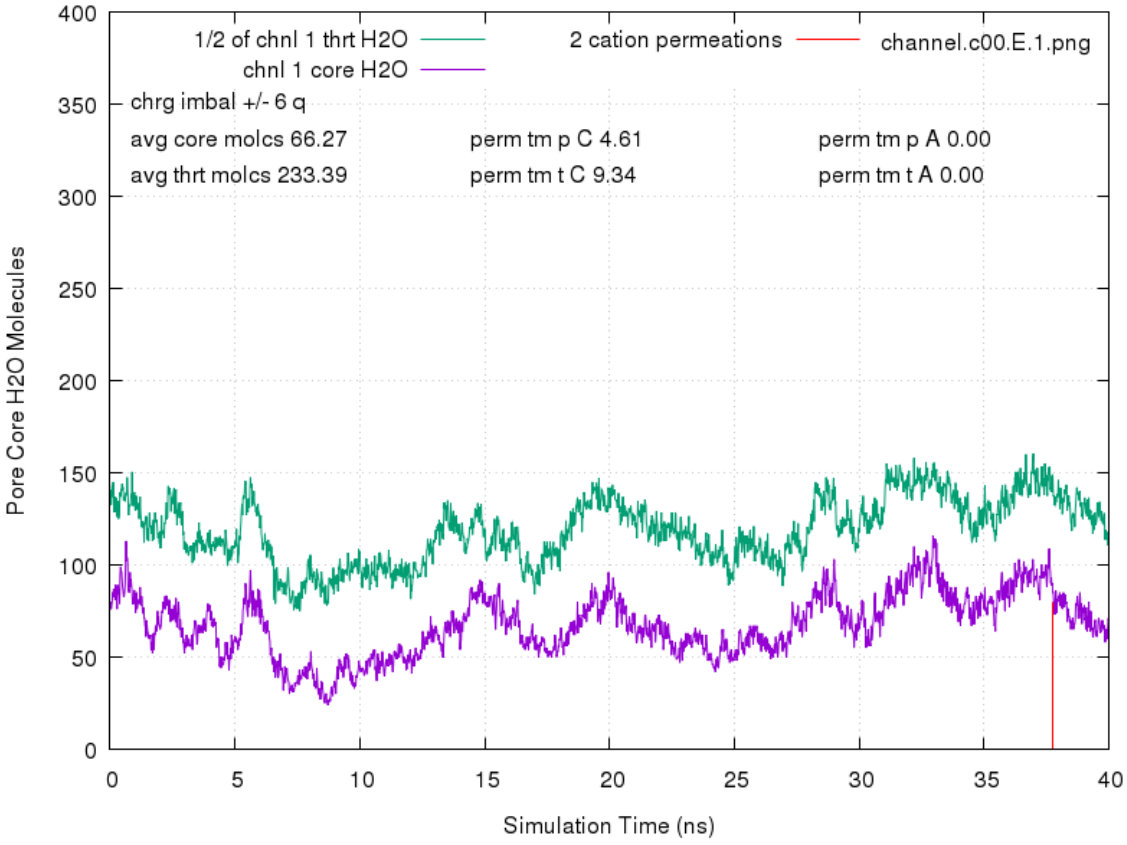

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
3 anion permeations
channel.c00.C.0.png
chrg imbal +/- 6 q
avg core molcs 95.94
avg thrt molcs 280.04
perm tm p C 2.60
perm tm t C 5.54
perm tm p A 1.21
perm tm t A 3.19
Pore Core H2O Molecules
Simulation Time (ns)

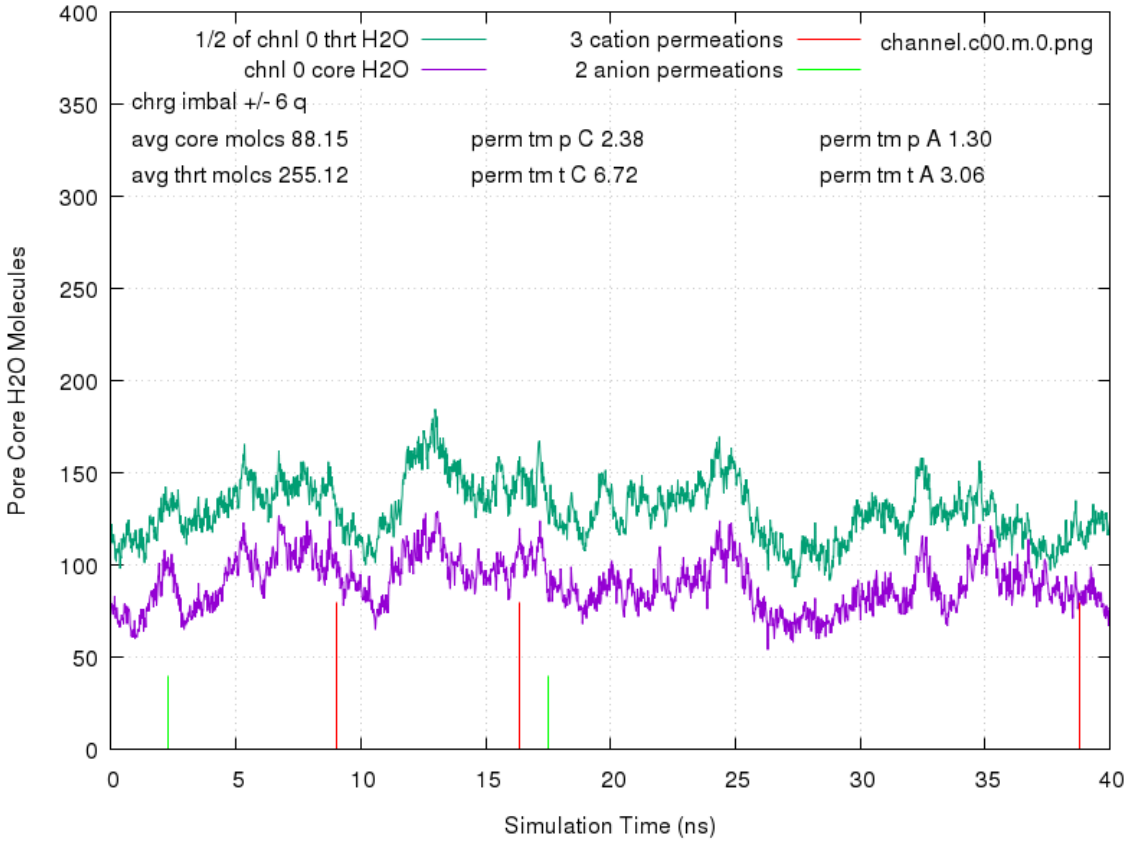

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
1 anion permeations
channel.c00.C.1.png
chrg imbal +/- 6 q
avg core molcs 90.60
avg thrt molcs 272.90
perm tm p C 2.38
perm tm t C 8.06
perm tm p A 1.74
perm tm t A 5.00
Pore Core H2O Molecules
Simulation Time (ns)

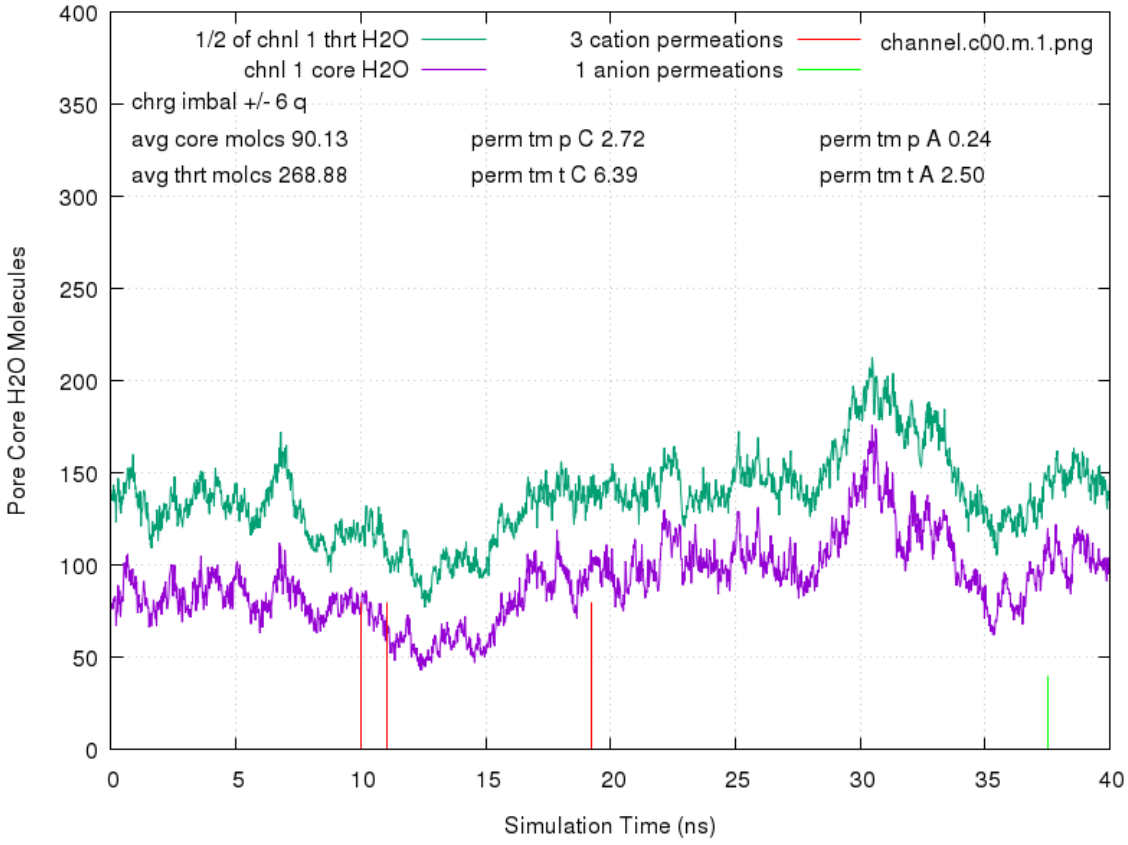

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
2 anion permeations
channel.c00.D.0.png
chrg imbal +/- 6 q
avg core molcs 95.34
avg thrt molcs 282.05
perm tm p C 2.32
perm tm t C 4.25
perm tm p A 0.42
perm tm t A 2.35
Pore Core H2O Molecules
Simulation Time (ns)

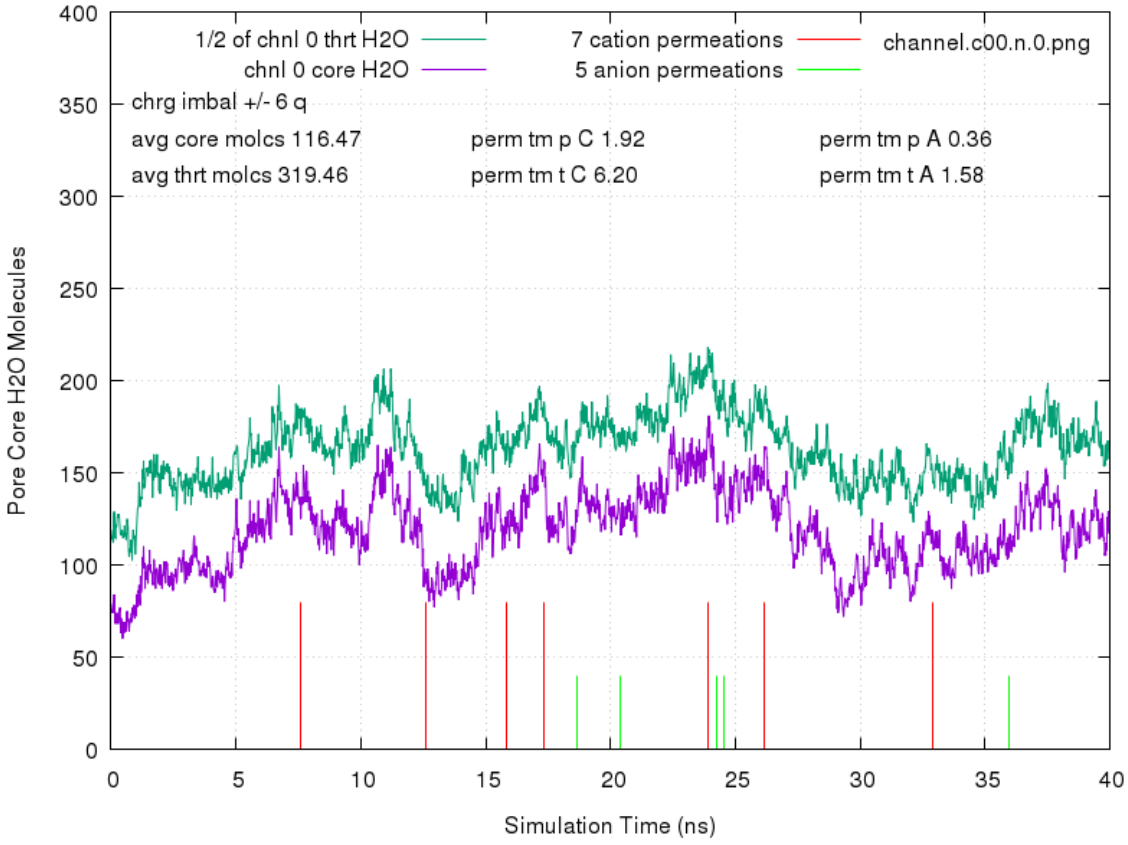

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
1 anion permeations
channel.c00.D.1.png
chrg imbal +/- 6 q
avg core molcs 78.81
avg thrt molcs 247.57
perm tm p C 5.00
perm tm t C 9.77
perm tm p A 1.82
perm tm t A 5.46
Pore Core H2O Molecules
Simulation Time (ns)

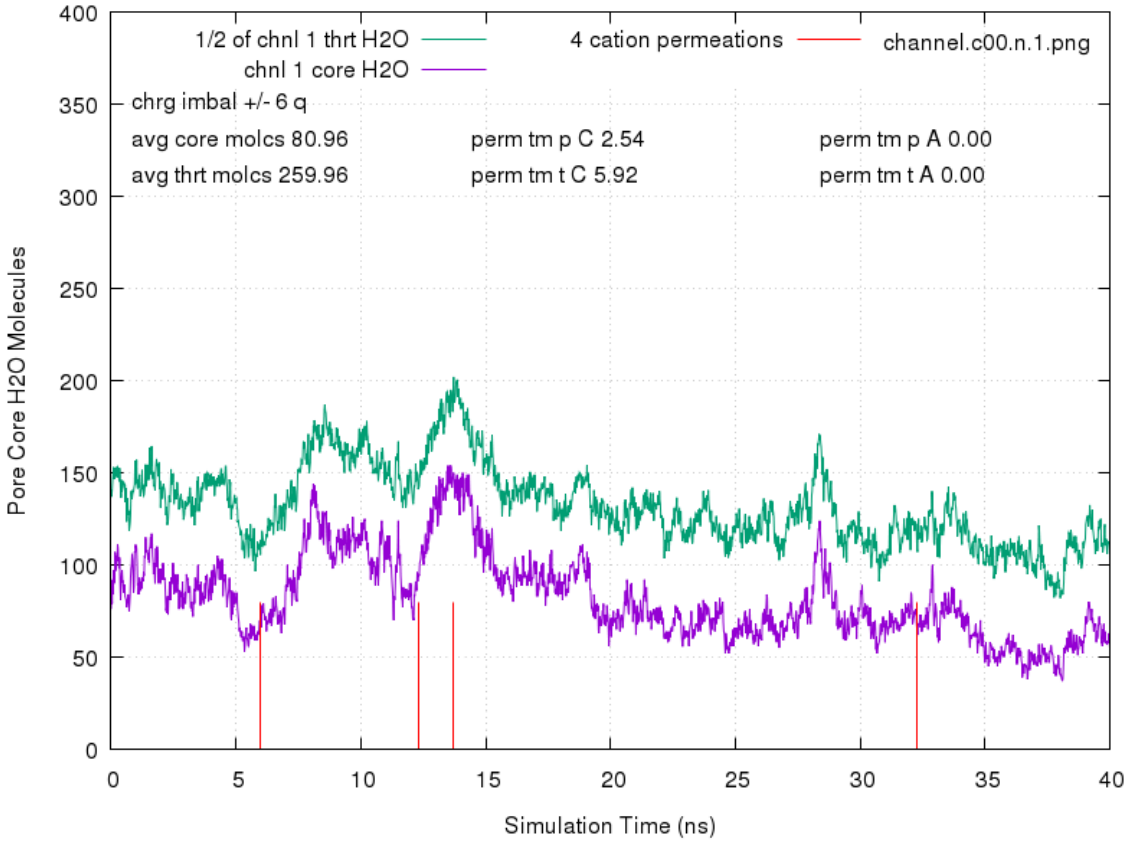

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
6 cation permeations
5 anion permeations
channel.c00.E.0.png
chrg imbal +/- 6 q
avg core molcs 97.72
avg thrt molcs 288.18
perm tm p C 3.00
perm tm t C 7.13
perm tm p A 1.66
perm tm t A 3.49
Pore Core H2O Molecules
Simulation Time (ns)

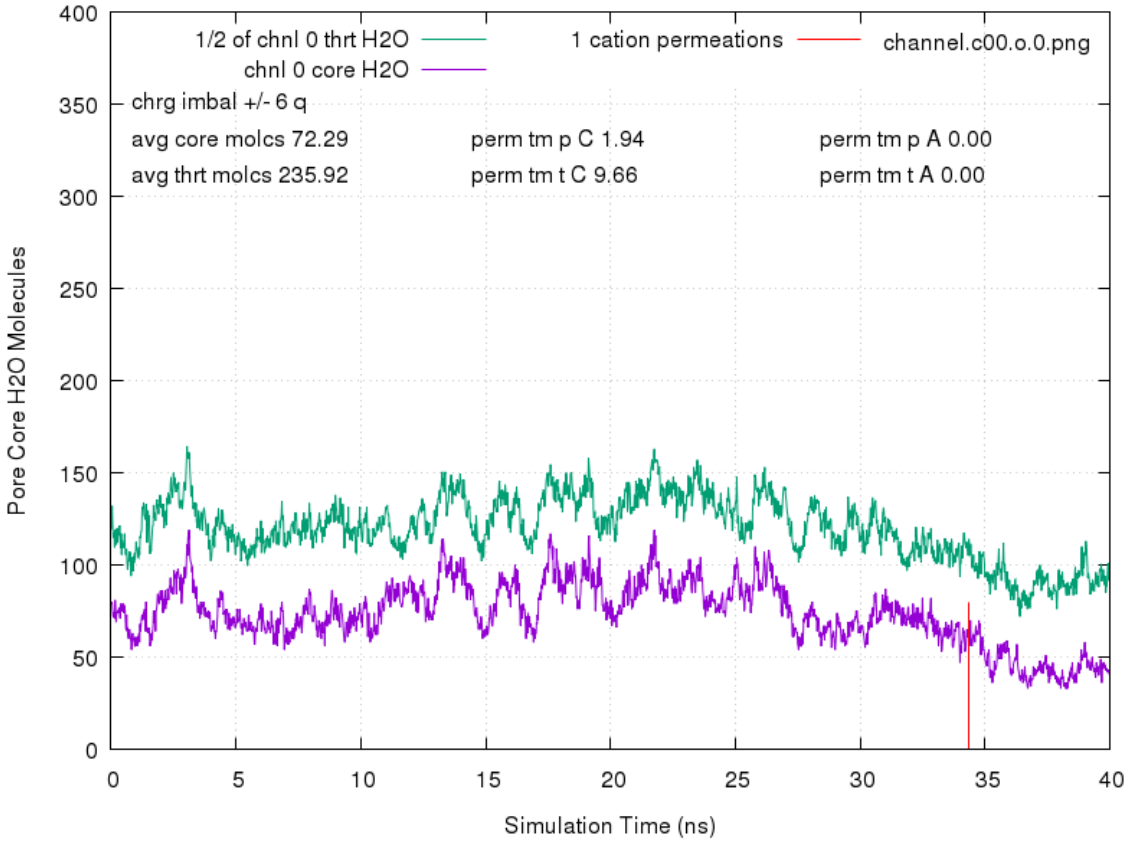

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
channel.c00.E.1.png
chrg imbal +/- 6 q
avg core molcs 66.27
avg thrt molcs 233.39
perm tm p C 4.61
perm tm t C 9.34
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

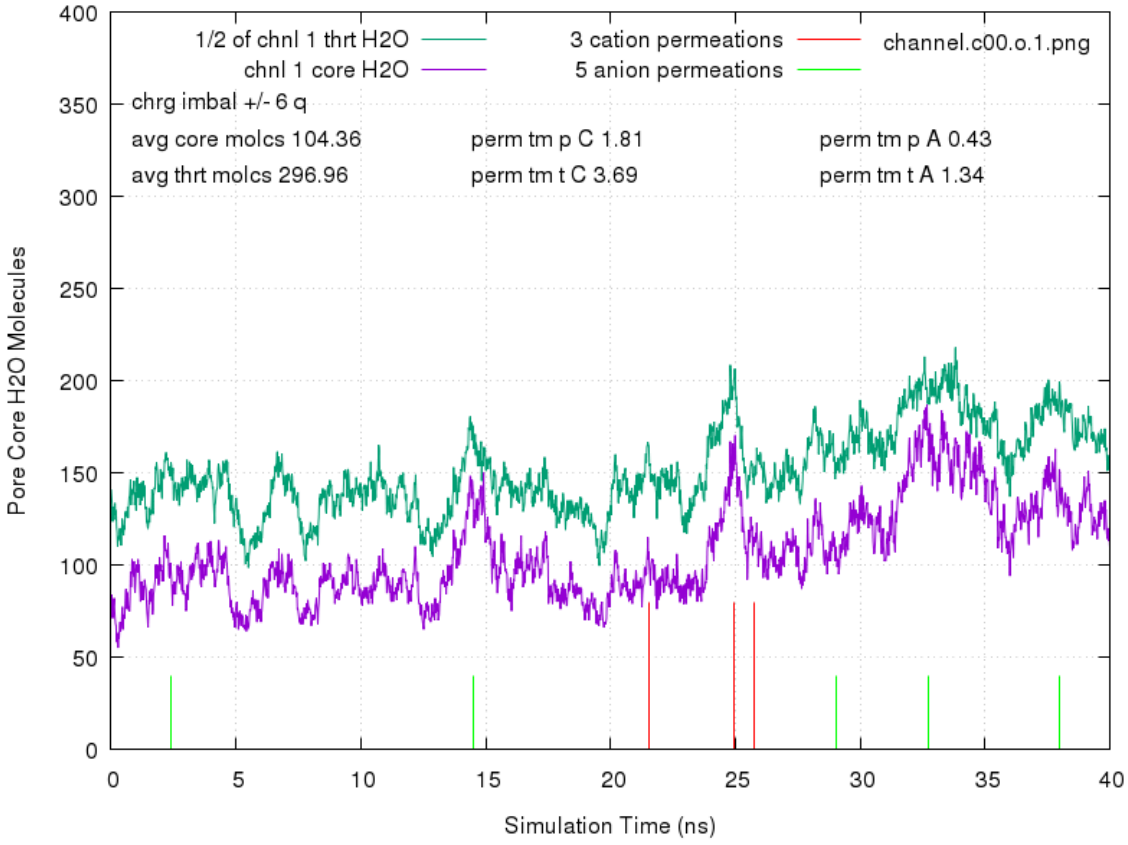

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
2 anion permeations
channel.c00.m.0.png
chrg imbal +/- 6 q
avg core molcs 88.15
avg thrt molcs 255.12
perm tm p C 2.38
perm tm t C 6.72
perm tm p A 1.30
perm tm t A 3.06
Pore Core H2O Molecules
Simulation Time (ns)

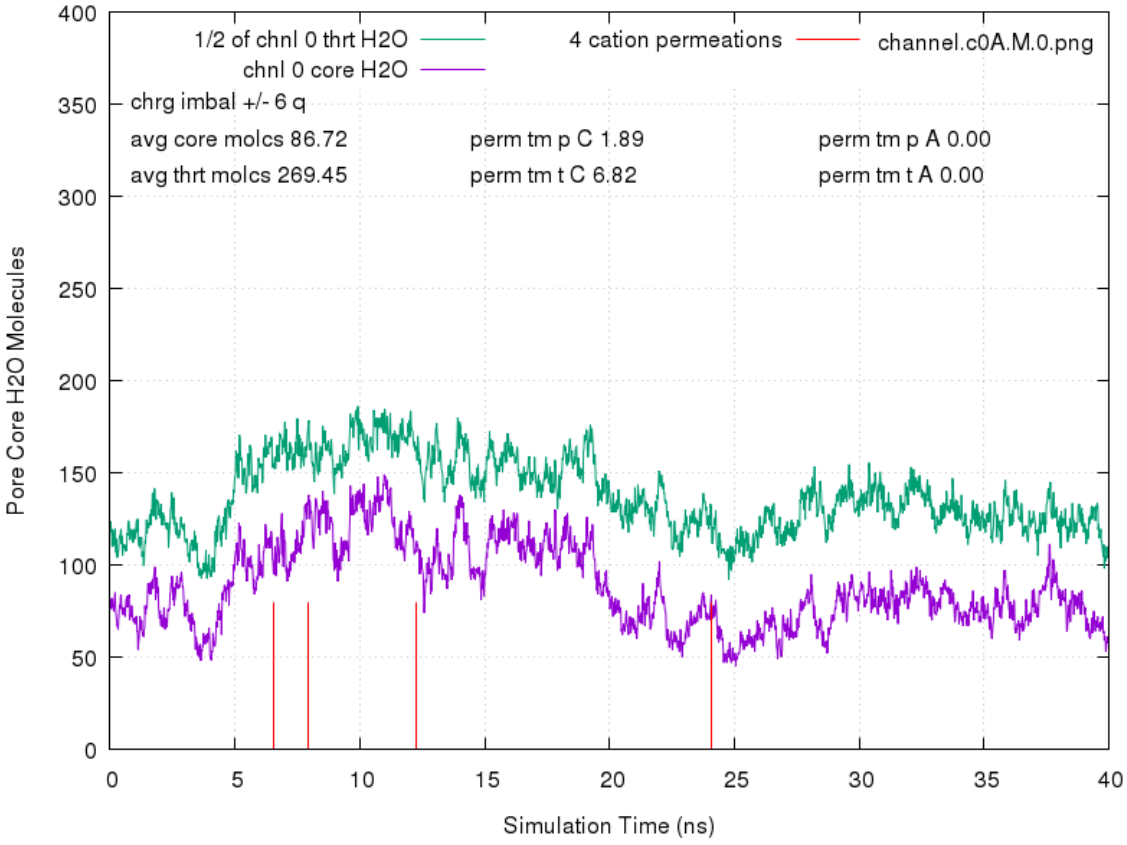

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
1 anion permeations
channel.c00.m.1.png
chrg imbal +/- 6 q
avg core molcs 90.13
avg thrt molcs 268.88
perm tm p C 2.72
perm tm t C 6.39
perm tm p A 0.24
perm tm t A 2.50
Pore Core H2O Molecules
Simulation Time (ns)

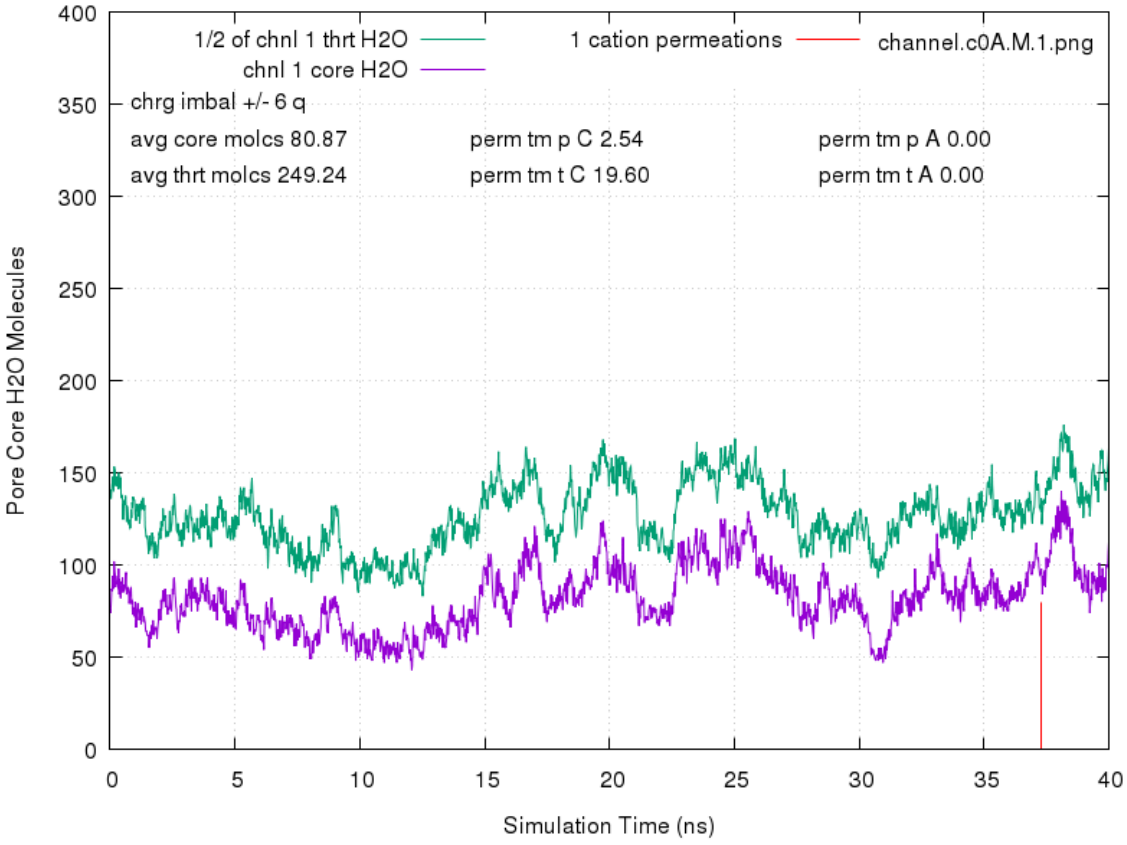

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
7 cation permeations
5 anion permeations
channel.c00.n.0.png
chrg imbal +/- 6 q
avg core molcs 116.47
avg thrt molcs 319.46
perm tm p C 1.92
perm tm t C 6.20
perm tm p A 0.36
perm tm t A 1.58
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

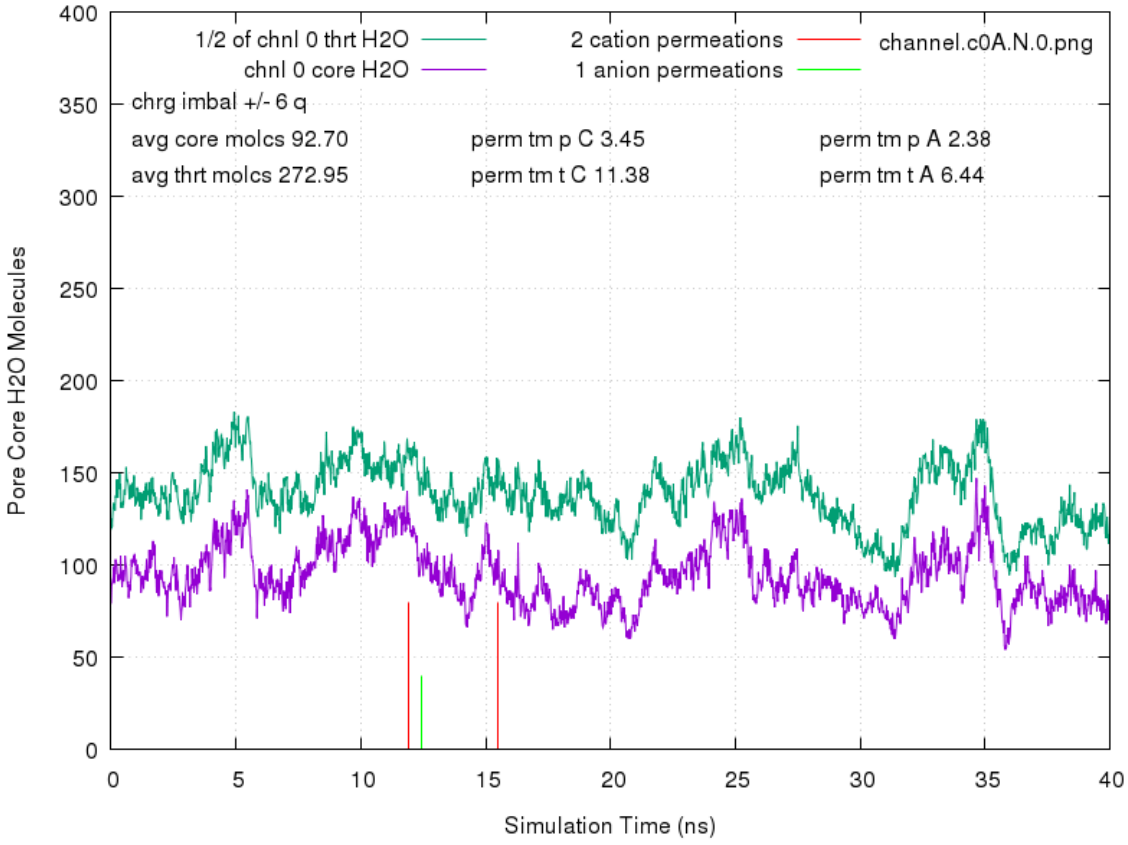

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
channel.c00.n.1.png
chrg imbal +/- 6 q
avg core molcs 80.96
avg thrt molcs 259.96
perm tm p C 2.54
perm tm t C 5.92
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

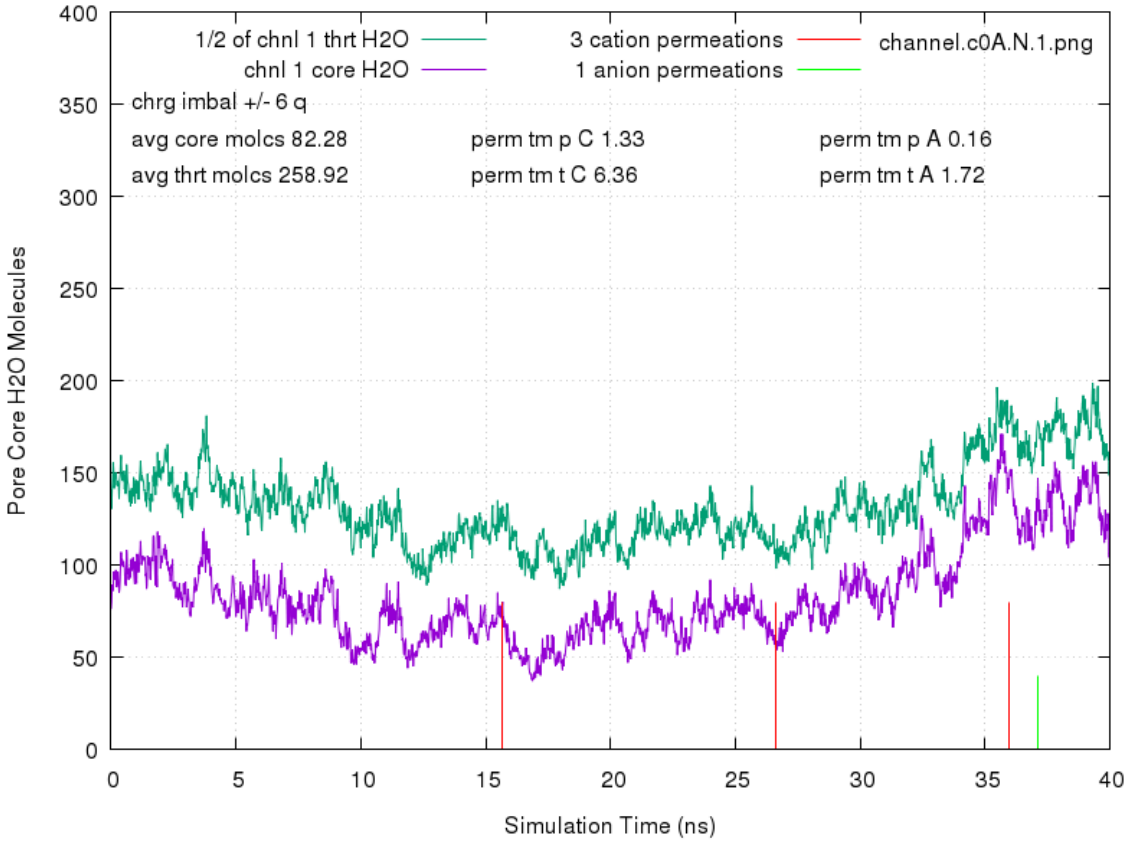

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c00.o.0.png
chrg imbal +/- 6 q
avg core molcs 72.29
avg thrt molcs 235.92
perm tm p C 1.94
perm tm t C 9.66
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

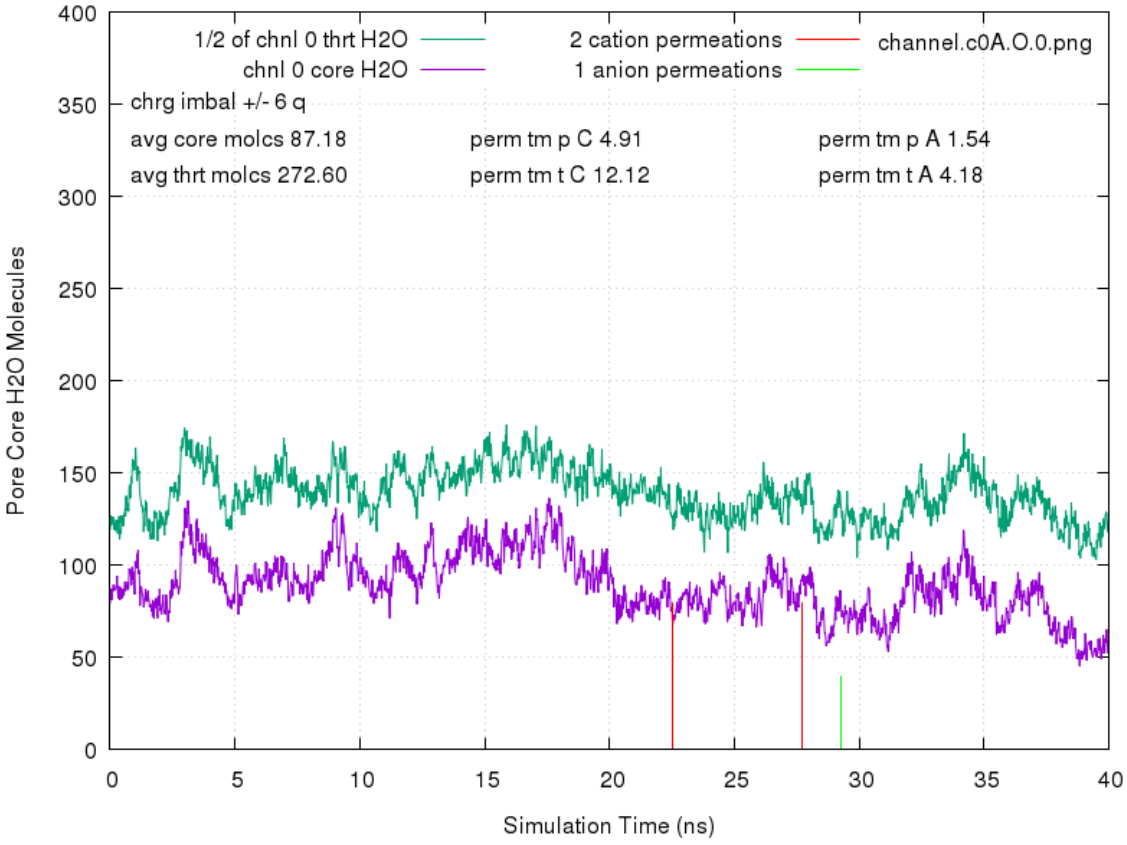

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
5 anion permeations
channel.c00.o.1.png
chrg imbal +/- 6 q
avg core molcs 104.36
avg thrt molcs 296.96
perm tm p C 1.81
perm tm t C 3.69
perm tm p A 0.43
perm tm t A 1.34
Pore Core H2O Molecules
Simulation Time (ns)

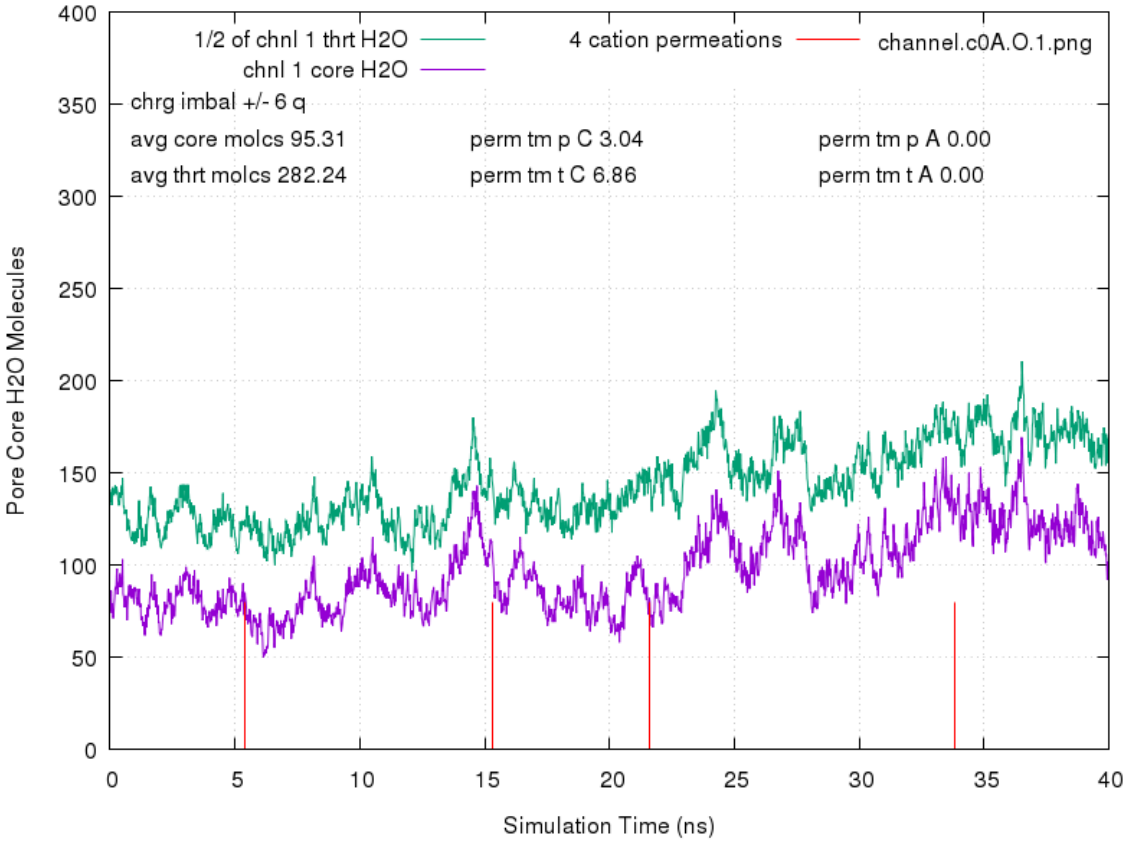

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
4 cation permeations
channel.c0A.M.0.png
chrg imbal +/- 6 q
avg core molcs 86.72
avg thrt molcs 269.45
perm tm p C 1.89
perm tm t C 6.82
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

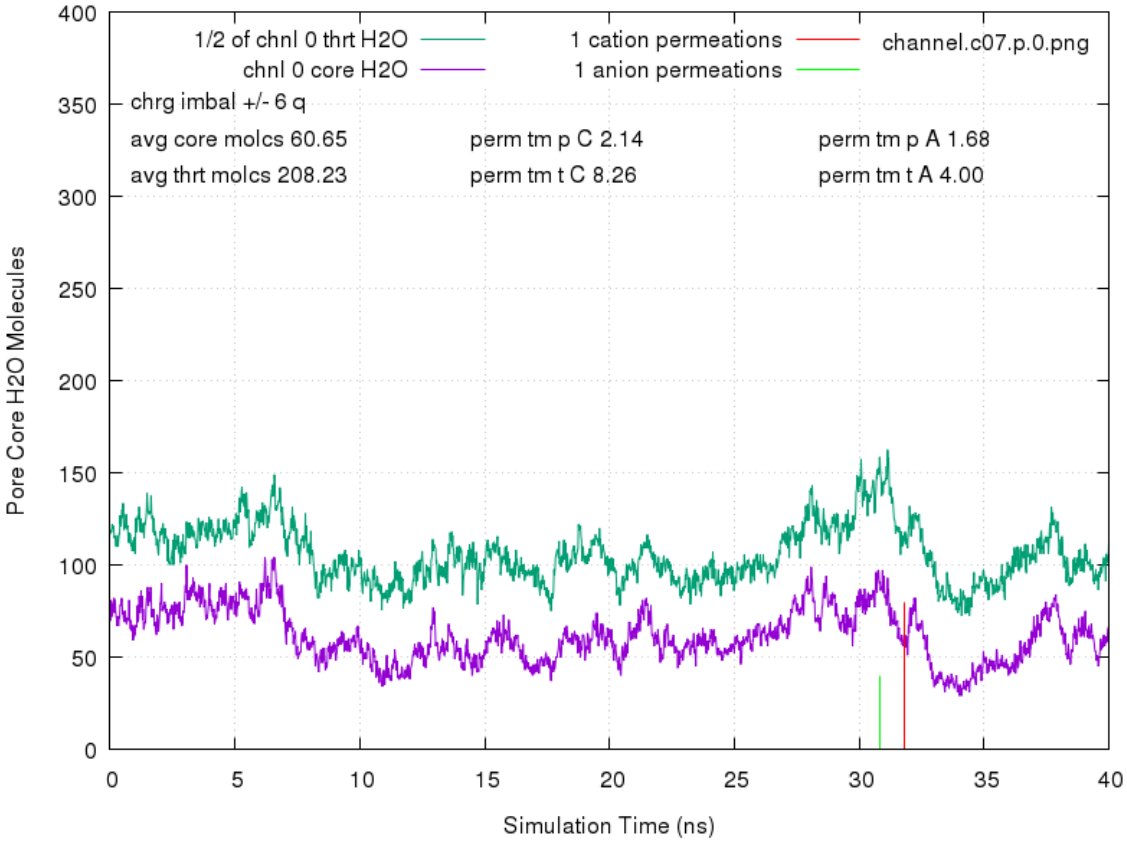

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
channel.c0A.M.1.png
chrg imbal +/- 6 q
avg core molcs 80.87
avg thrt molcs 249.24
perm tm p C 2.54
perm tm t C 19.60
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

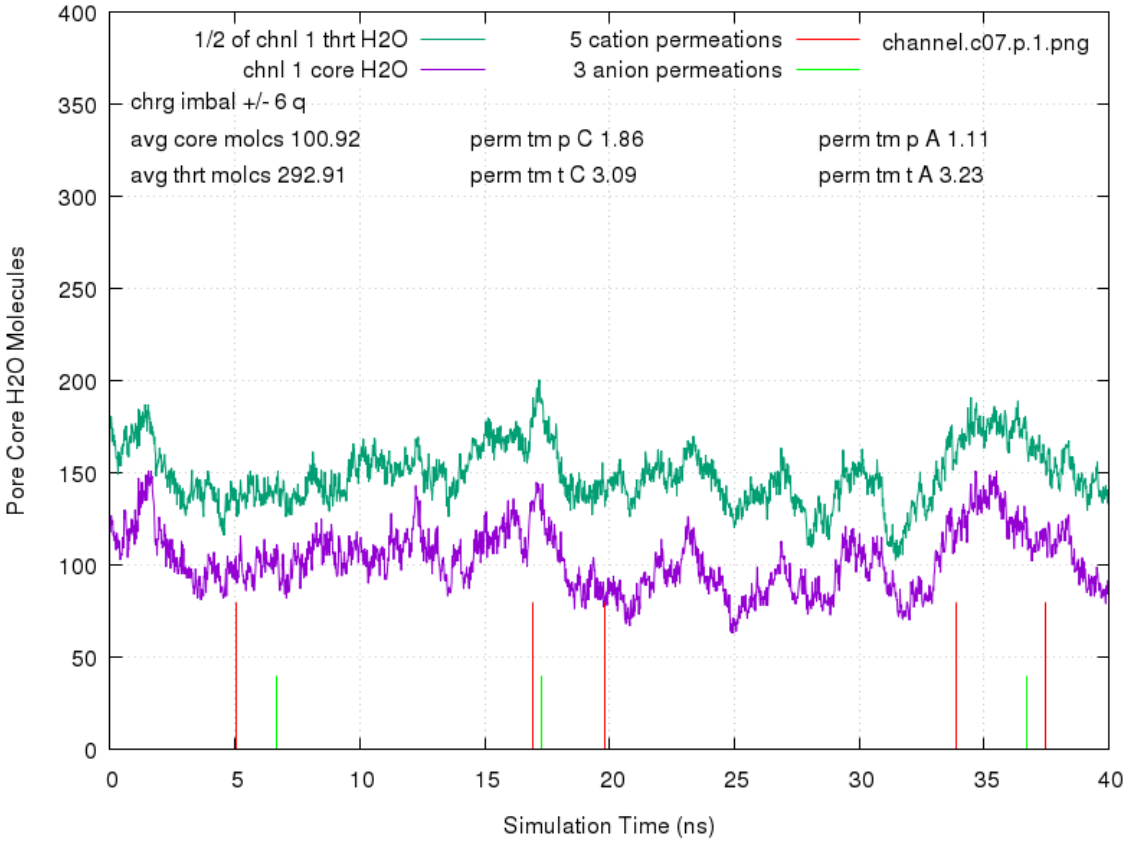

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
1 anion permeations
channel.c0A.N.0.png
chrg imbal +/- 6 q
avg core molcs 92.70
avg thrt molcs 272.95
perm tm p C 3.45
perm tm t C 11.38
perm tm p A 2.38
perm tm t A 6.44
Pore Core H2O Molecules
Simulation Time (ns)

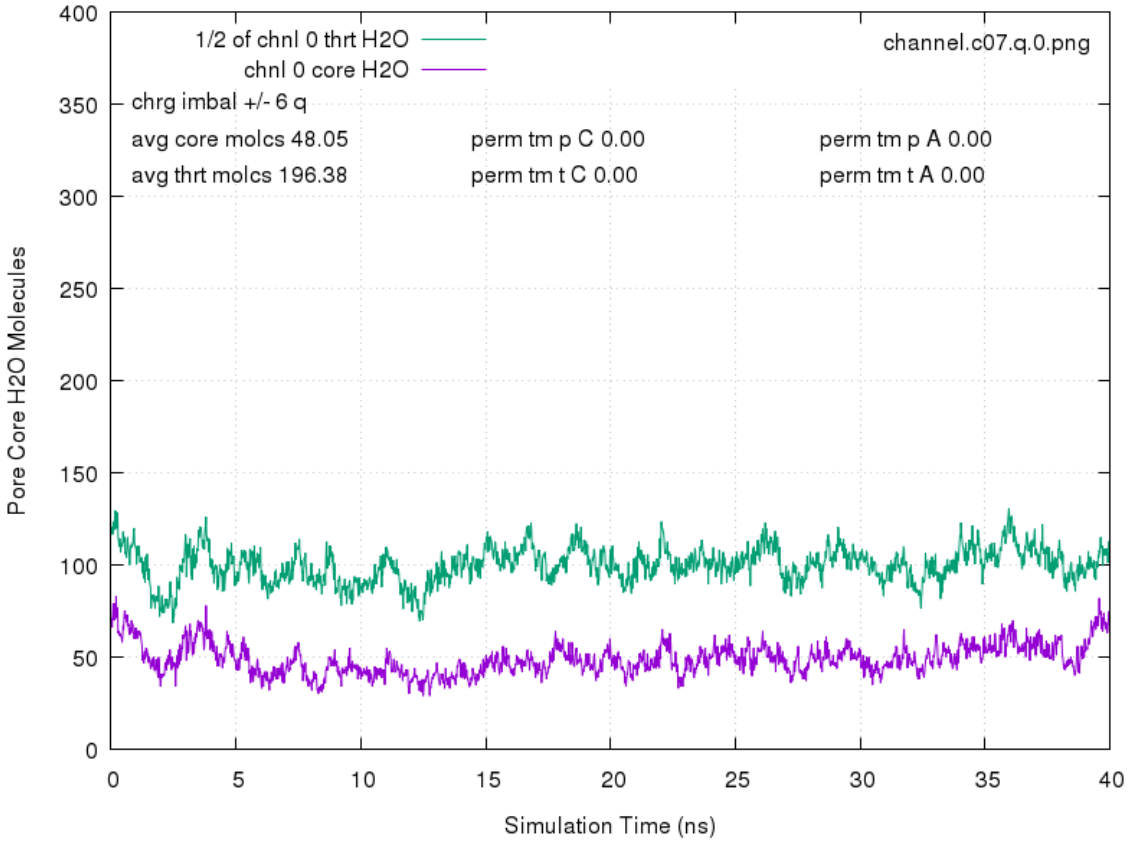

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
1 anion permeations
channel.c0A.N.1.png
chrg imbal +/- 6 q
avg core molcs 82.28
avg thrt molcs 258.92
perm tm p C 1.33
perm tm t C 6.36
perm tm p A 0.16
perm tm t A 1.72
Pore Core H2O Molecules
Simulation Time (ns)

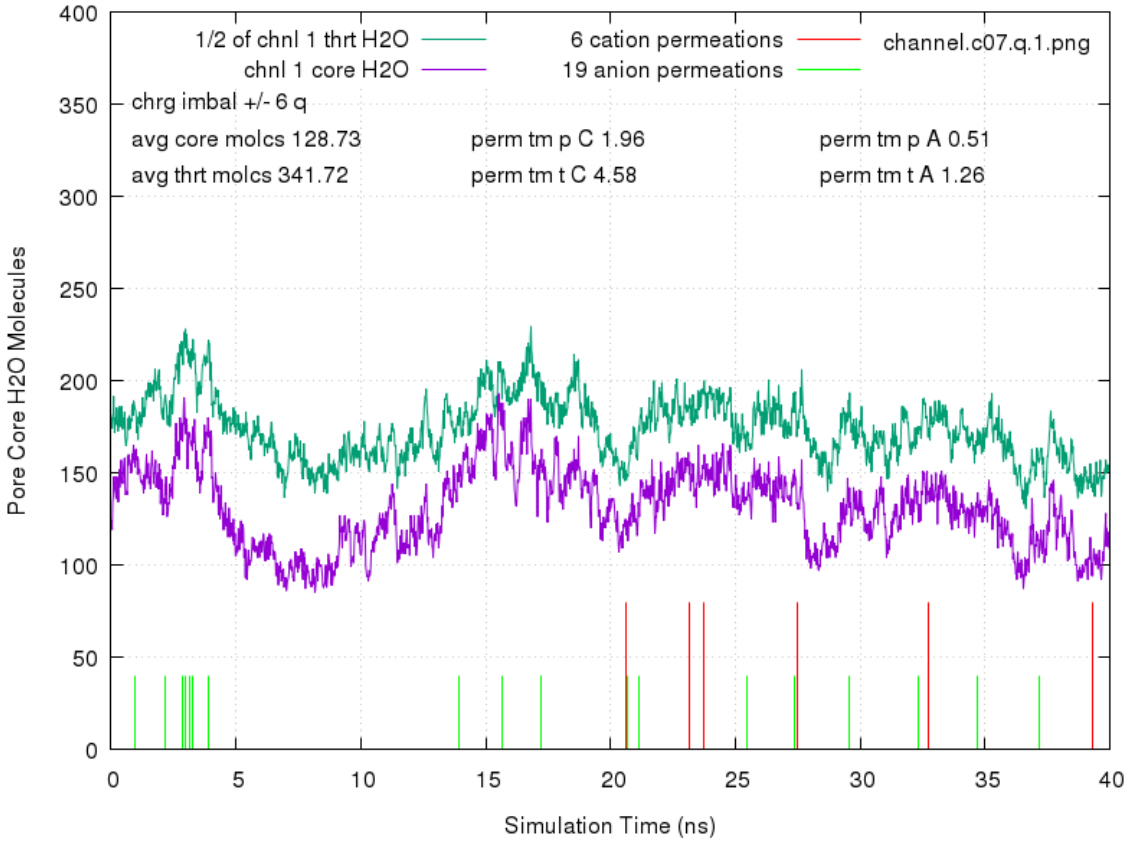

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
1 anion permeations
channel.c0A.O.0.png
chrg imbal +/- 6 q
avg core molcs 87.18
avg thrt molcs 272.60
perm tm p C 4.91
perm tm t C 12.12
perm tm p A 1.54
perm tm t A 4.18
Pore Core H2O Molecules
Simulation Time (ns)

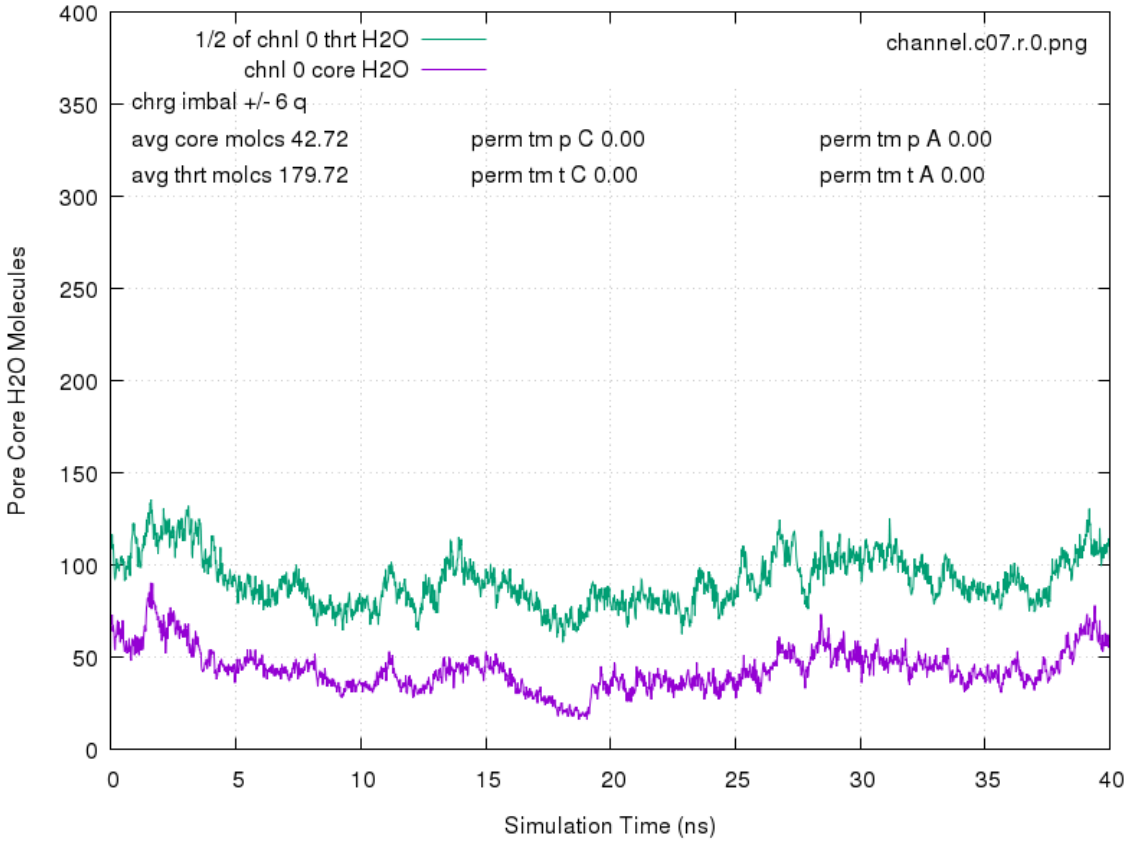

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
channel.c0A.O.1.png
chrg imbal +/- 6 q
avg core molcs 95.31
avg thrt molcs 282.24
perm tm p C 3.04
perm tm t C 6.86
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

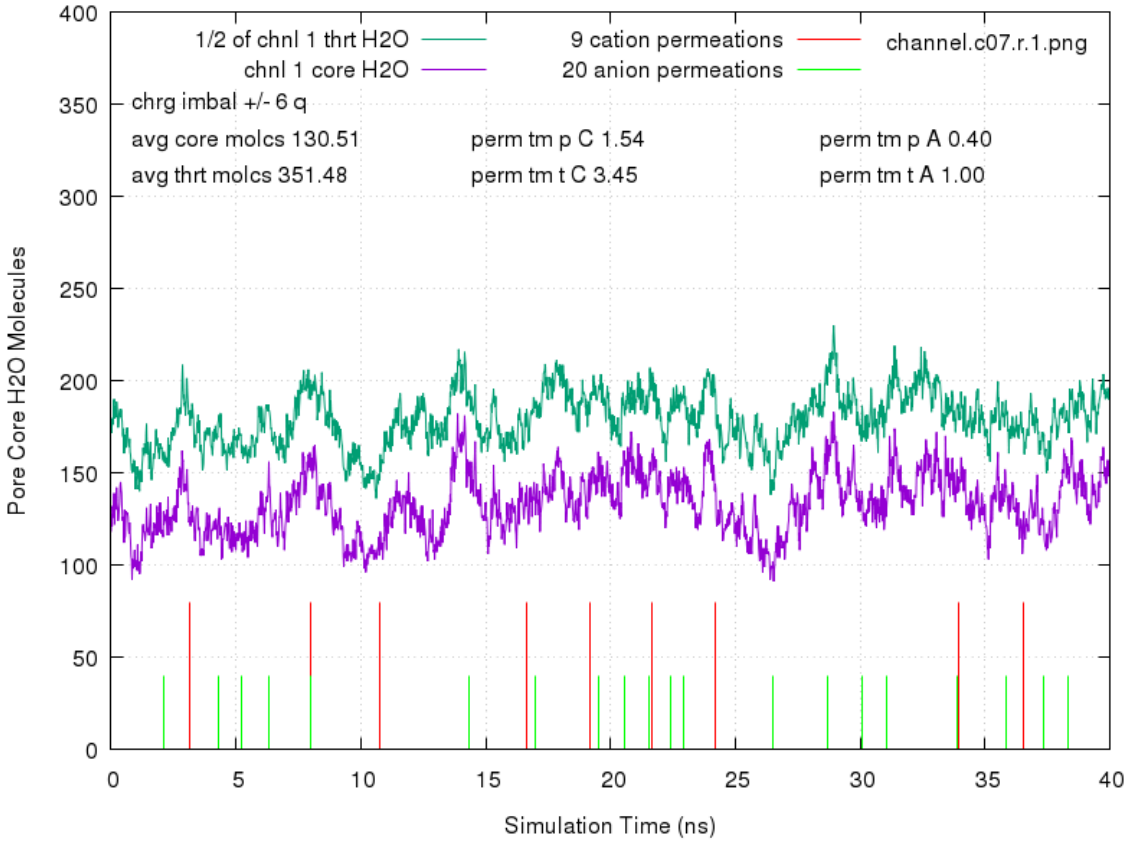

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
1 anion permeations
channel.c07.p.0.png
chrg imbal +/- 6 q
avg core molcs 60.65
avg thrt molcs 208.23
perm tm p C 2.14
perm tm t C 8.26
perm tm p A 1.68
perm tm t A 4.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

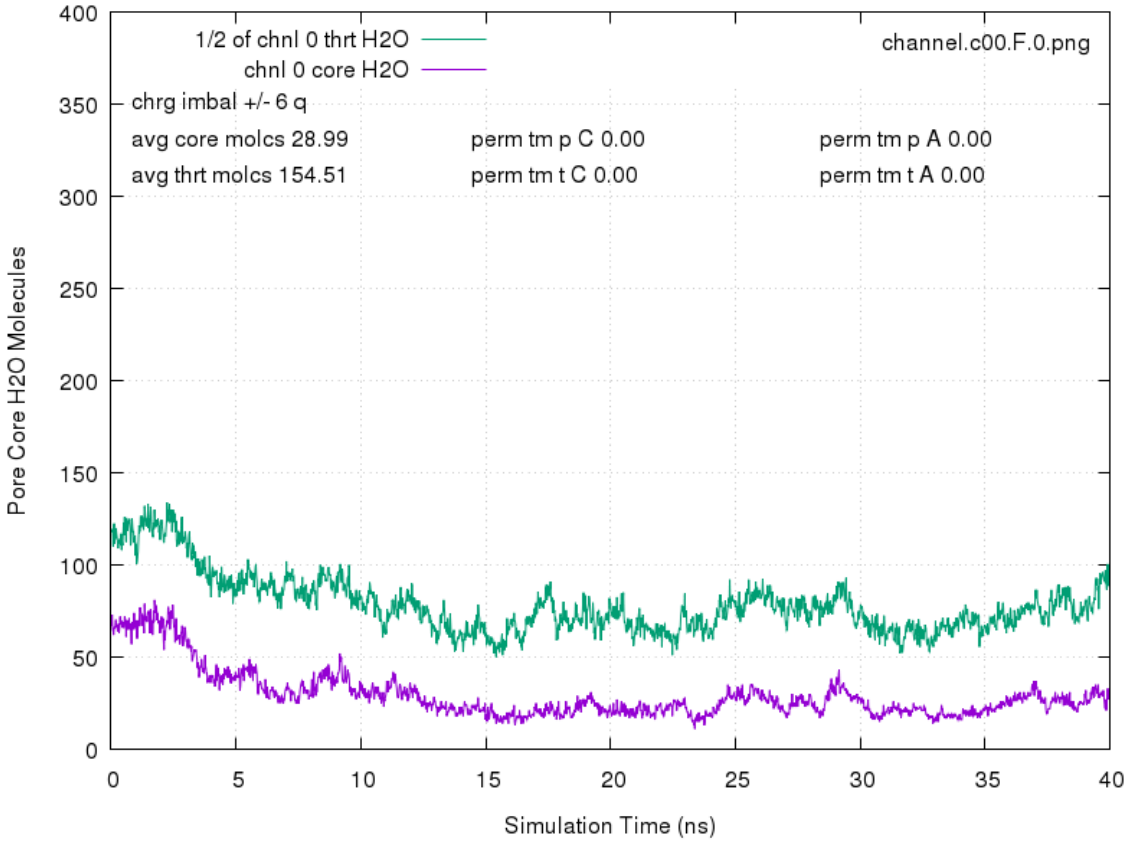

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
5 cation permeations
3 anion permeations
channel.c07.p.1.png
chrg imbal +/- 6 q
avg core molcs 100.92
avg thrt molcs 292.91
perm tm p C 1.86
perm tm t C 3.09
perm tm p A 1.11
perm tm t A 3.23
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

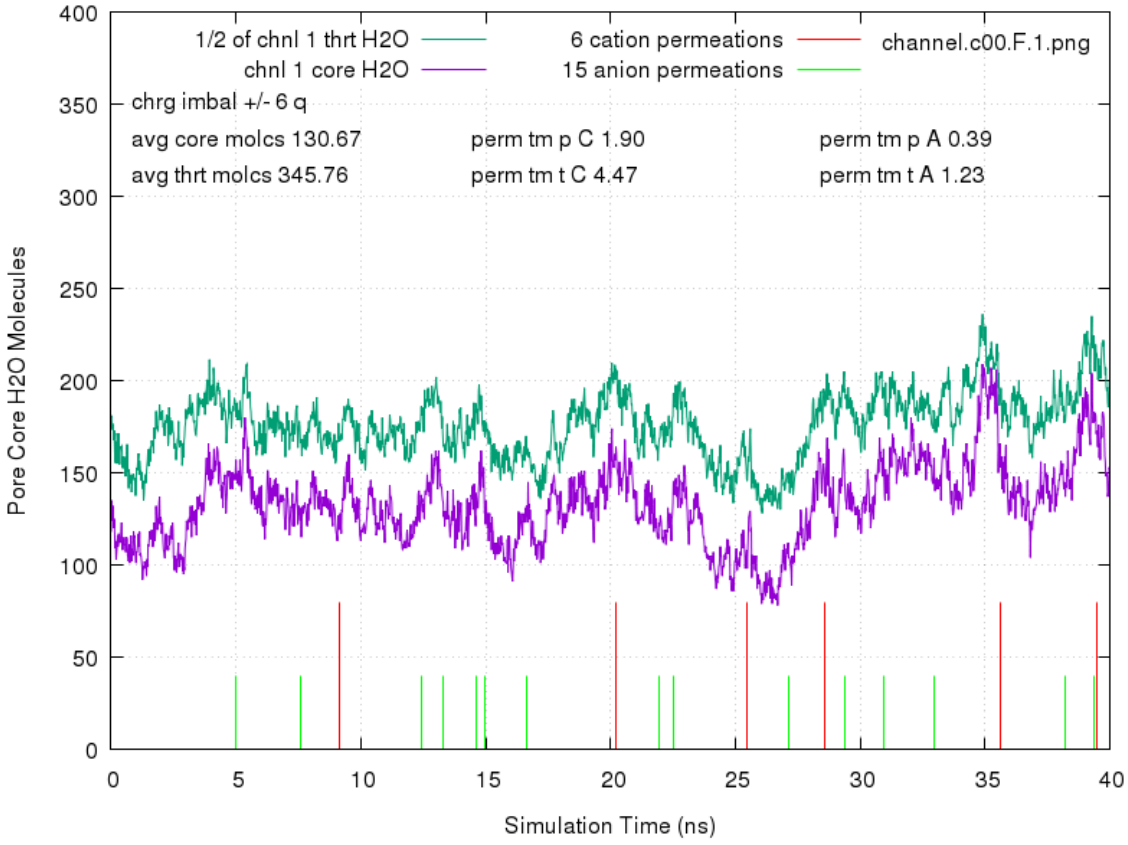

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c07.q.0.png
chrg imbal +/- 6 q
avg core molcs 48.05
avg thrt molcs 196.38
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

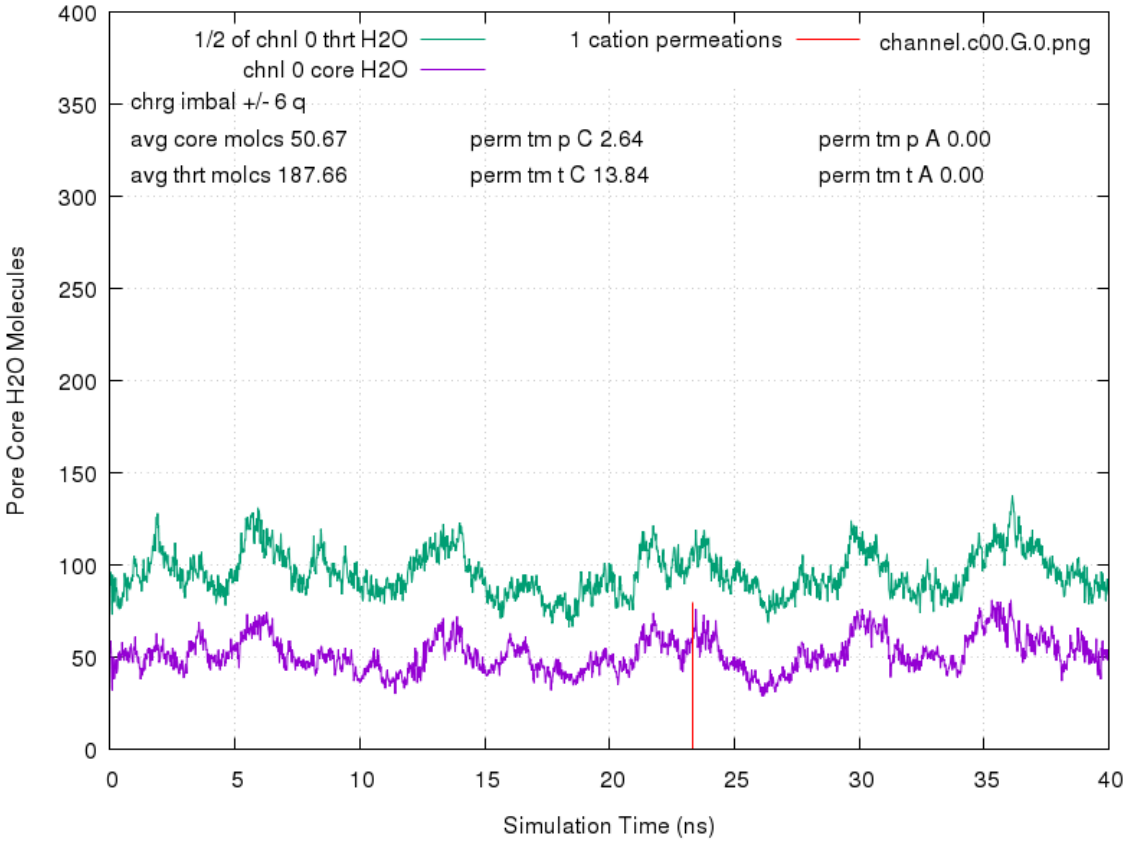

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
6 cation permeations
19 anion permeations
channel.c07.q.1.png
chrg imbal +/- 6 q
avg core molcs 128.73
avg thrt molcs 341.72
perm tm p C 1.96
perm tm t C 4.58
perm tm p A 0.51
perm tm t A 1.26
Pore Core H2O Molecules
Simulation Time (ns)

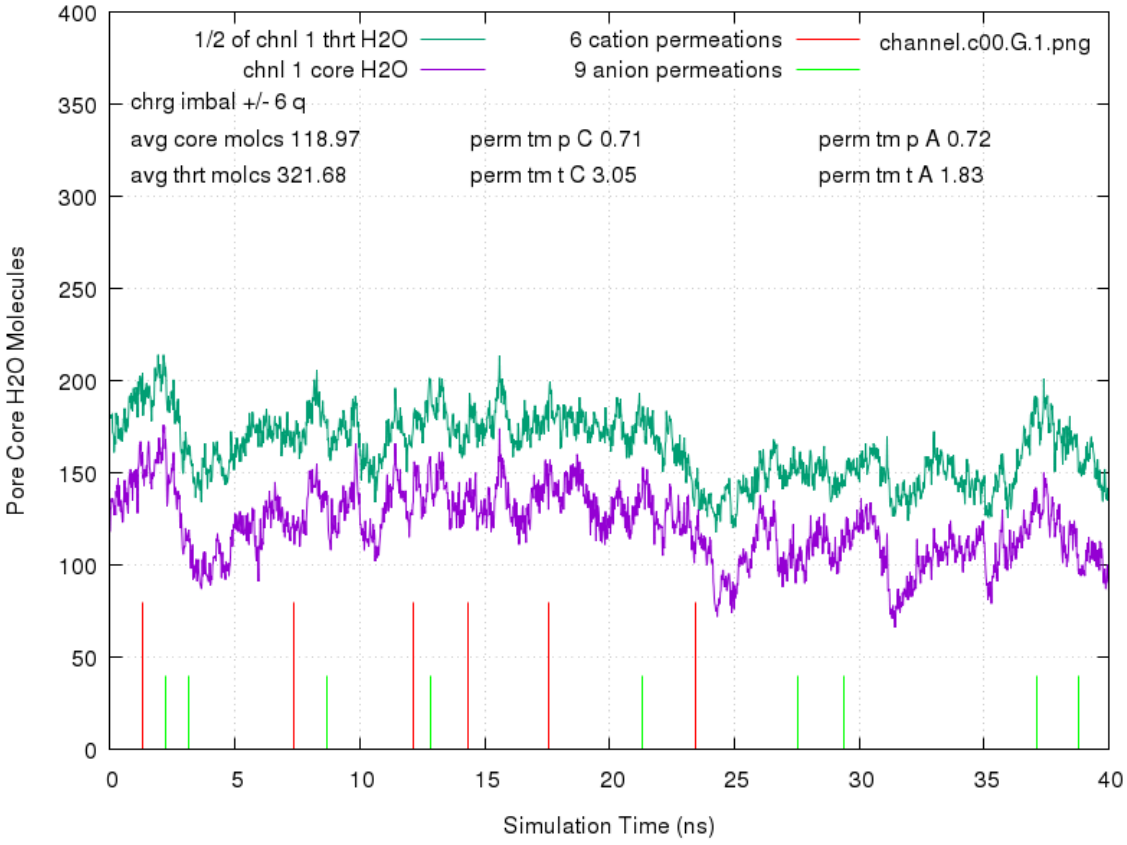

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c07.r.0.png
chrg imbal +/- 6 q
avg core molcs 42.72
avg thrt molcs 179.72
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

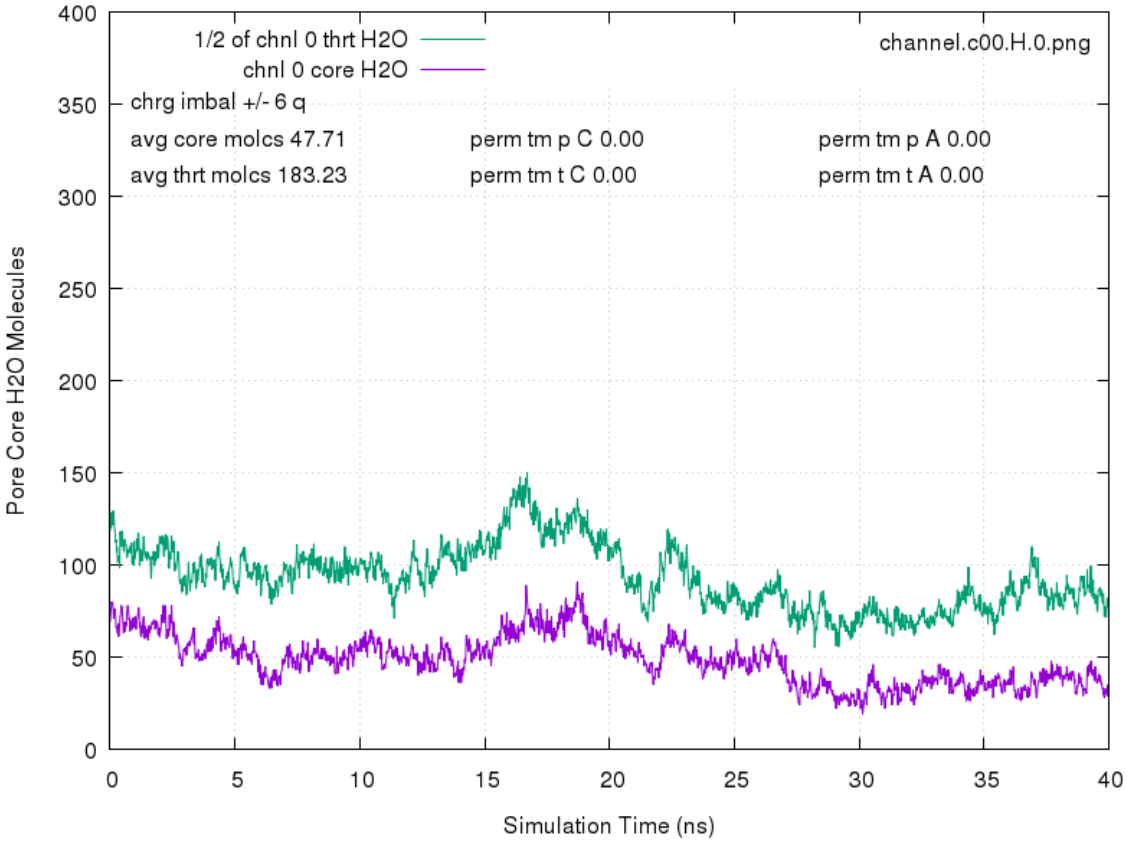

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
9 cation permeations
20 anion permeations
channel.c07.r.1.png
chrg imbal +/- 6 q
avg core molcs 130.51
avg thrt molcs 351.48
perm tm p C 1.54
perm tm t C 3.45
perm tm p A 0.40
perm tm t A 1.00
Pore Core H2O Molecules
Simulation Time (ns)

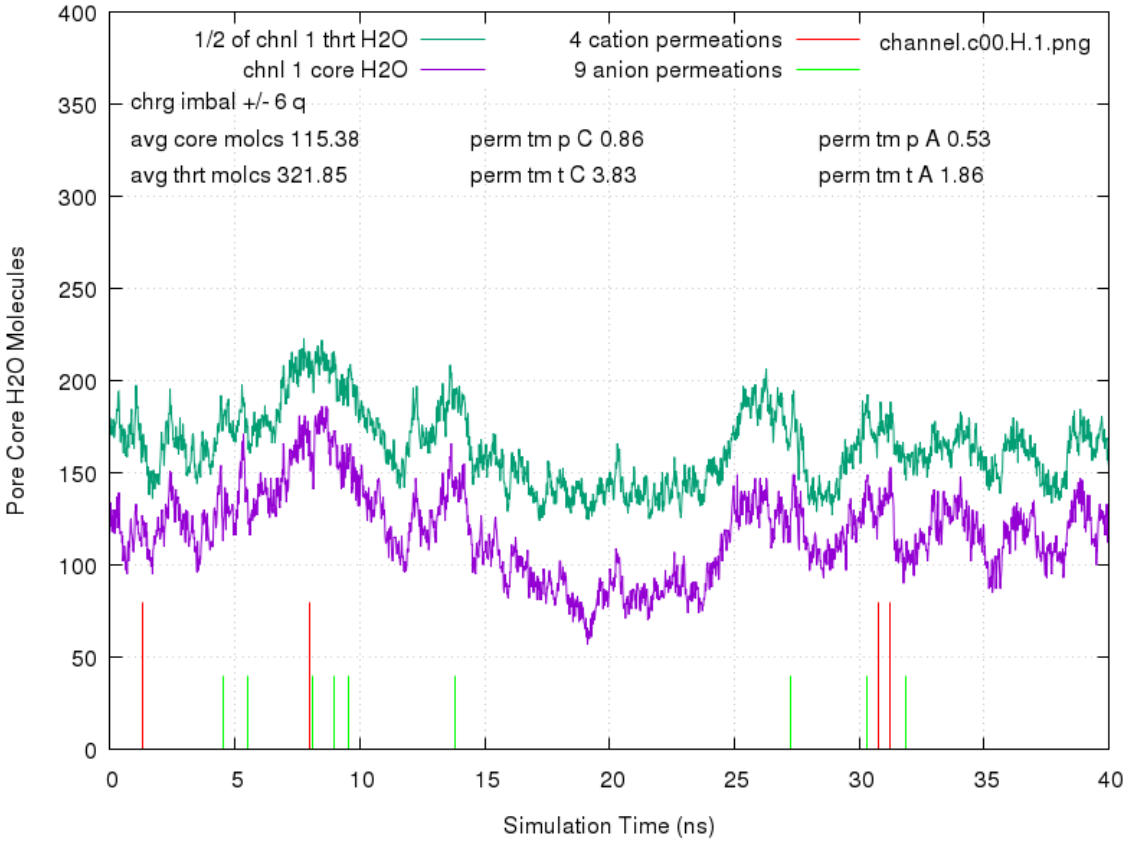

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c00.F.0.png
chrg imbal +/- 6 q
avg core molcs 28.99
avg thrt molcs 154.51
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

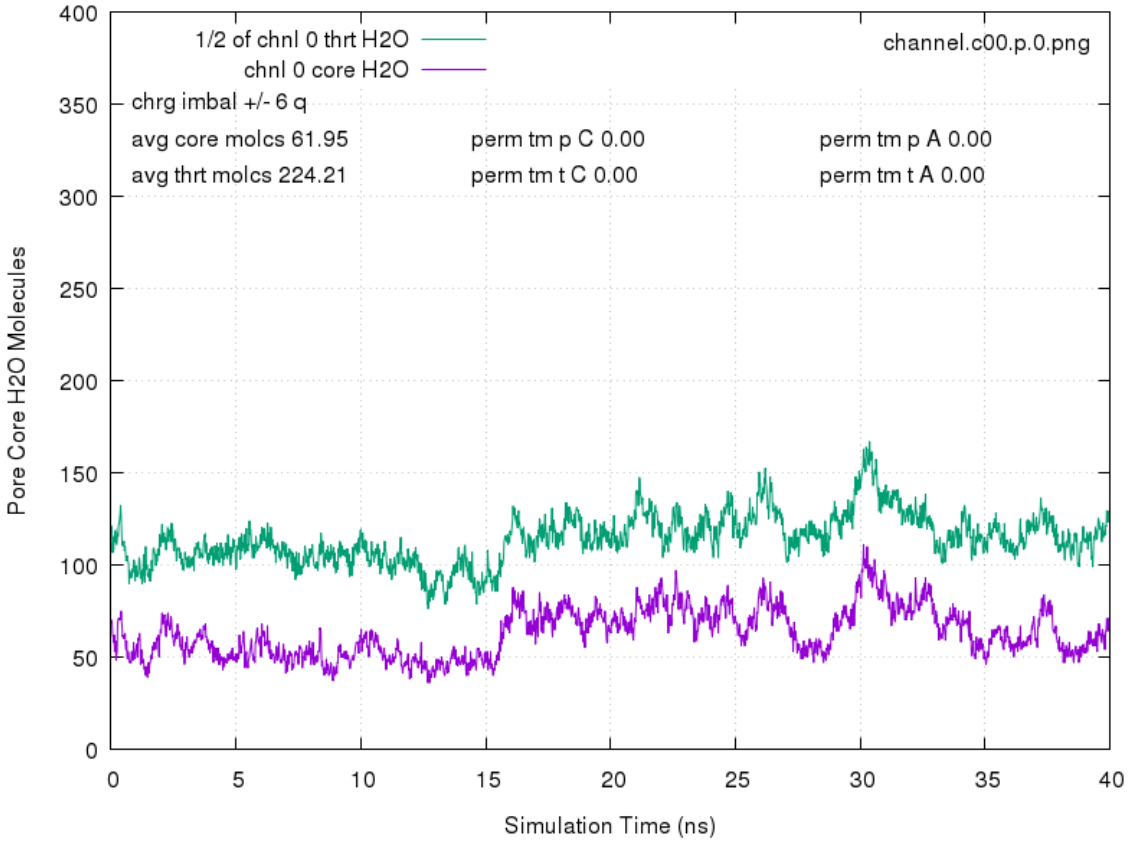

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
6 cation permeations
15 anion permeations
channel.c00.F.1.png
chrg imbal +/- 6 q
avg core molcs 130.67
avg thrt molcs 345.76
perm tm p C 1.90
perm tm t C 4.47
perm tm p A 0.39
perm tm t A 1.23
Pore Core H2O Molecules
Simulation Time (ns)

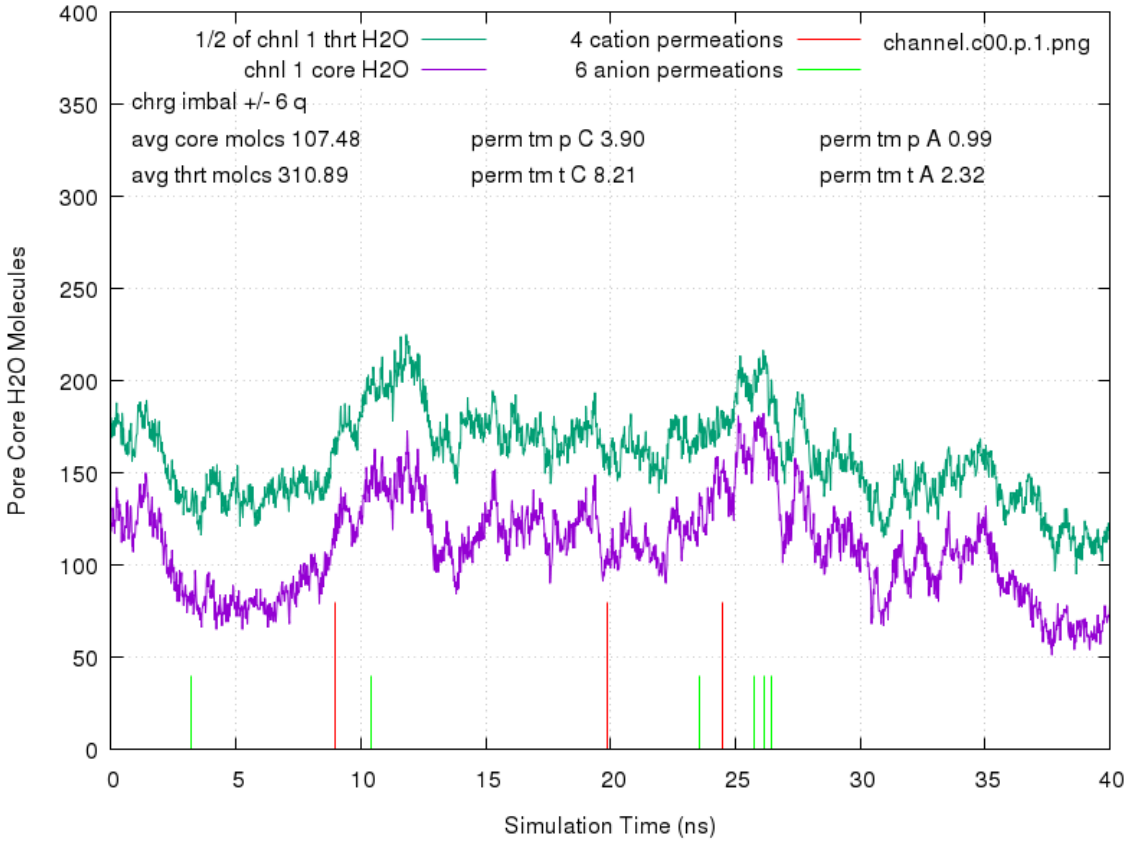

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
channel.c00.G.0.png
chrg imbal +/- 6 q
avg core molcs 50.67
avg thrt molcs 187.66
perm tm p C 2.64
perm tm t C 13.84
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

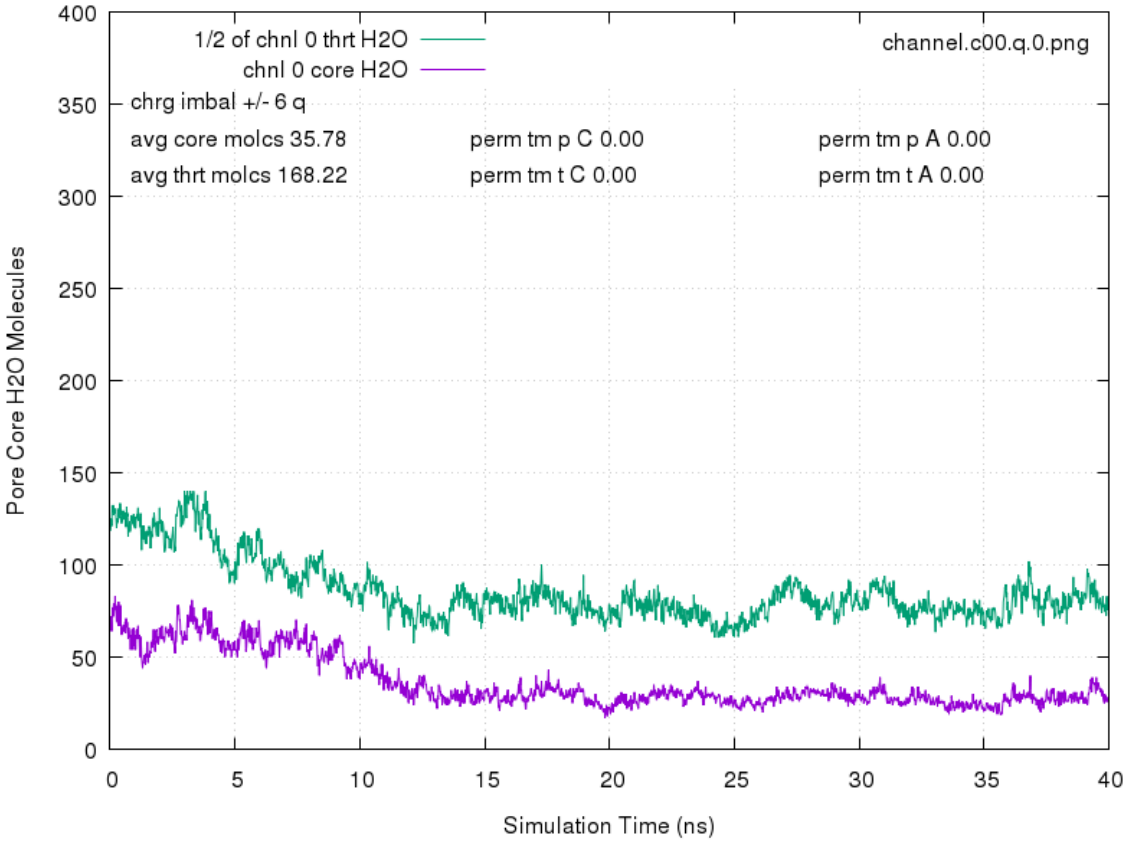

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
6 cation permeations
9 anion permeations
channel.c00.G.1.png
chrg imbal +/- 6 q
avg core molcs 118.97
avg thrt molcs 321.68
perm tm p C 0.71
perm tm t C 3.05
perm tm p A 0.72
perm tm t A 1.83
Pore Core H2O Molecules
Simulation Time (ns)

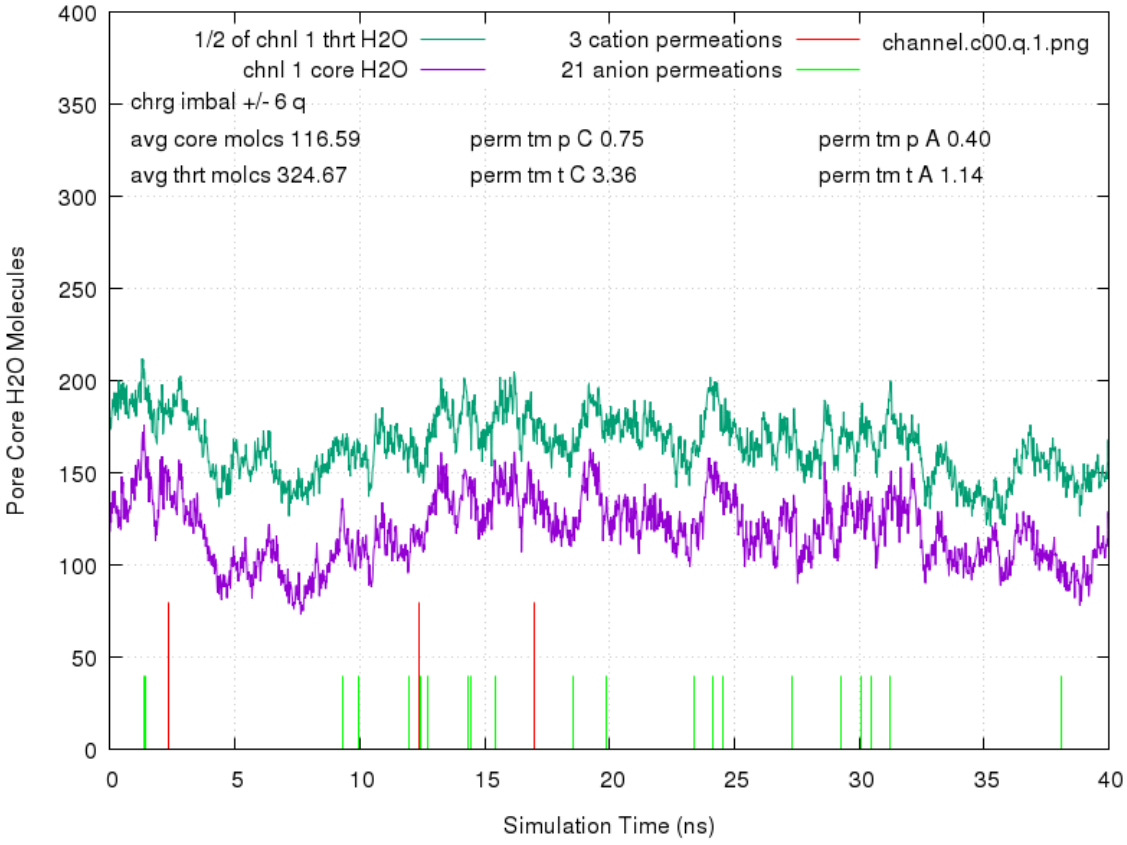

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c00.H.0.png
chrg imbal +/- 6 q
avg core molcs 47.71
avg thrt molcs 183.23
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

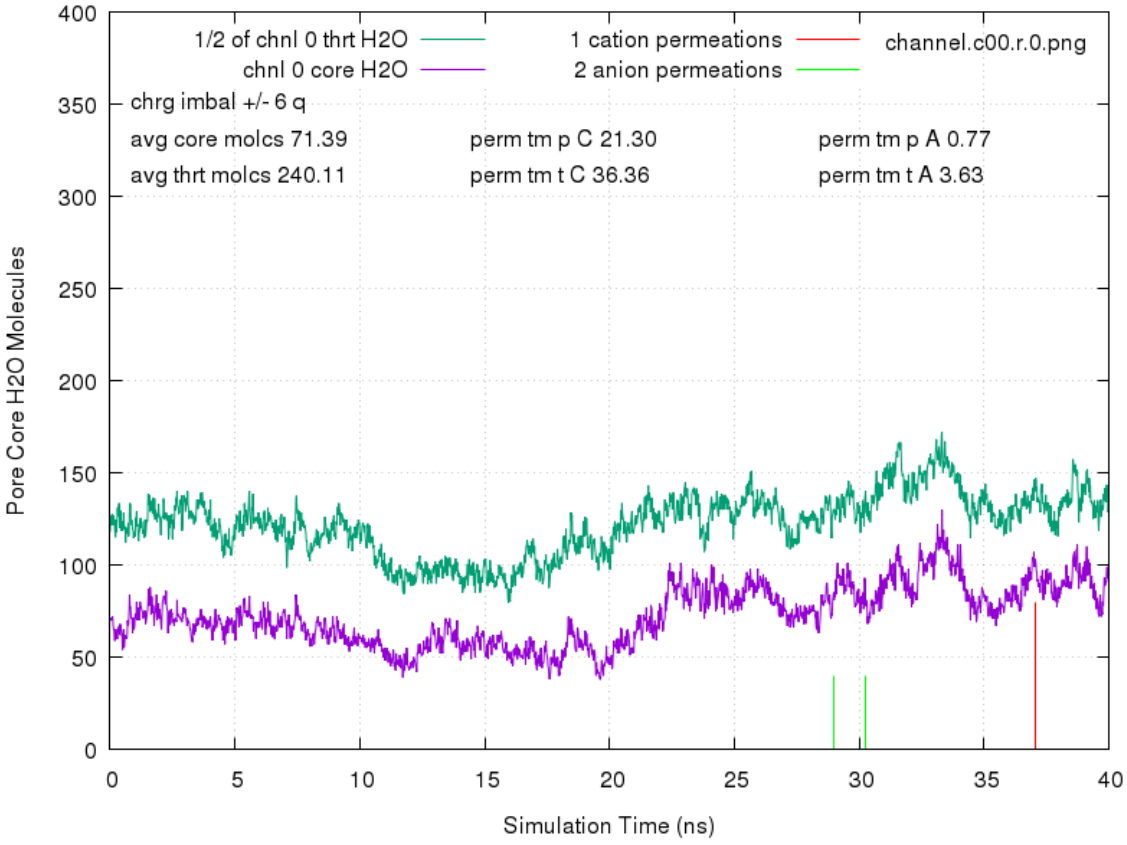

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
9 anion permeations
channel.c00.H.1.png
chrg imbal +/- 6 q
avg core molcs 115.38
avg thrt molcs 321.85
perm tm p C 0.86
perm tm t C 3.83
perm tm p A 0.53
perm tm t A 1.86
Pore Core H2O Molecules
Simulation Time (ns)

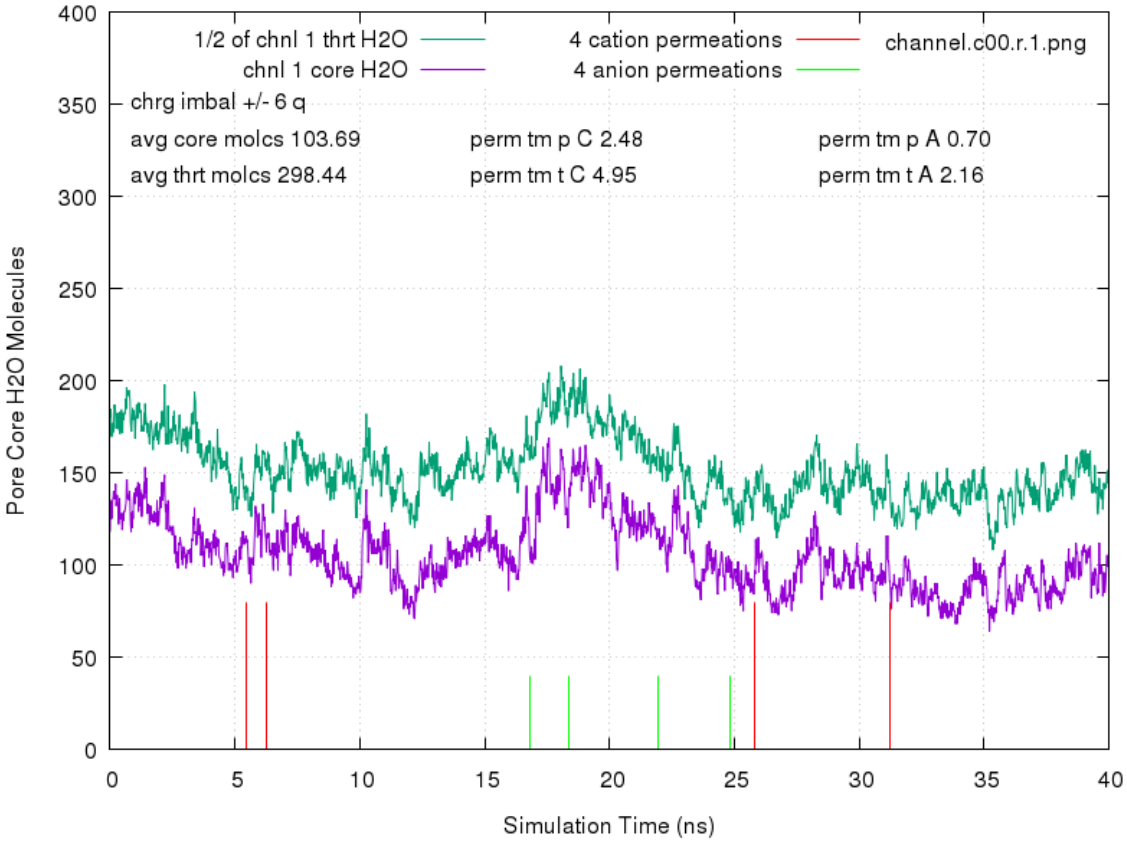

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c00.p.0.png
chrg imbal +/- 6 q
avg core molcs 61.95
avg thrt molcs 224.21
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

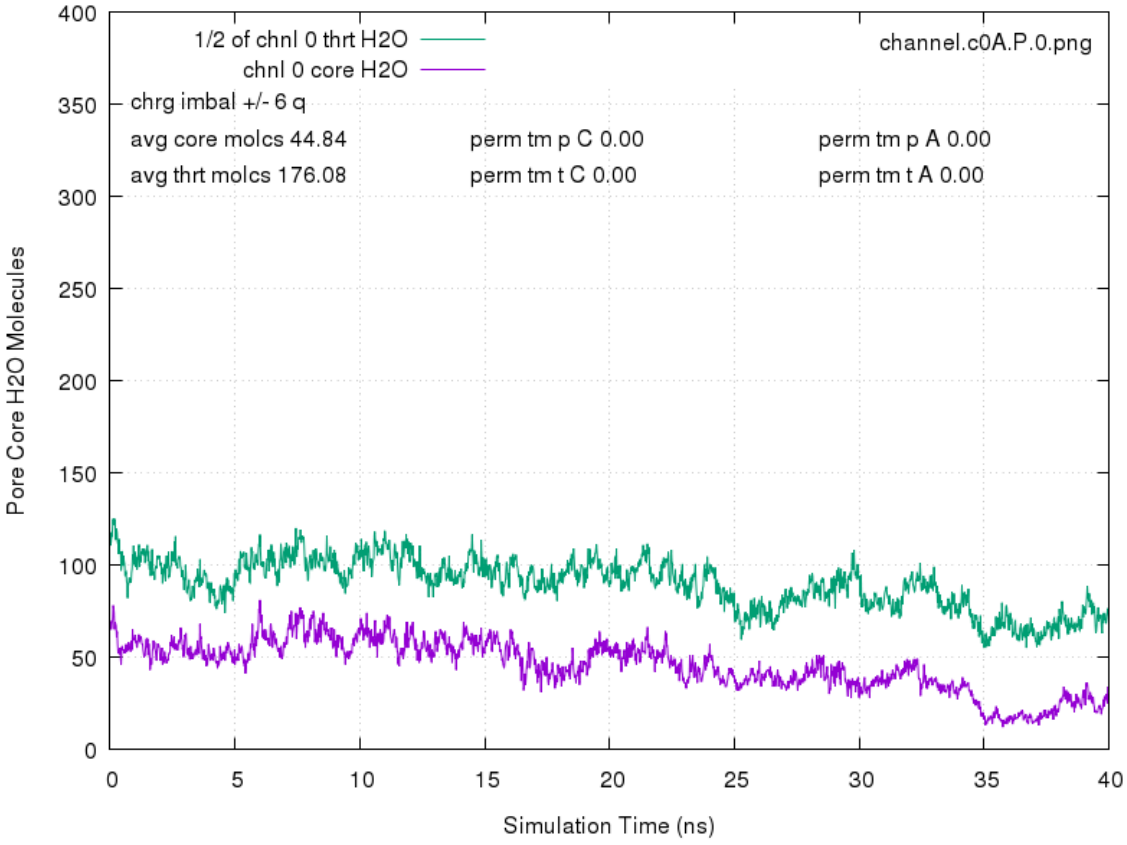

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
6 anion permeations
channel.c00.p.1.png
chrg imbal +/- 6 q
avg core molcs 107.48
avg thrt molcs 310.89
perm tm p C 3.90
perm tm t C 8.21
perm tm p A 0.99
perm tm t A 2.32
Pore Core H2O Molecules
Simulation Time (ns)

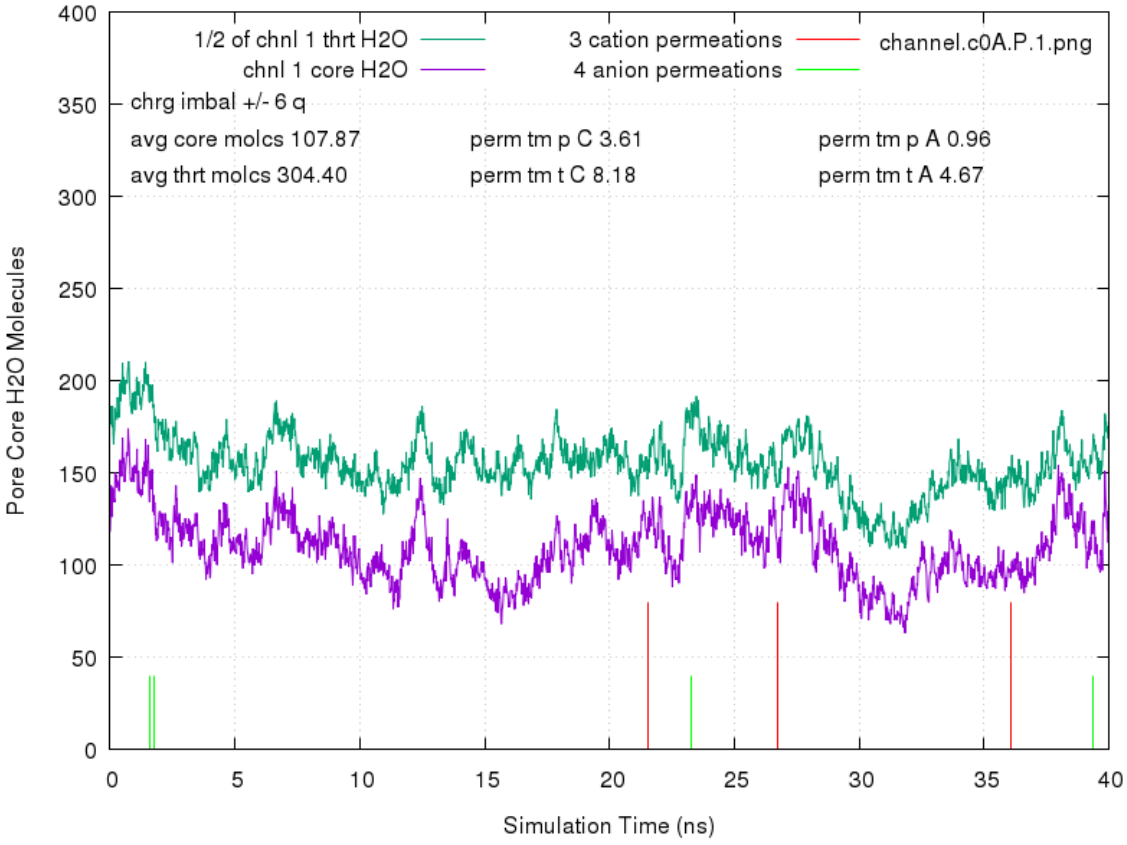

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c00.q.0.png
chrg imbal +/- 6 q
avg core molcs 35.78
avg thrt molcs 168.22
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

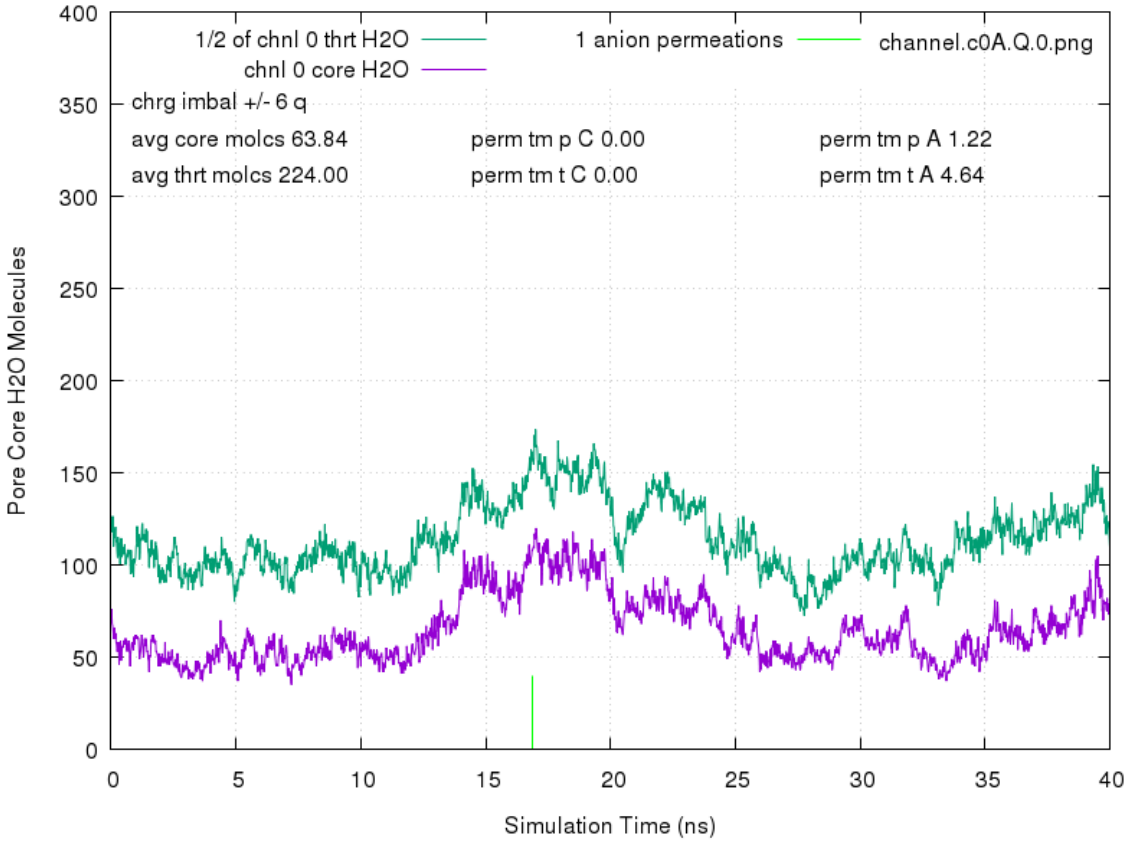

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
21 anion permeations
channel.c00.q.1.png
chrg imbal +/- 6 q
avg core molcs 116.59
avg thrt molcs 324.67
perm tm p C 0.75
perm tm t C 3.36
perm tm p A 0.40
perm tm t A 1.14
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

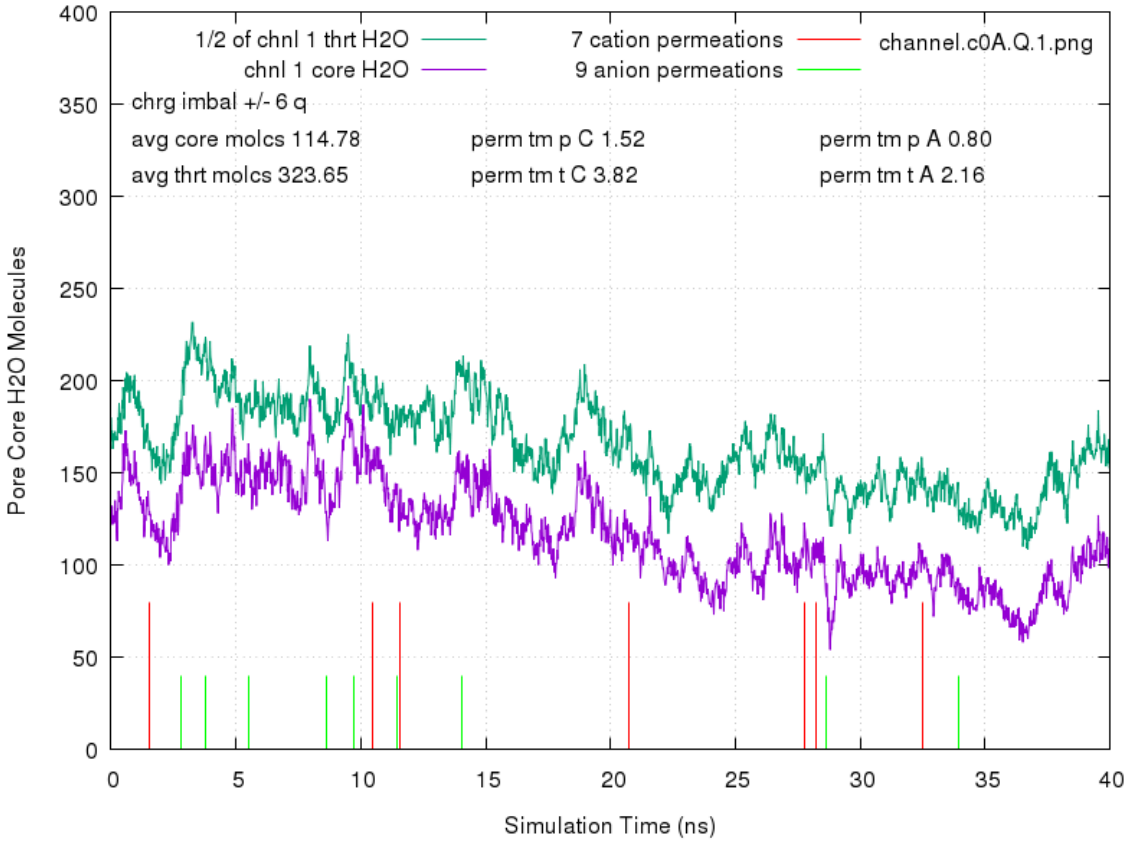

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
2 anion permeations
channel.c00.r.0.png
chrg imbal +/- 6 q
avg core molcs 71.39
avg thrt molcs 240.11
perm tm p C 21.30
perm tm t C 36.36
perm tm p A 0.77
perm tm t A 3.63
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

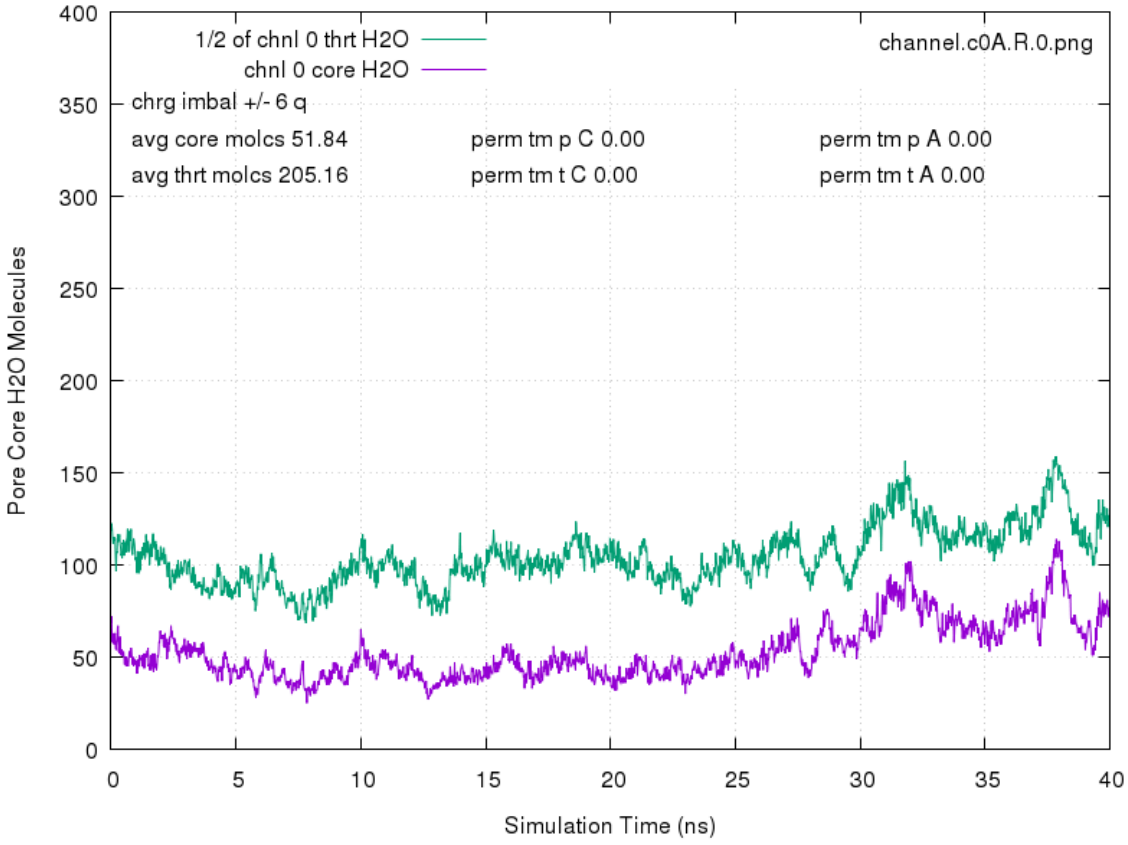

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
4 anion permeations
channel.c00.r.1.png
chrg imbal +/- 6 q
avg core molcs 103.69
avg thrt molcs 298.44
perm tm p C 2.48
perm tm t C 4.95
perm tm p A 0.70
perm tm t A 2.16
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

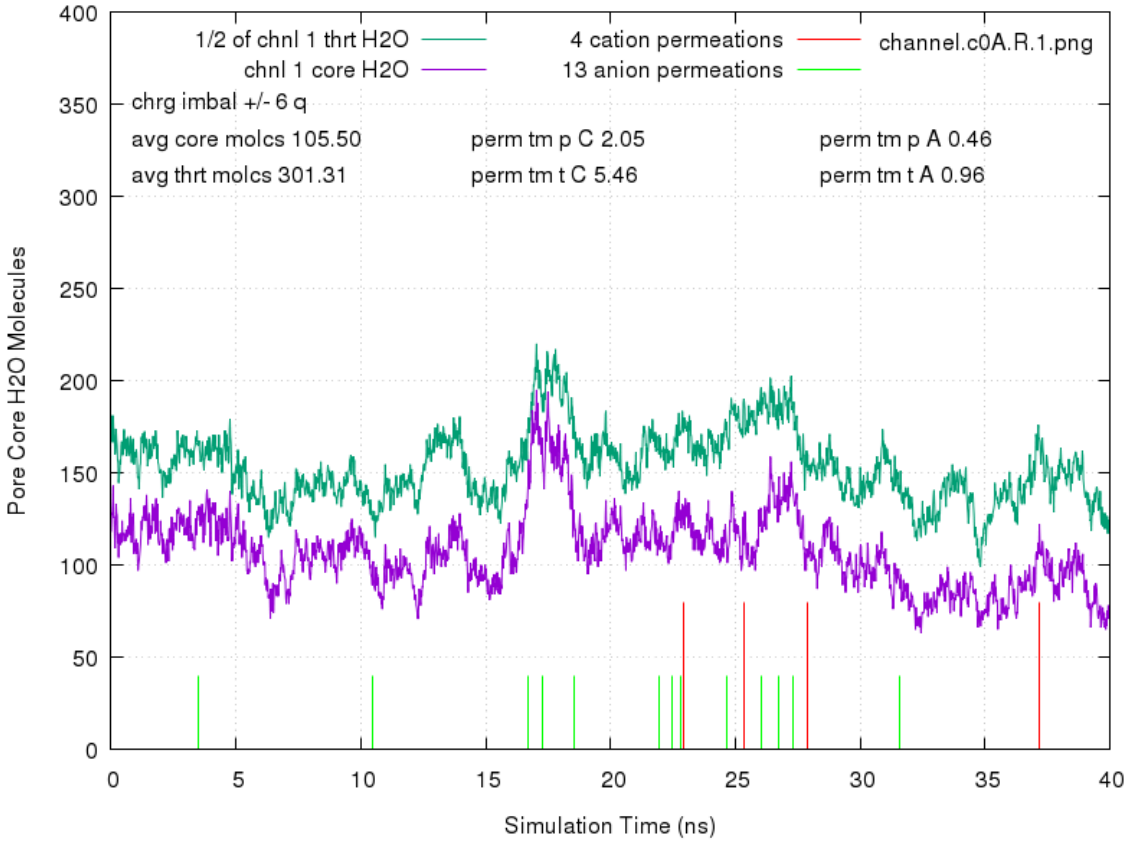

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c0A.P.0.png
chrg imbal +/- 6 q
avg core molcs 44.84
avg thrt molcs 176.08
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

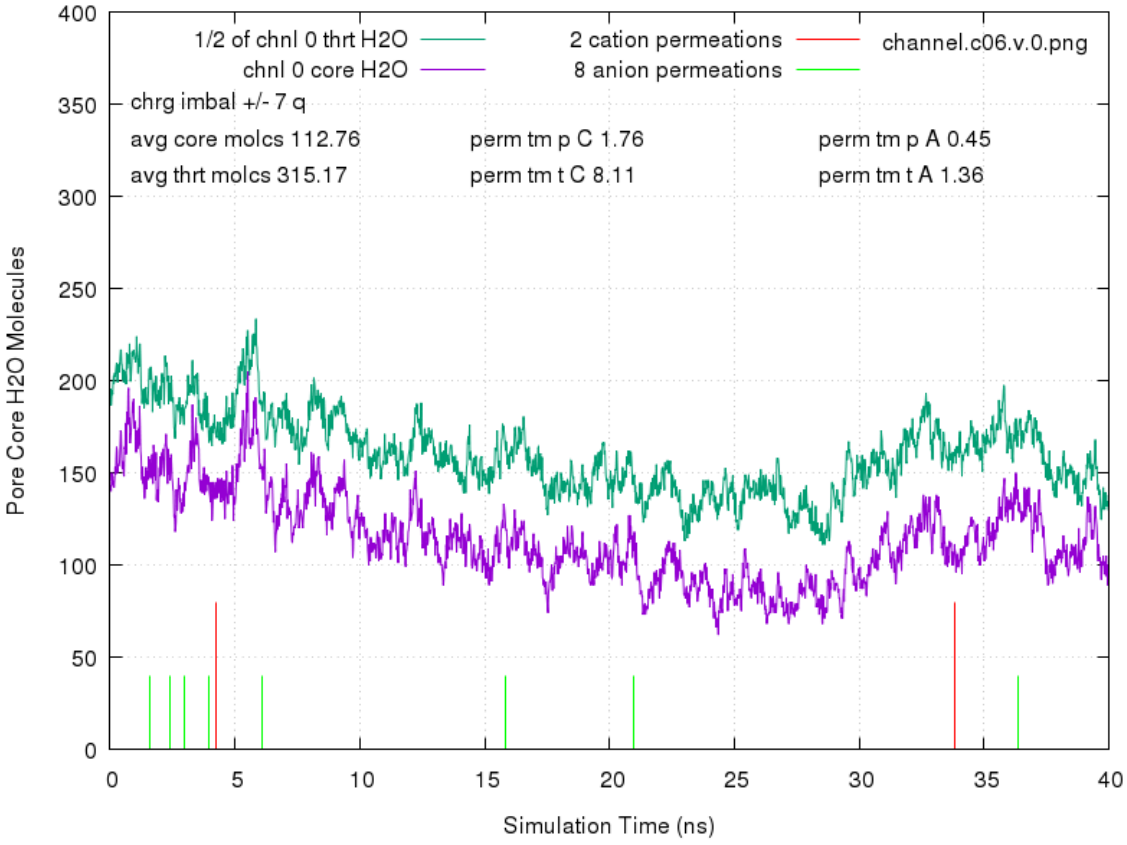

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
4 anion permeations
channel.c0A.P.1.png
chrg imbal +/- 6 q
avg core molcs 107.87
avg thrt molcs 304.40
perm tm p C 3.61
perm tm t C 8.18
perm tm p A 0.96
perm tm t A 4.67
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

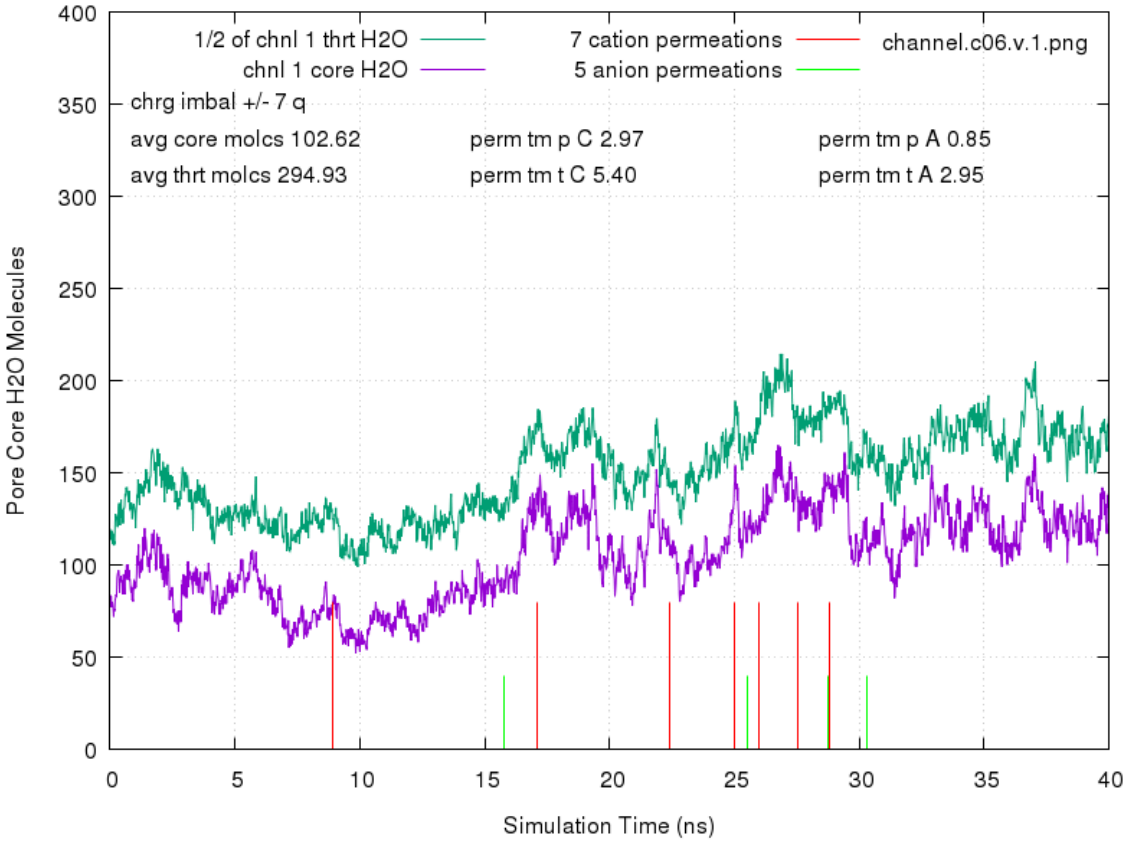

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 anion permeations
channel.c0A.Q.0.png
chrg imbal +/- 6 q
avg core molcs 63.84
avg thrt molcs 224.00
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 1.22
perm tm t A 4.64
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

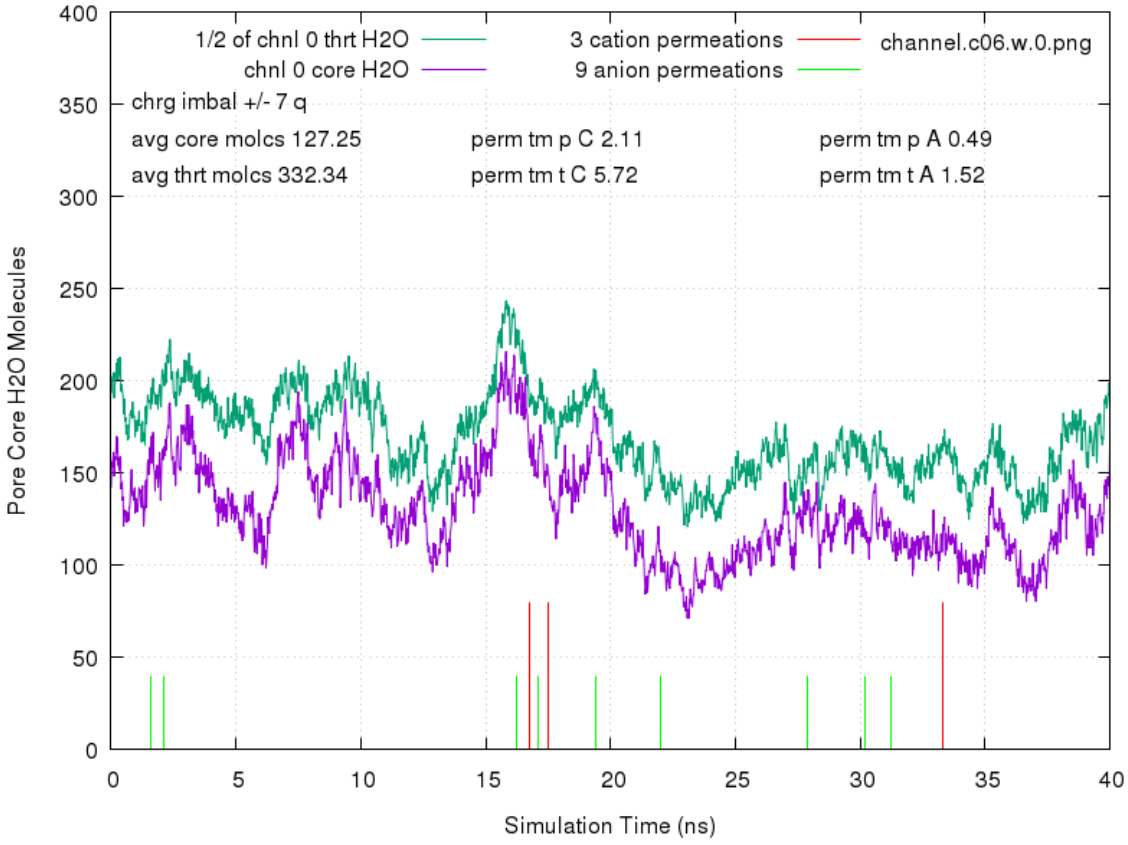

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
7 cation permeations
9 anion permeations
channel.c0A.Q.1.png
chrg imbal +/- 6 q
avg core molcs 114.78
avg thrt molcs 323.65
perm tm p C 1.52
perm tm t C 3.82
perm tm p A 0.80
perm tm t A 2.16
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

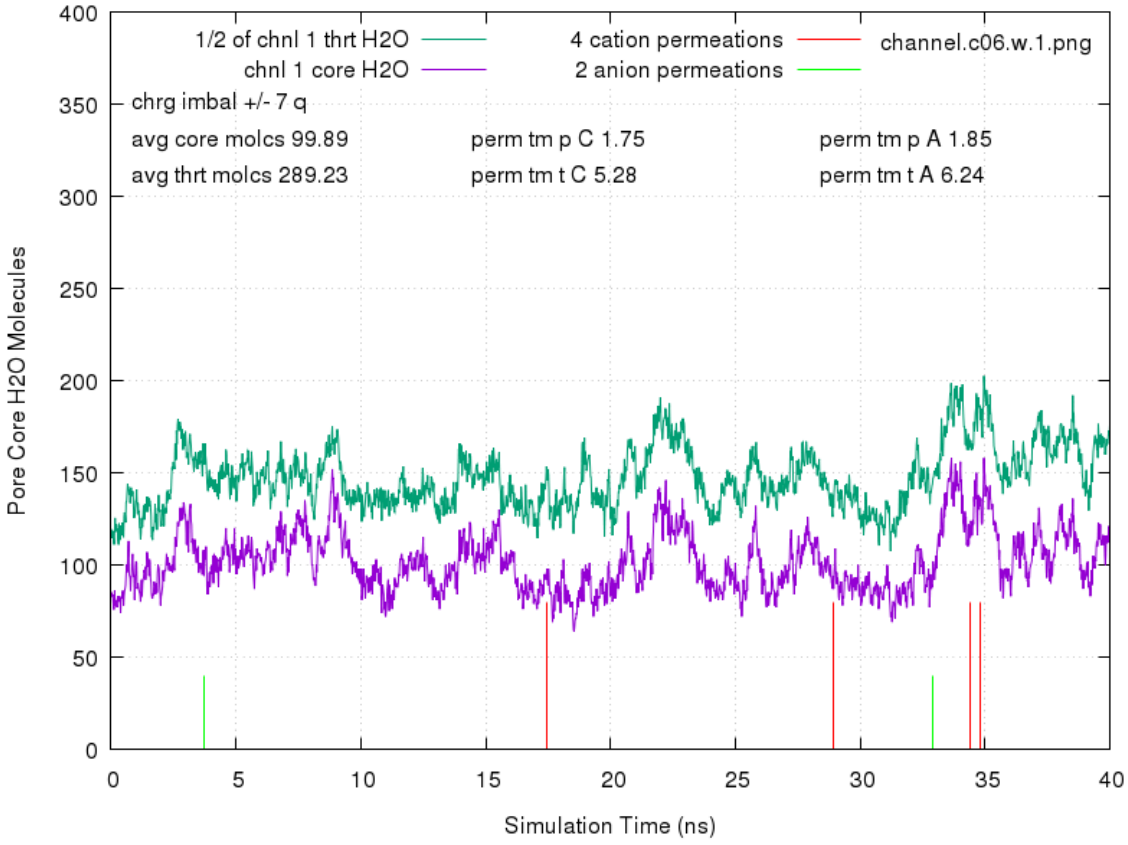

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
channel.c0A.R.0.png
chrg imbal +/- 6 q
avg core molcs 51.84
avg thrt molcs 205.16
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 0.00
perm tm t A 0.00
Pore Core H2O Molecules
Simulation Time (ns)

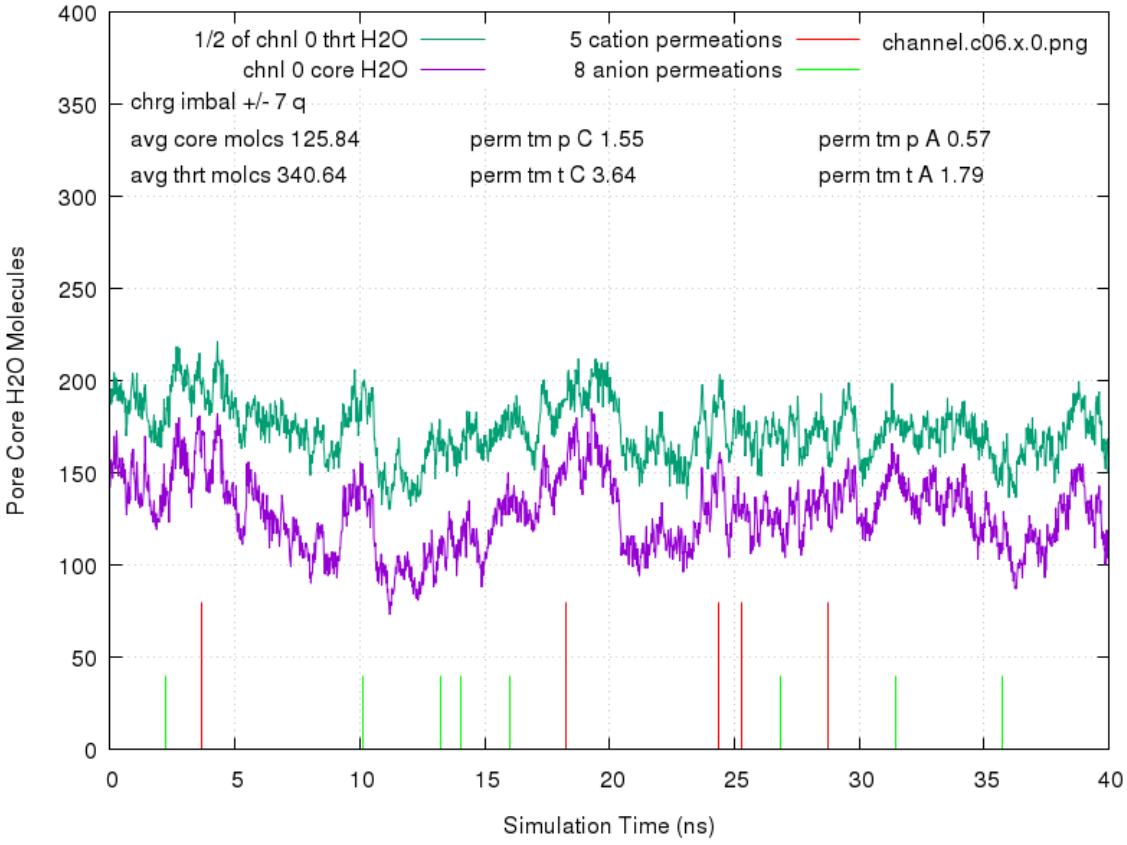

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
13 anion permeations
channel.c0A.R.1.png
chrg imbal +/- 6 q
avg core molcs 105.50
avg thrt molcs 301.31
perm tm p C 2.05
perm tm t C 5.46
perm tm p A 0.46
perm tm t A 0.96
Pore Core H2O Molecules
Simulation Time (ns)

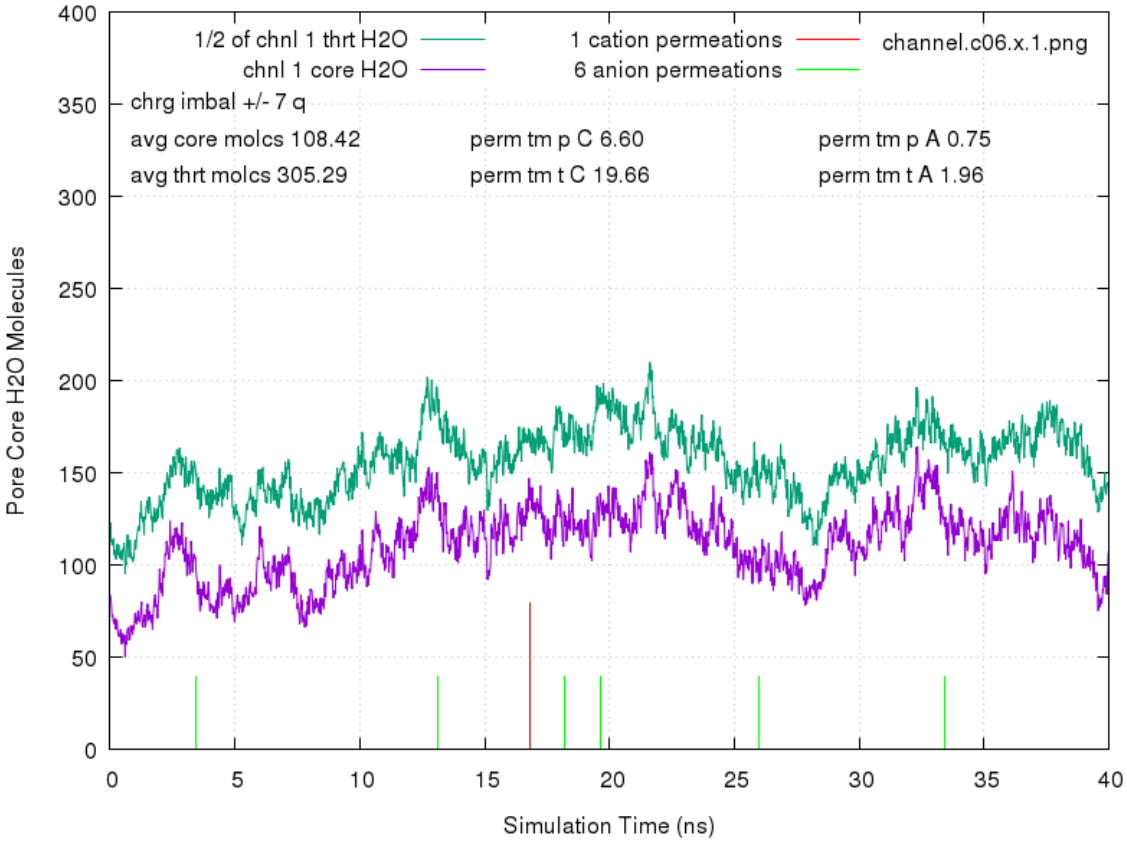

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
8 anion permeations
channel.c06.v.0.png
chrg imbal +/- 7 q
avg core molcs 112.76
avg thrt molcs 315.17
perm tm p C 1.76
perm tm t C 8.11
perm tm p A 0.45
perm tm t A 1.36
Pore Core H2O Molecules
Simulation Time (ns)

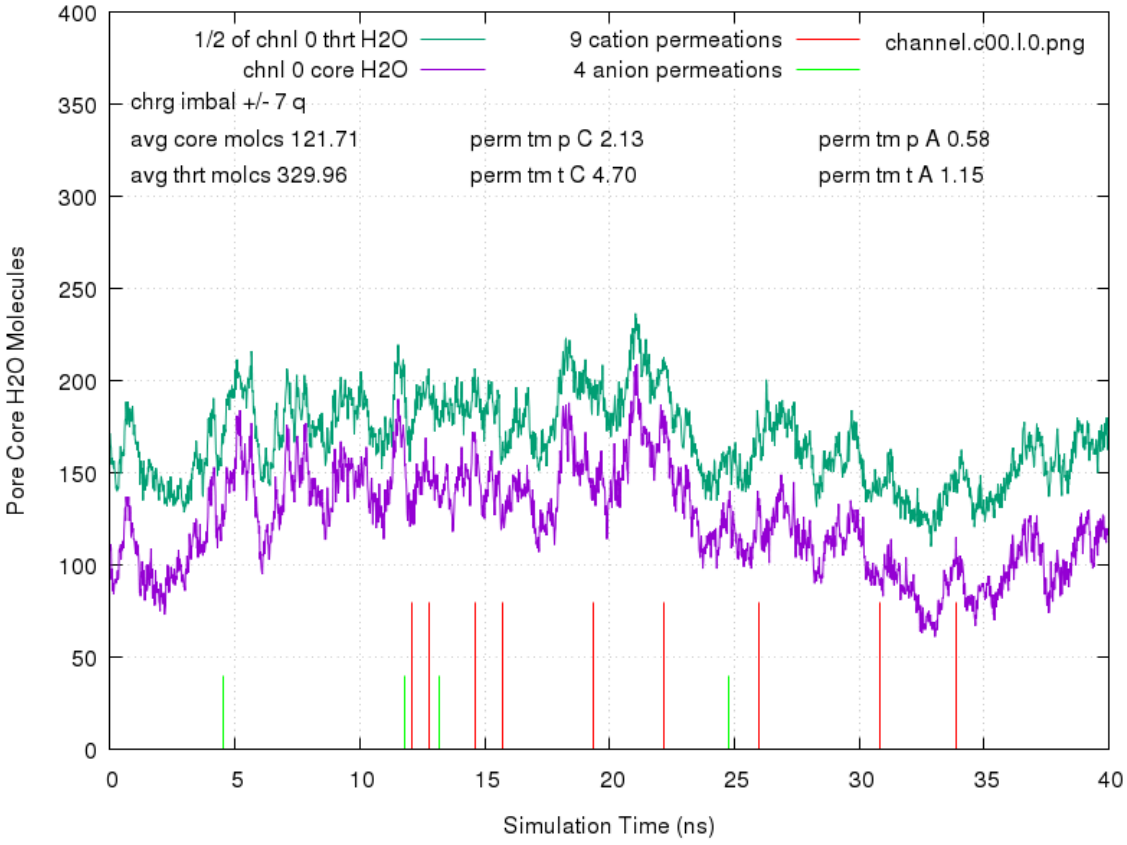

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
7 cation permeations
5 anion permeations
channel.c06.v.1.png
chrg imbal +/- 7 q
avg core molcs 102.62
avg thrt molcs 294.93
perm tm p C 2.97
perm tm t C 5.40
perm tm p A 0.85
perm tm t A 2.95
Pore Core H2O Molecules
Simulation Time (ns)

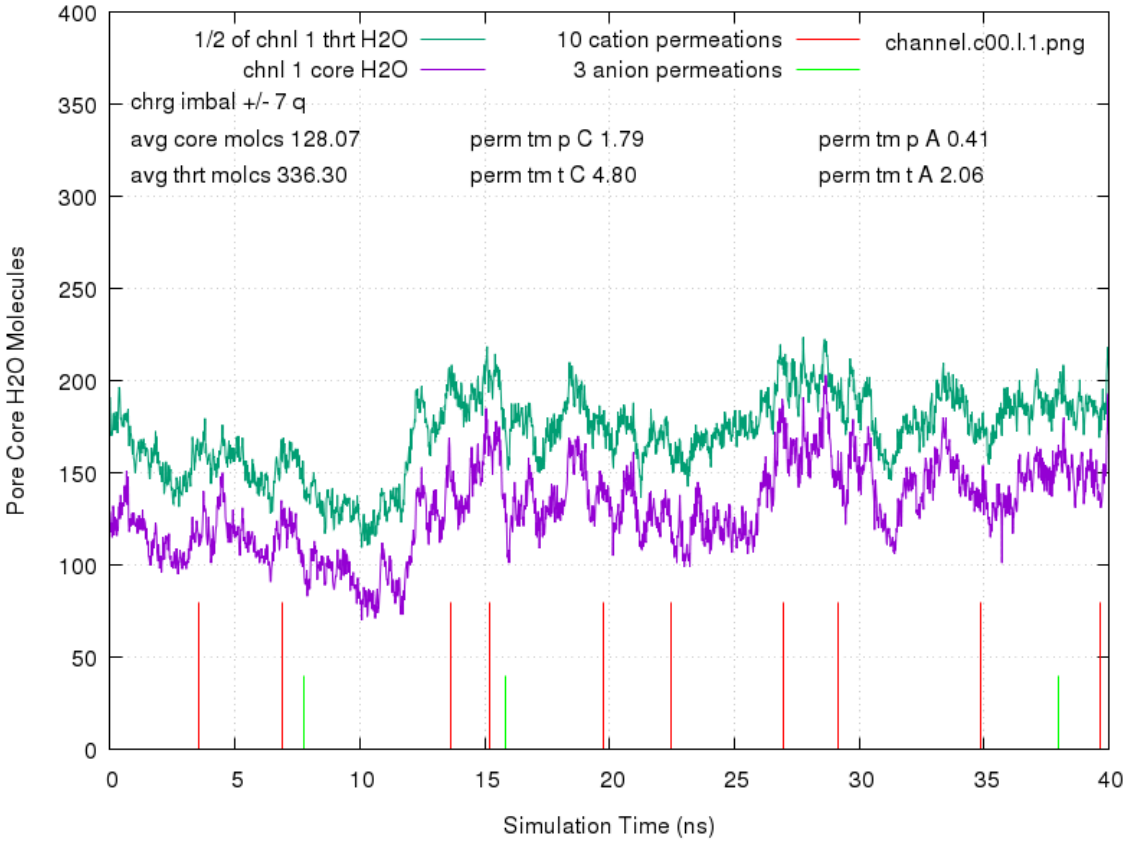

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
9 anion permeations
channel.c06.w.0.png
chrg imbal +/- 7 q
avg core molcs 127.25
avg thrt molcs 332.34
perm tm p C 2.11
perm tm t C 5.72
perm tm p A 0.49
perm tm t A 1.52
Pore Core H2O Molecules
Simulation Time (ns)

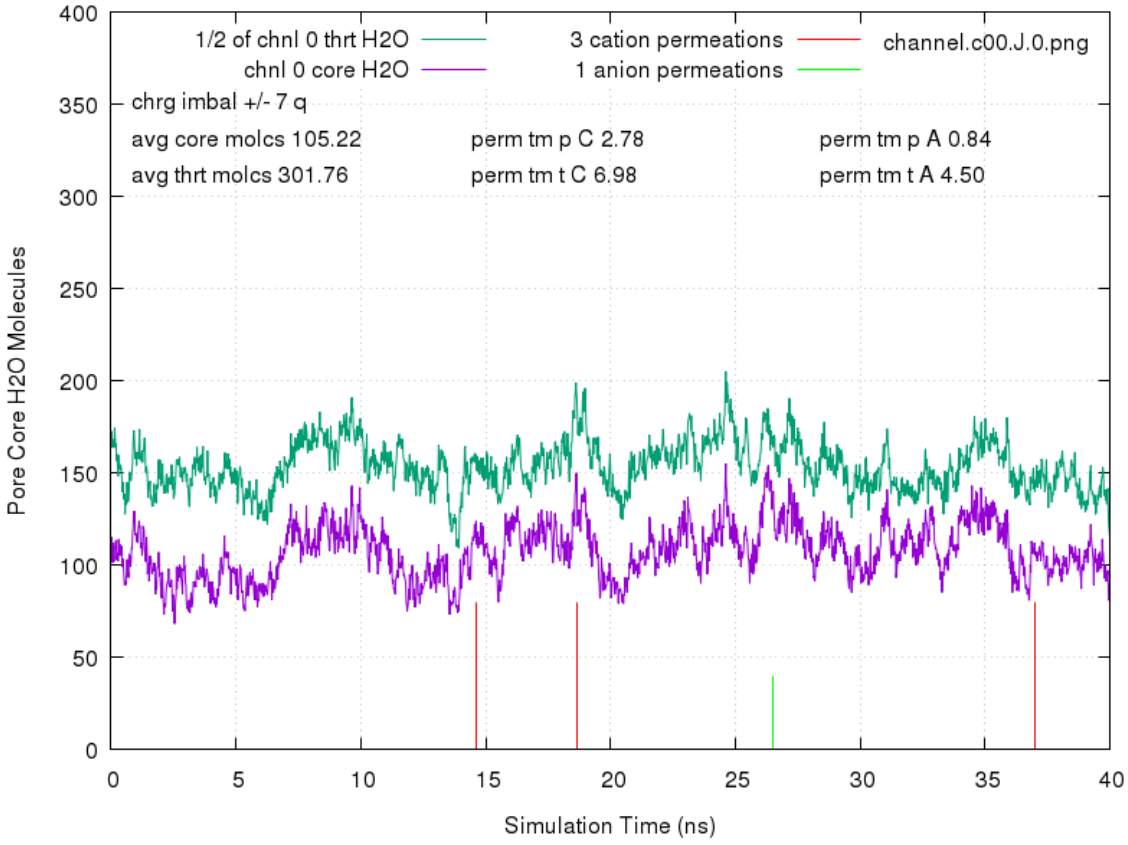

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
2 anion permeations
channel.c06.w.1.png
chrg imbal +/- 7 q
avg core molcs 99.89
avg thrt molcs 289.23
perm tm p C 1.75
perm tm t C 5.28
perm tm p A 1.85
perm tm t A 6.24
Pore Core H2O Molecules
Simulation Time (ns)

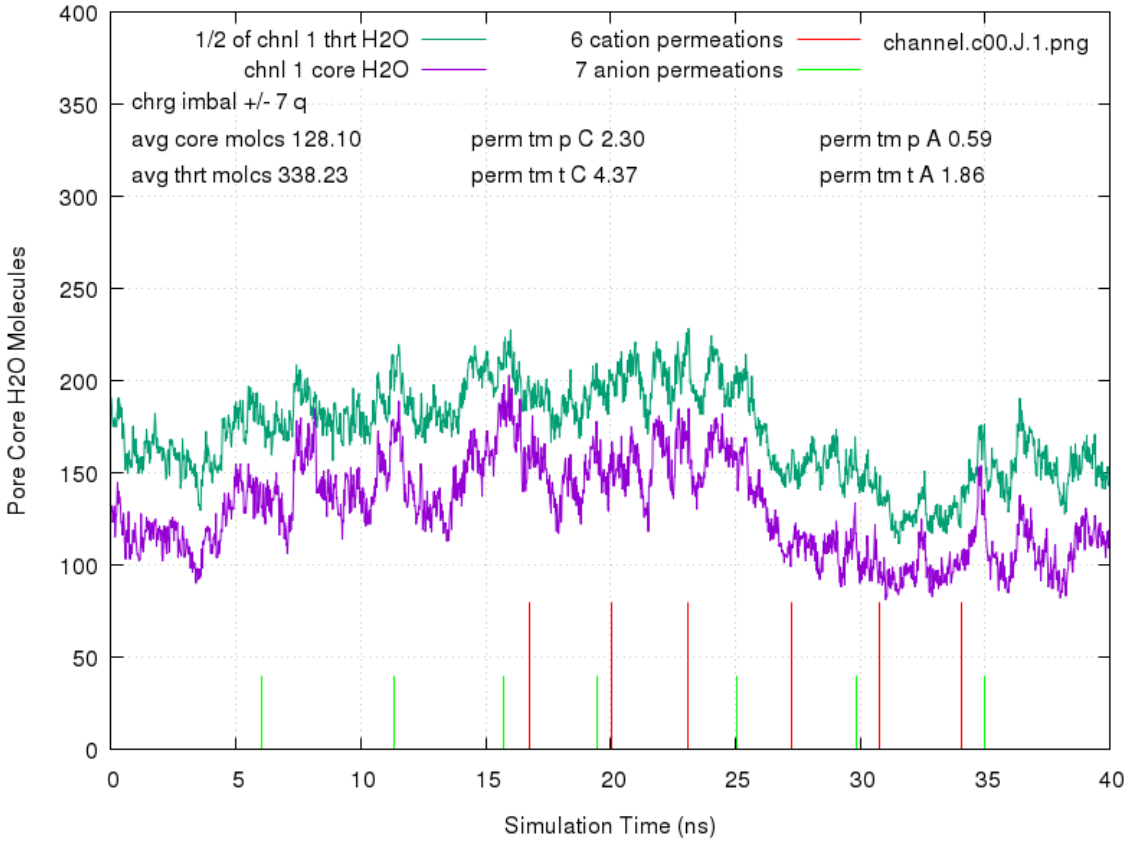

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
8 anion permeations
channel.c06.x.0.png
chrg imbal +/- 7 q
avg core molcs 125.84
avg thrt molcs 340.64
perm tm p C 1.55
perm tm t C 3.64
perm tm p A 0.57
perm tm t A 1.79
Pore Core H2O Molecules
Simulation Time (ns)

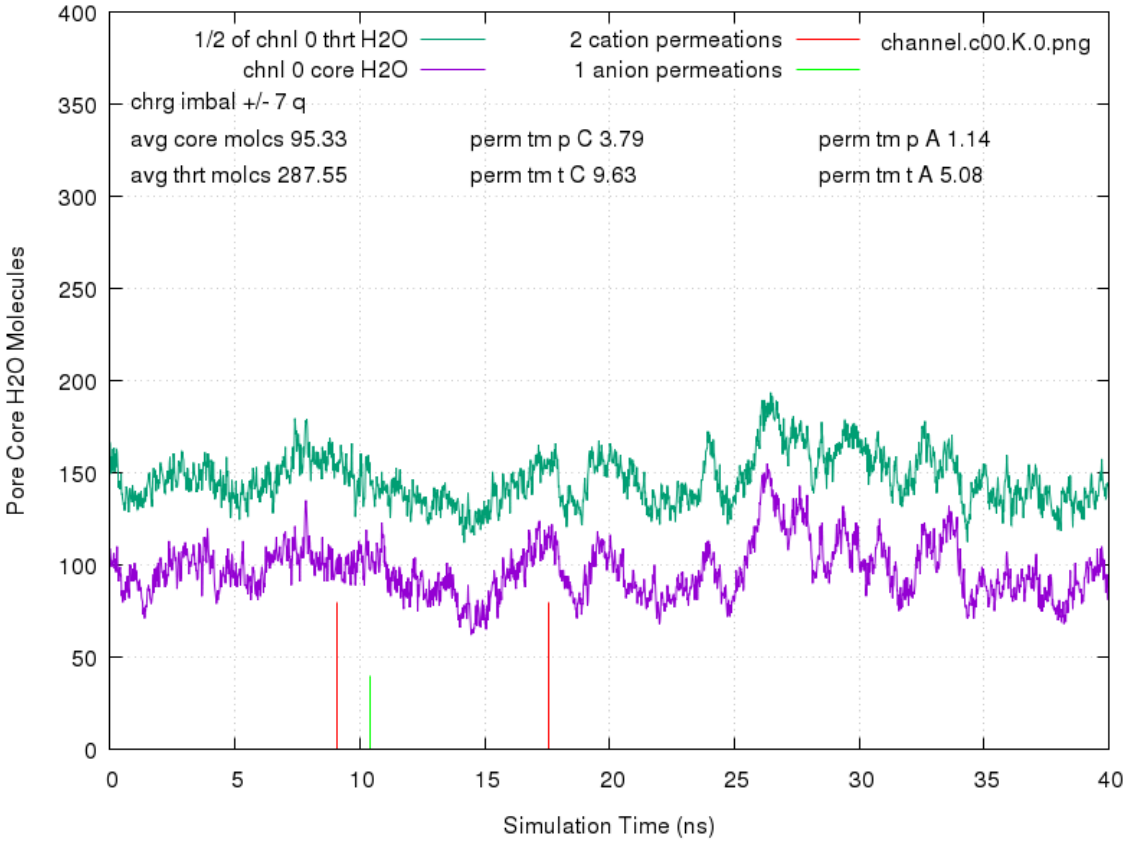

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
6 anion permeations
channel.c06.x.1.png
chrg imbal +/- 7 q
avg core molcs 108.42
avg thrt molcs 305.29
perm tm p C 6.60
perm tm t C 19.66
perm tm p A 0.75
perm tm t A 1.96
Pore Core H2O Molecules
Simulation Time (ns)

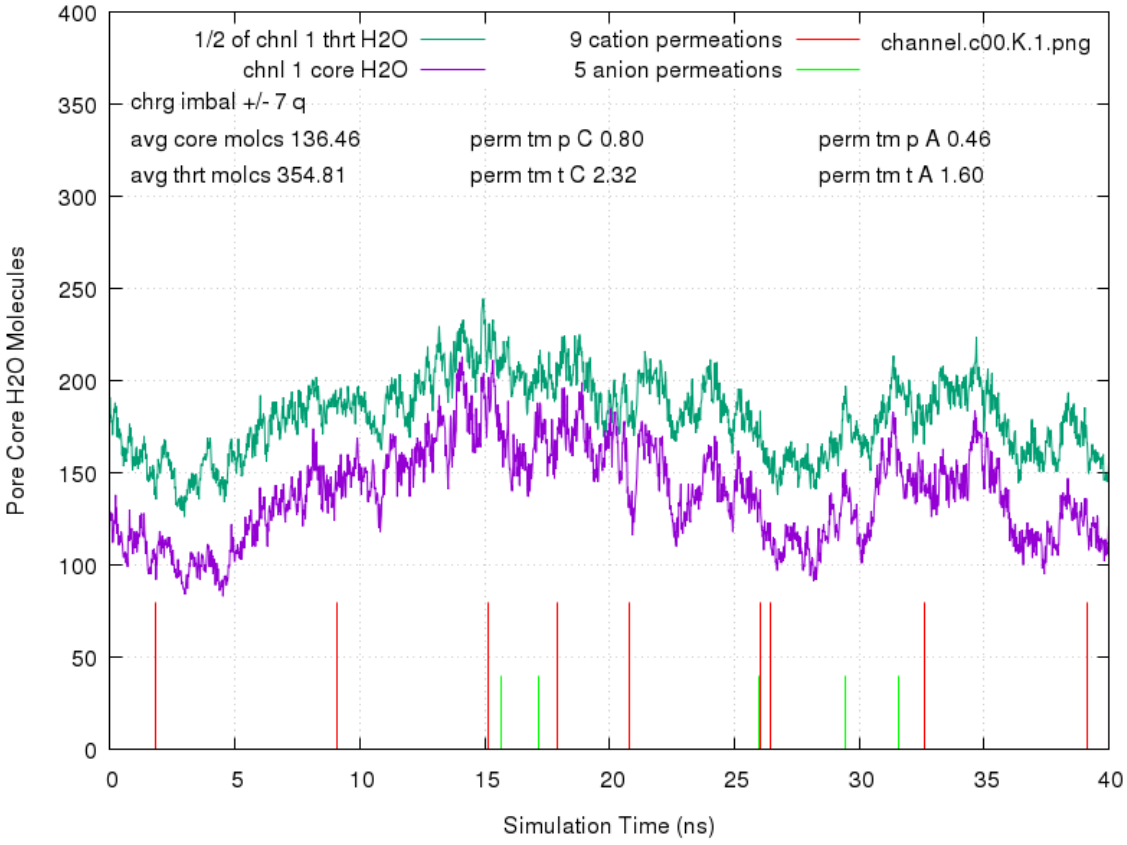

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
9 cation permeations
4 anion permeations
channel.c00.l.0.png
chrg imbal +/- 7 q
avg core molcs 121.71
avg thrt molcs 329.96
perm tm p C 2.13
perm tm t C 4.70
perm tm p A 0.58
perm tm t A 1.15
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

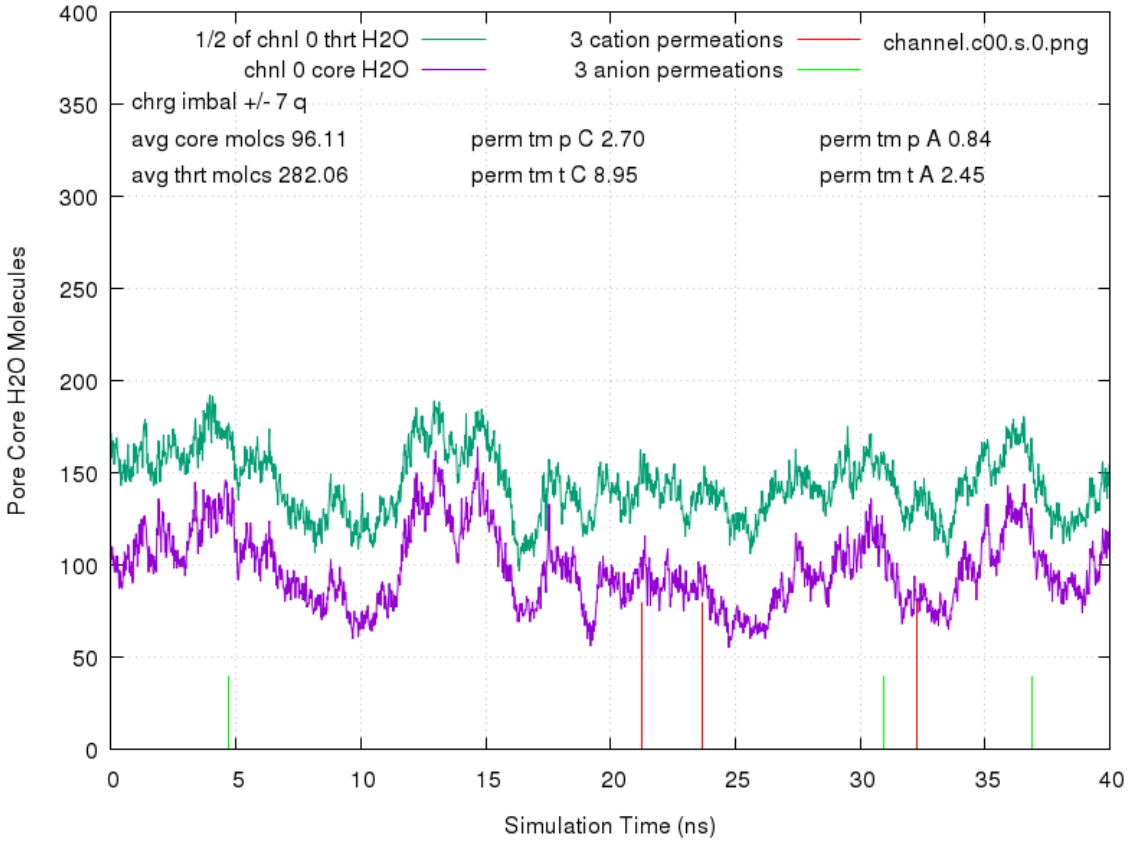

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
10 cation permeations
3 anion permeations
channel.c00.l.1.png
chrg imbal +/- 7 q
avg core molcs 128.07
avg thrt molcs 336.30
perm tm p C 1.79
perm tm t C 4.80
perm tm p A 0.41
perm tm t A 2.06
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

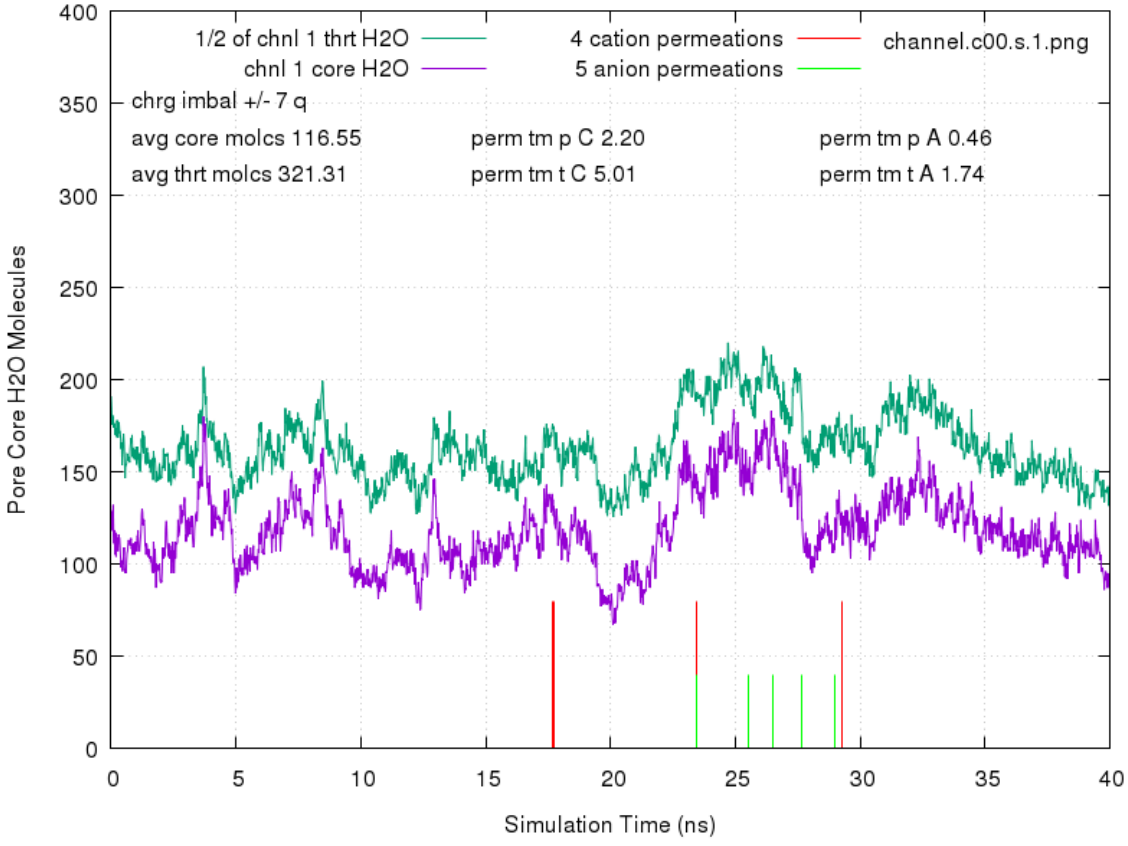

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
1 anion permeations
channel.c00.J.0.png
chrg imbal +/- 7 q
avg core molcs 105.22
avg thrt molcs 301.76
perm tm p C 2.78
perm tm t C 6.98
perm tm p A 0.84
perm tm t A 4.50
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

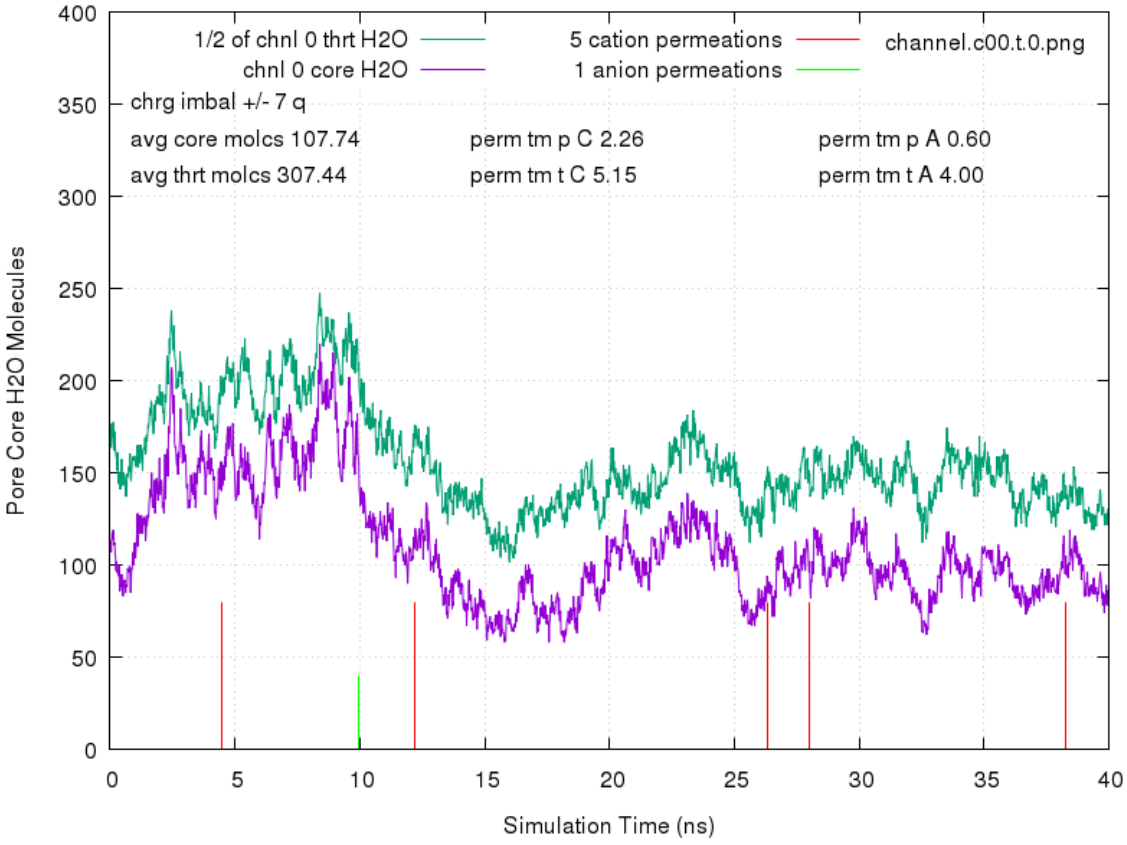

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
6 cation permeations
7 anion permeations
channel.c00.J.1.png
chrg imbal +/- 7 q
avg core molcs 128.10
avg thrt molcs 338.23
perm tm p C 2.30
perm tm t C 4.37
perm tm p A 0.59
perm tm t A 1.86
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

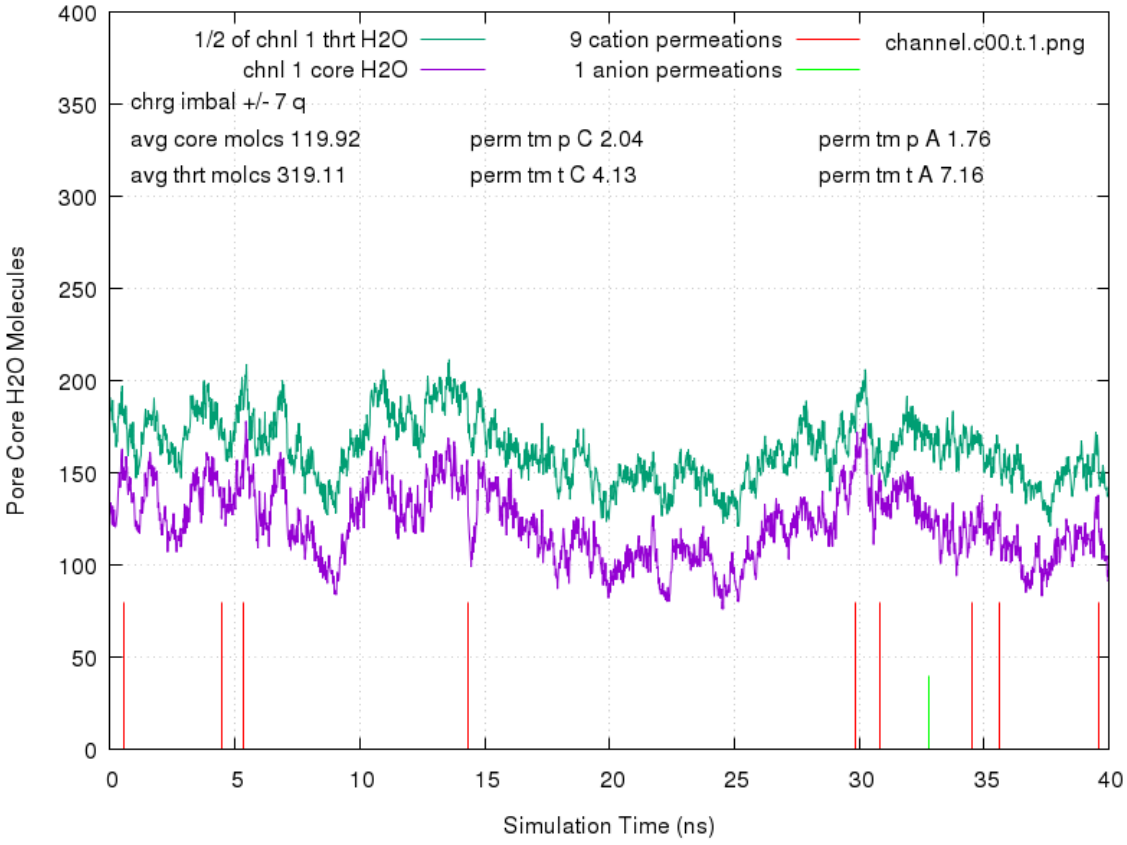

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
2 cation permeations
1 anion permeations
channel.c00.K.0.png
chrg imbal +/- 7 q
avg core molcs 95.33
avg thrt molcs 287.55
perm tm p C 3.79
perm tm t C 9.63
perm tm p A 1.14
perm tm t A 5.08
Pore Core H2O Molecules
Simulation Time (ns)

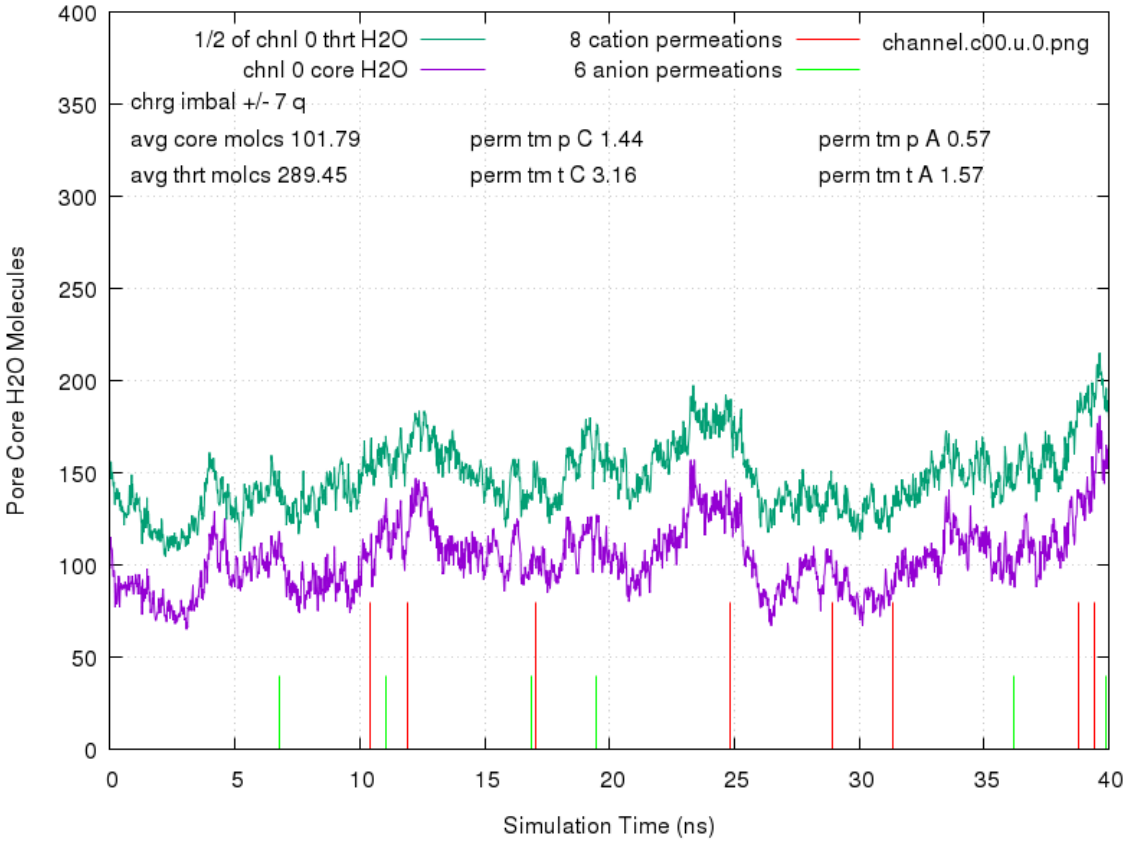

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
9 cation permeations
5 anion permeations
channel.c00.K.1.png
chrg imbal +/- 7 q
avg core molcs 136.46
avg thrt molcs 354.81
perm tm p C 0.80
perm tm t C 2.32
perm tm p A 0.46
perm tm t A 1.60
Pore Core H2O Molecules
Simulation Time (ns)

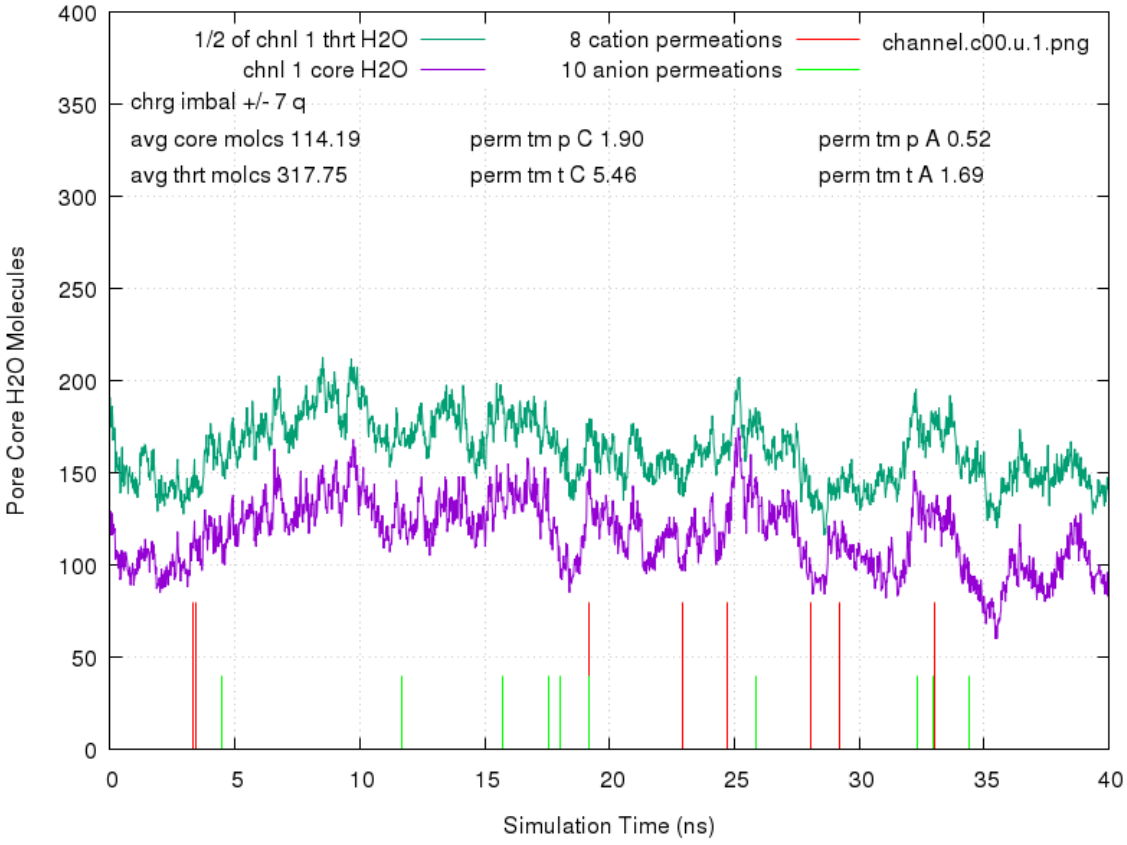

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
3 anion permeations
channel.c00.s.0.png
chrg imbal +/- 7 q
avg core molcs 96.11
avg thrt molcs 282.06
perm tm p C 2.70
perm tm t C 8.95
perm tm p A 0.84
perm tm t A 2.45
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

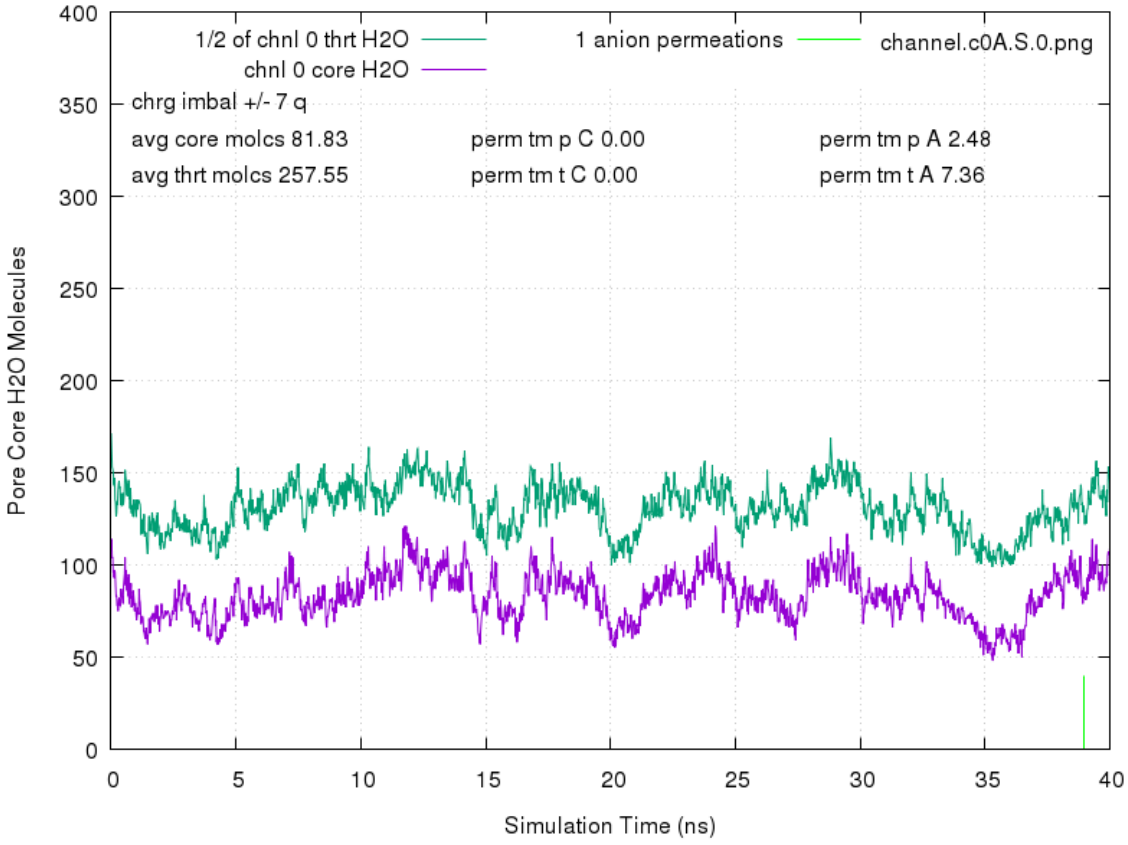

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
5 anion permeations
channel.c00.s.1.png
chrg imbal +/- 7 q
avg core molcs 116.55
avg thrt molcs 321.31
perm tm p C 2.20
perm tm t C 5.01
perm tm p A 0.46
perm tm t A 1.74
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

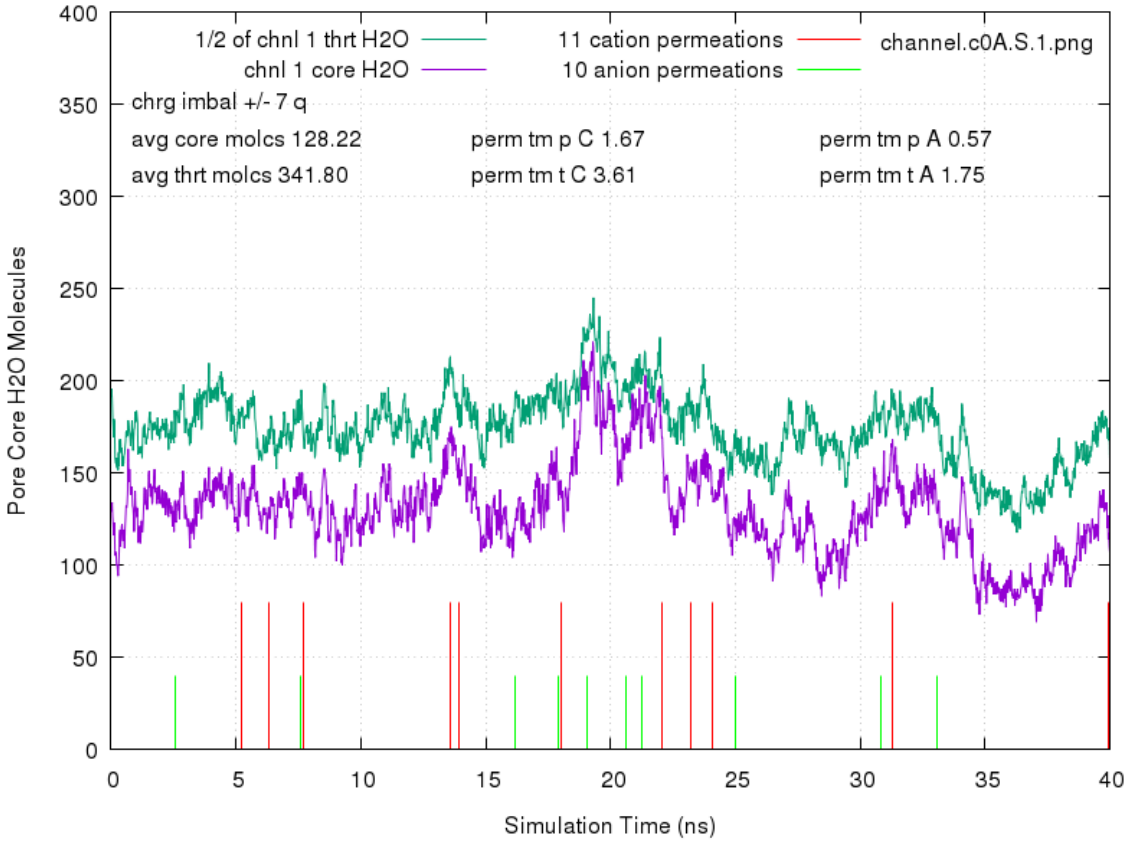

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
1 anion permeations
channel.c00.t.0.png
chrg imbal +/- 7 q
avg core molcs 107.74
avg thrt molcs 307.44
perm tm p C 2.26
perm tm t C 5.15
perm tm p A 0.60
perm tm t A 4.00
Pore Core H2O Molecules
Simulation Time (ns)

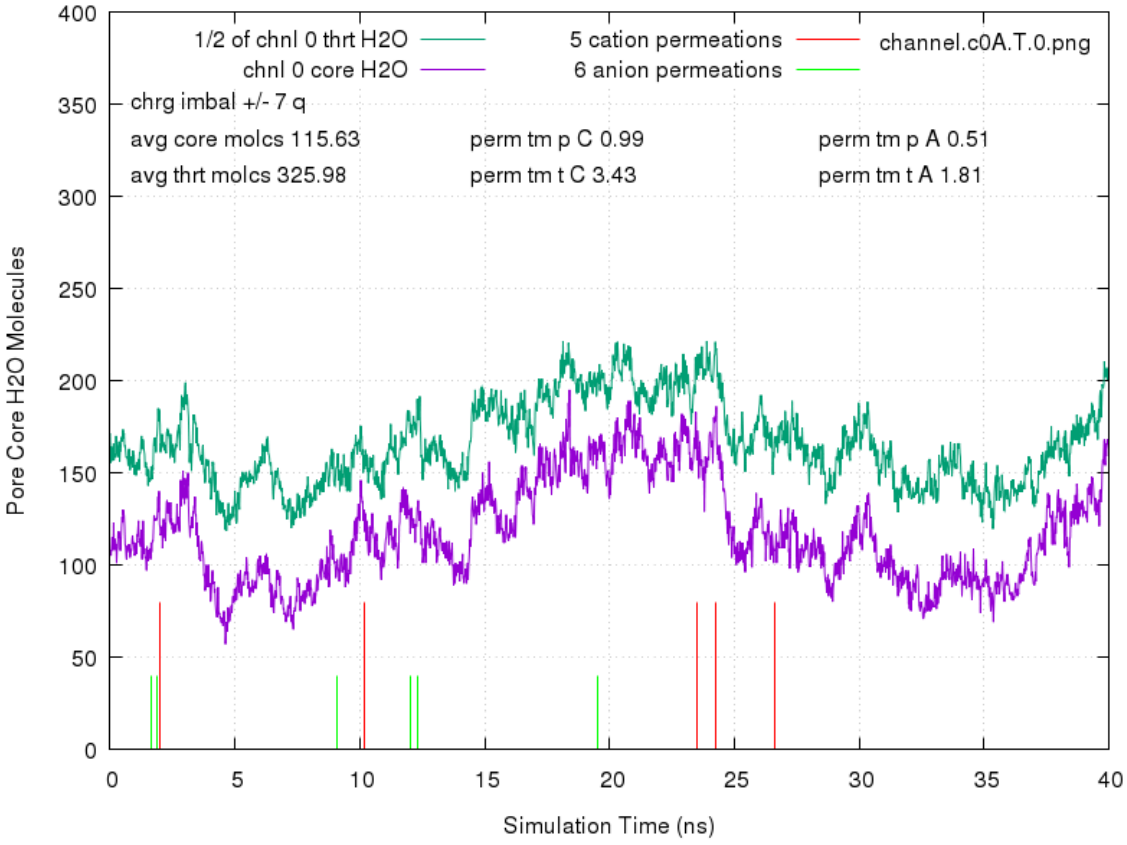

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
9 cation permeations
1 anion permeations
channel.c00.t.1.png
chrg imbal +/- 7 q
avg core molcs 119.92
avg thrt molcs 319.11
perm tm p C 2.04
perm tm t C 4.13
perm tm p A 1.76
perm tm t A 7.16
Pore Core H2O Molecules
Simulation Time (ns)

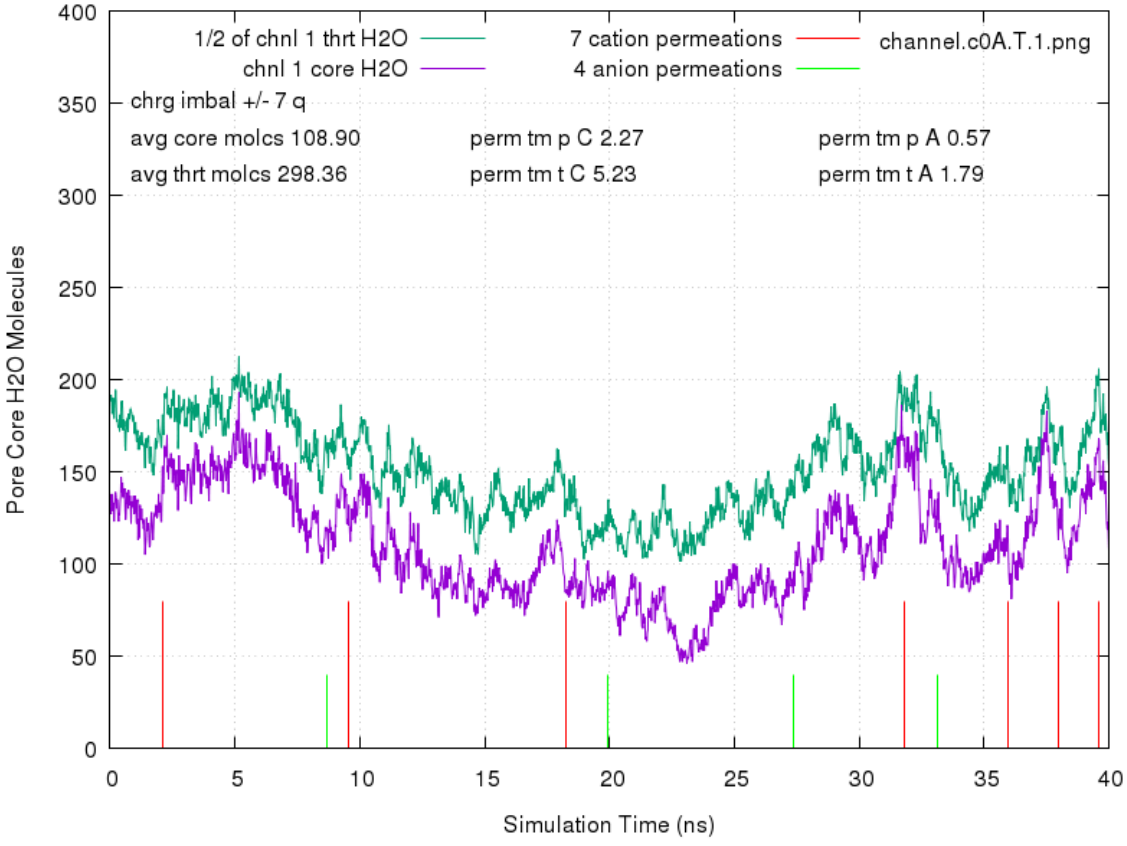

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
6 anion permeations
channel.c00.u.0.png
chrg imbal +/- 7 q
avg core molcs 101.79
avg thrt molcs 289.45
perm tm p C 1.44
perm tm t C 3.16
perm tm p A 0.57
perm tm t A 1.57
Pore Core H2O Molecules
Simulation Time (ns)

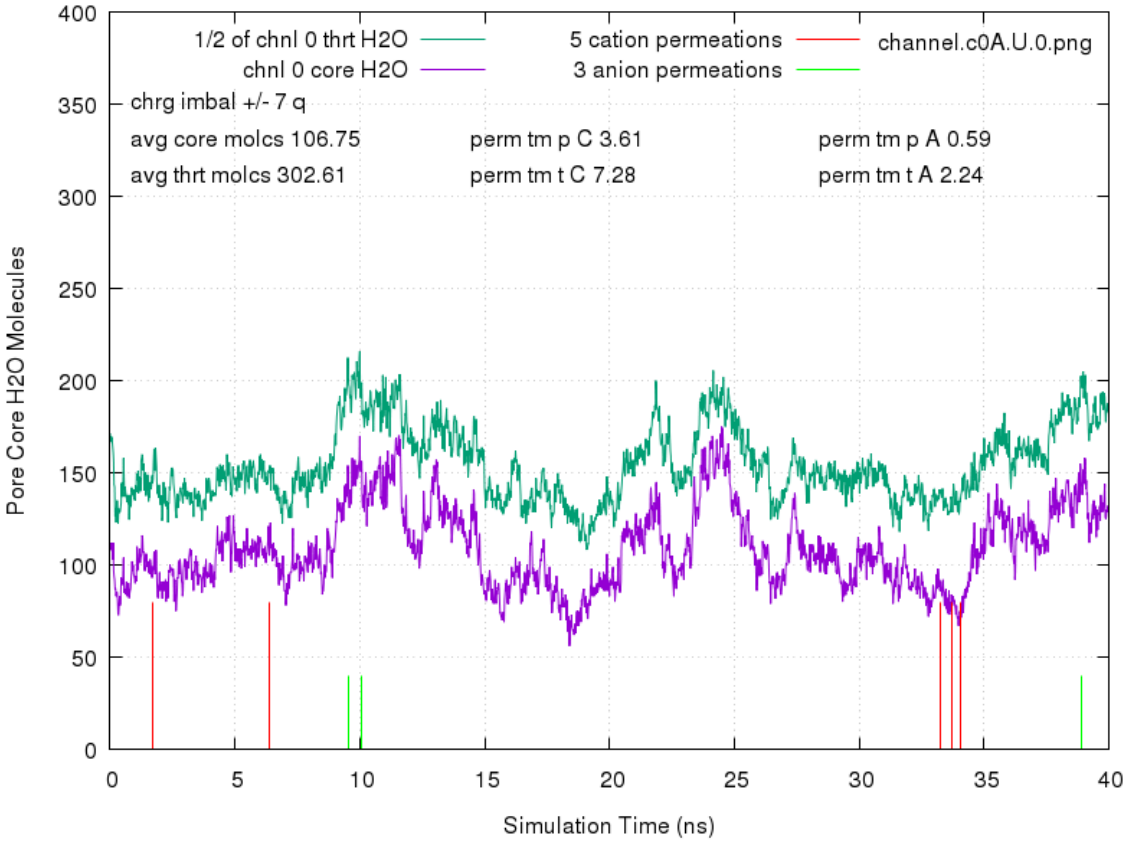

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
8 cation permeations
10 anion permeations
channel.c00.u.1.png
chrg imbal +/- 7 q
avg core molcs 114.19
avg thrt molcs 317.75
perm tm p C 1.90
perm tm t C 5.46
perm tm p A 0.52
perm tm t A 1.69
Pore Core H2O Molecules
Simulation Time (ns)

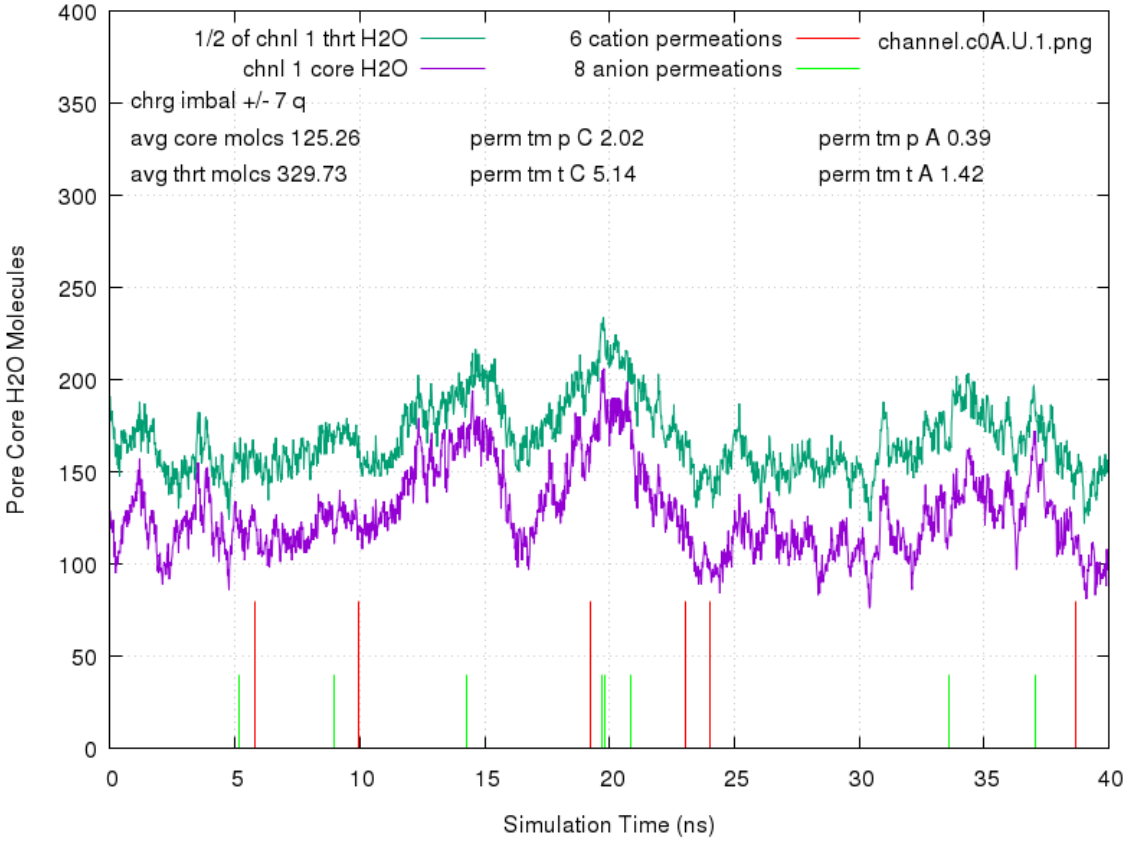

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 anion permeations
channel.c0A.S.0.png
chrg imbal +/- 7 q
avg core molcs 81.83
avg thrt molcs 257.55
perm tm p C 0.00
perm tm t C 0.00
perm tm p A 2.48
perm tm t A 7.36
Pore Core H2O Molecules
Simulation Time (ns)

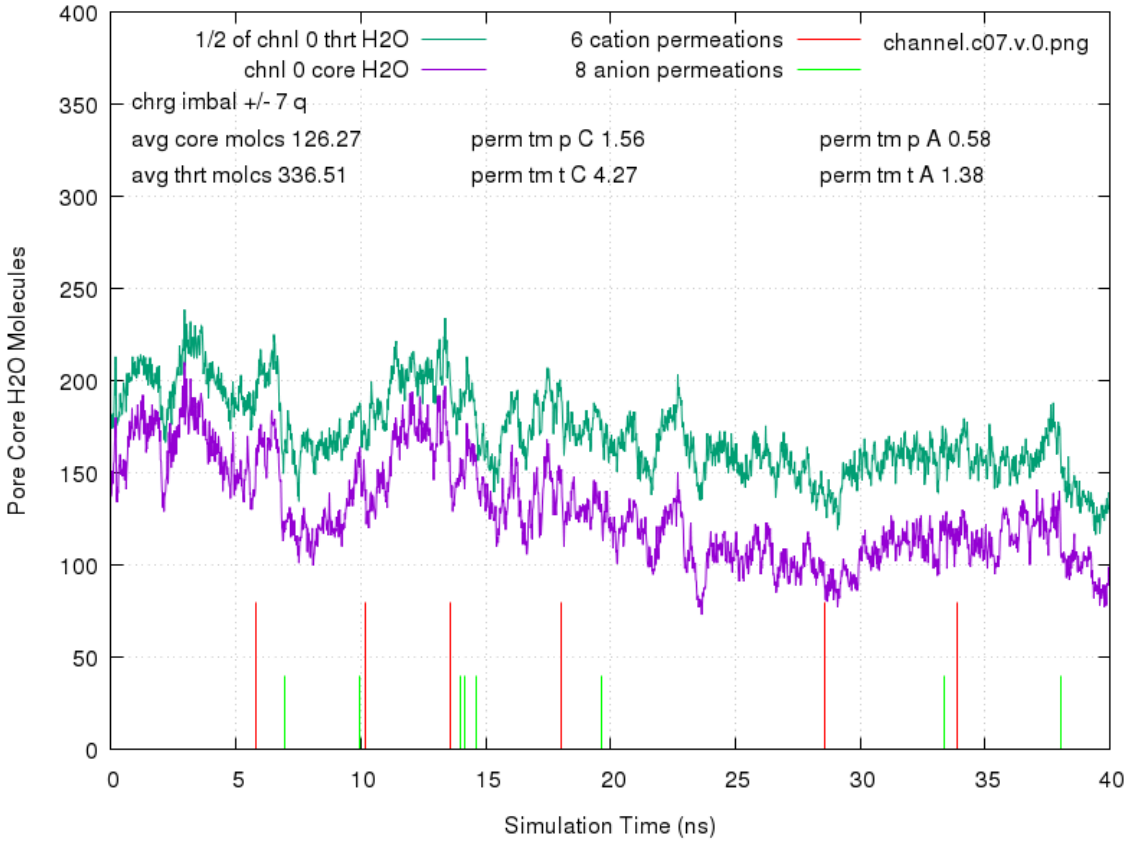

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
11 cation permeations
10 anion permeations
channel.c0A.S.1.png
chrg imbal +/- 7 q
avg core molcs 128.22
avg thrt molcs 341.80
perm tm p C 1.67
perm tm t C 3.61
perm tm p A 0.57
perm tm t A 1.75
Pore Core H2O Molecules
Simulation Time (ns)

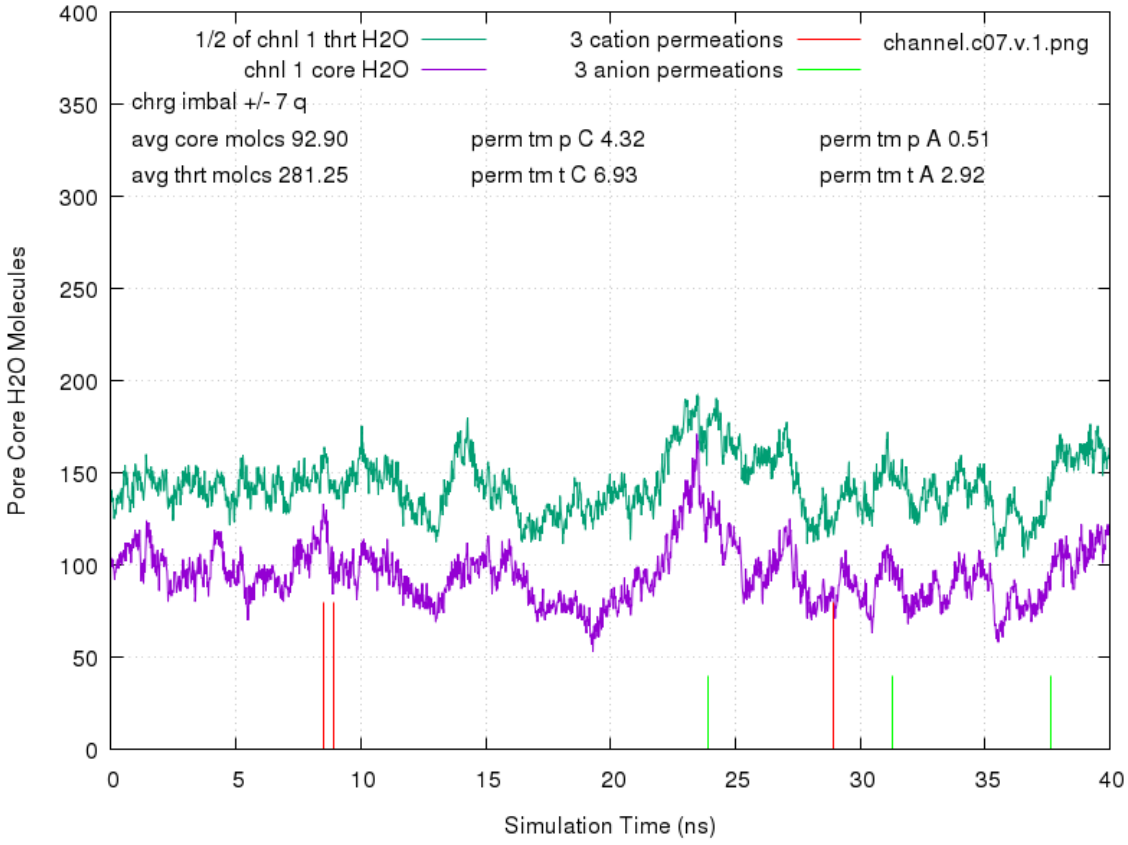

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
6 anion permeations
channel.c0A.T.0.png
chrg imbal +/- 7 q
avg core molcs 115.63
avg thrt molcs 325.98
perm tm p C 0.99
perm tm t C 3.43
perm tm p A 0.51
perm tm t A 1.81
Pore Core H2O Molecules
Simulation Time (ns)

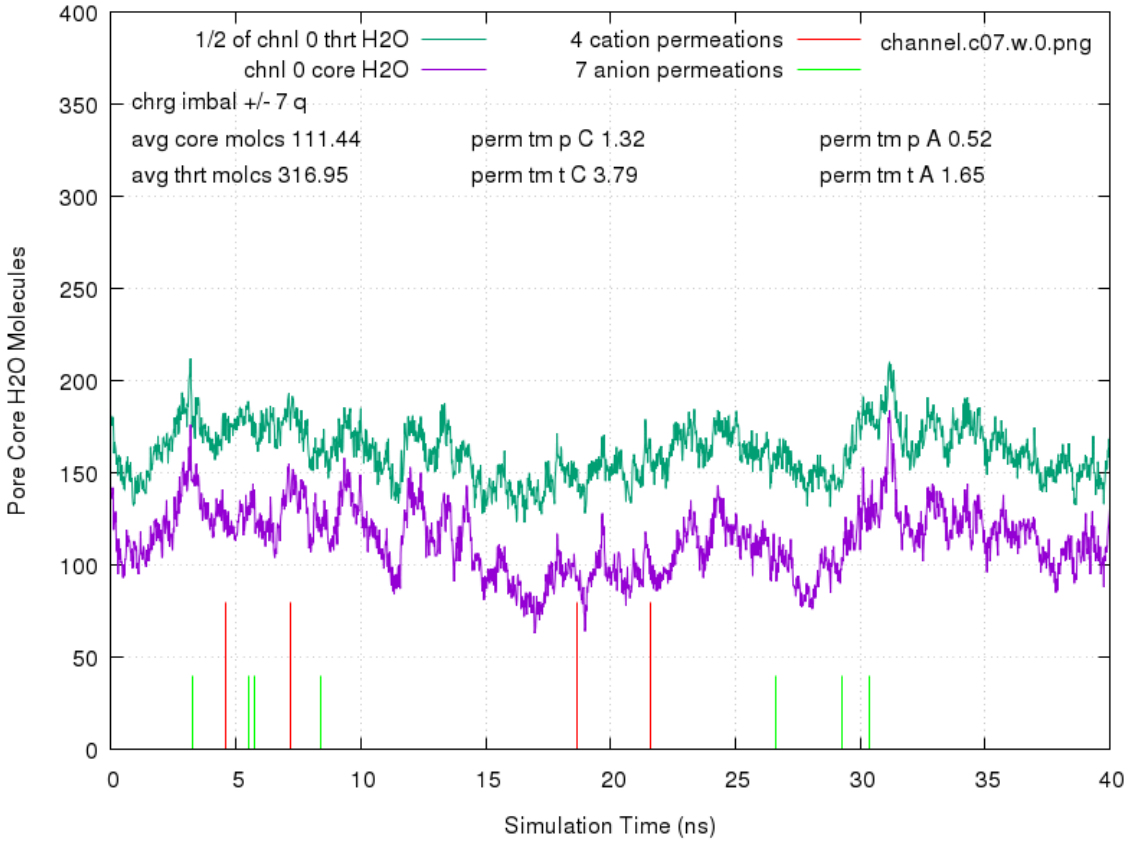

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
7 cation permeations
4 anion permeations
channel.c0A.T.1.png
chrg imbal +/- 7 q
avg core molcs 108.90
avg thrt molcs 298.36
perm tm p C 2.27
perm tm t C 5.23
perm tm p A 0.57
perm tm t A 1.79
Pore Core H2O Molecules
Simulation Time (ns)

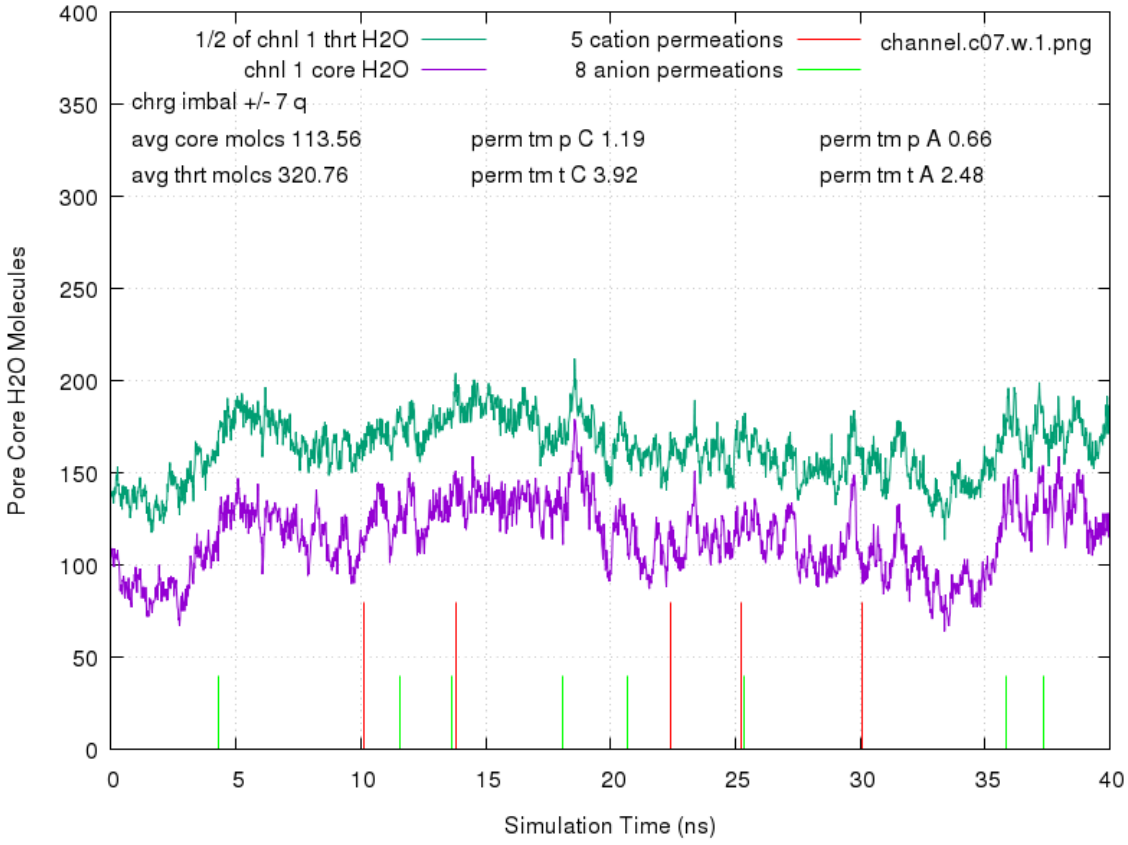

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
3 anion permeations
channel.c0A.U.0.png
chrg imbal +/- 7 q
avg core molcs 106.75
avg thrt molcs 302.61
perm tm p C 3.61
perm tm t C 7.28
perm tm p A 0.59
perm tm t A 2.24
Pore Core H2O Molecules
Simulation Time (ns)

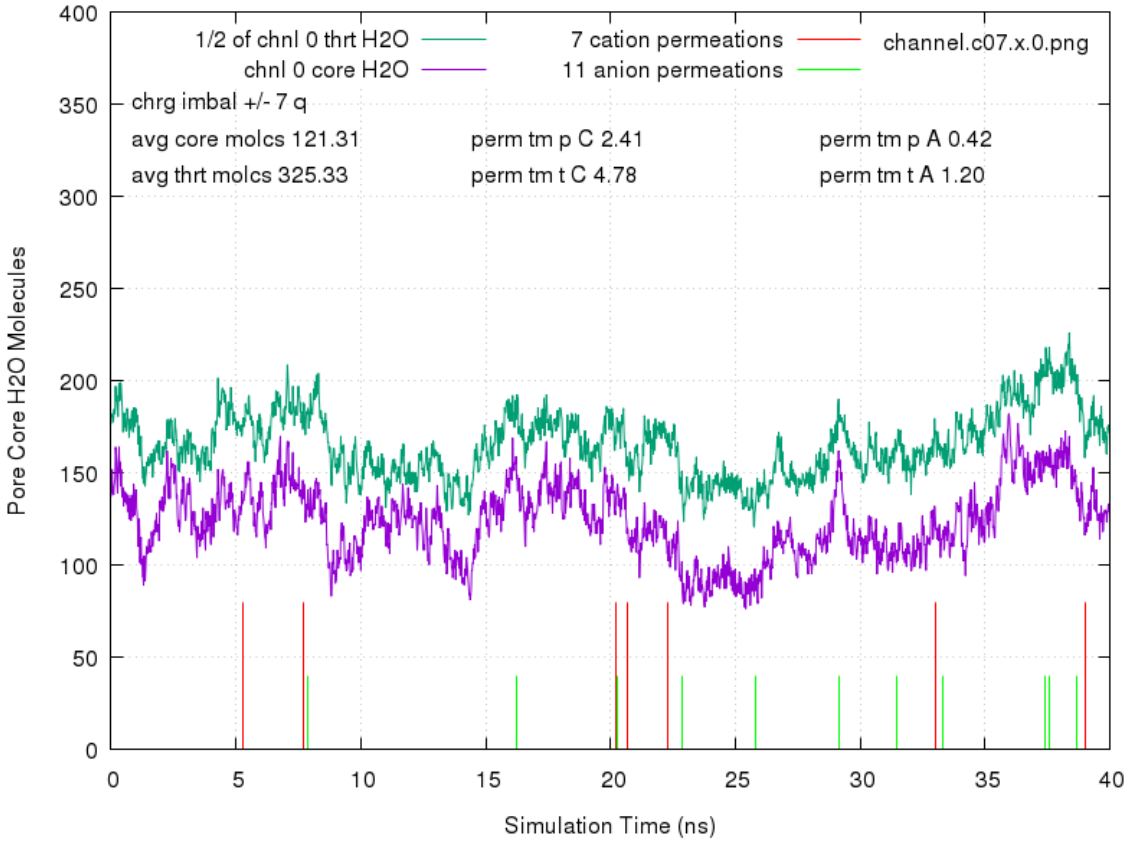

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
6 cation permeations
8 anion permeations
channel.c0A.U.1.png
chrg imbal +/- 7 q
avg core molcs 125.26
avg thrt molcs 329.73
perm tm p C 2.02
perm tm t C 5.14
perm tm p A 0.39
perm tm t A 1.42
Pore Core H2O Molecules
Simulation Time (ns)

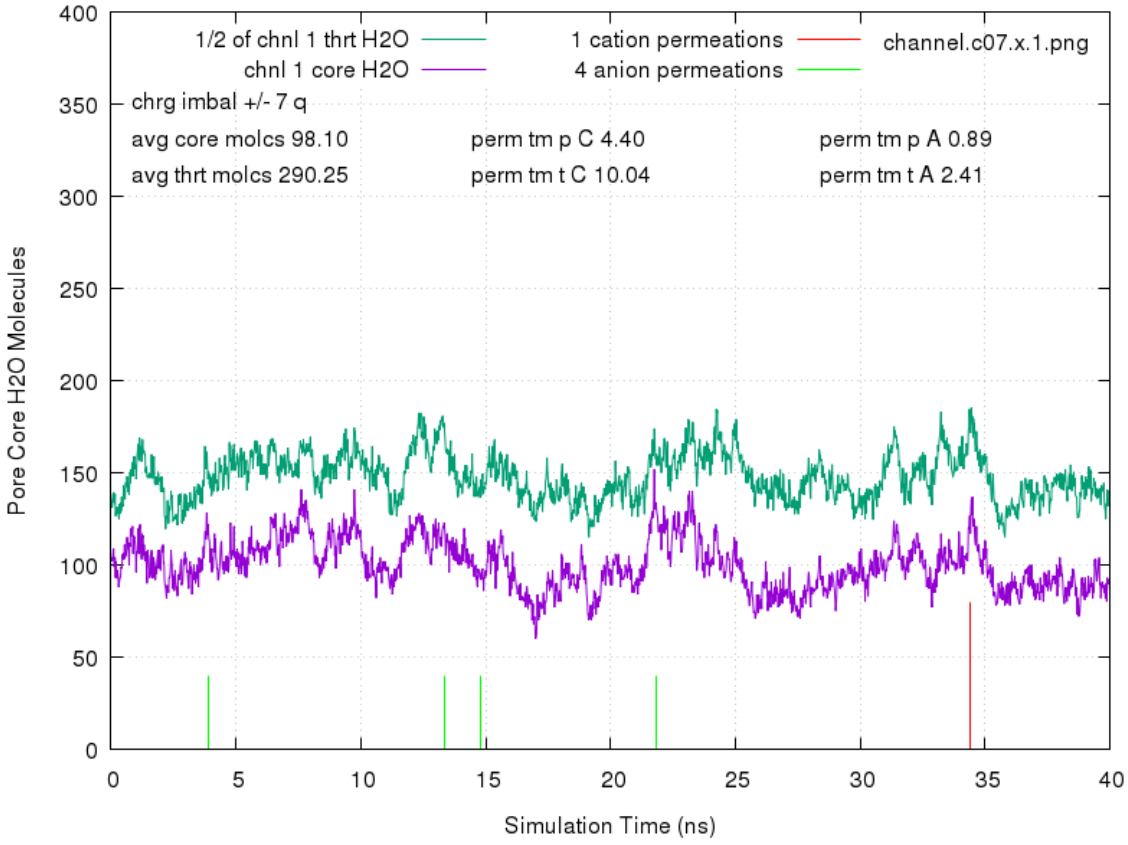

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
6 cation permeations
8 anion permeations
channel.c07.v.0.png
chrg imbal +/- 7 q
avg core molcs 126.27
avg thrt molcs 336.51
perm tm p C 1.56
perm tm t C 4.27
perm tm p A 0.58
perm tm t A 1.38
Pore Core H2O Molecules
Simulation Time (ns)

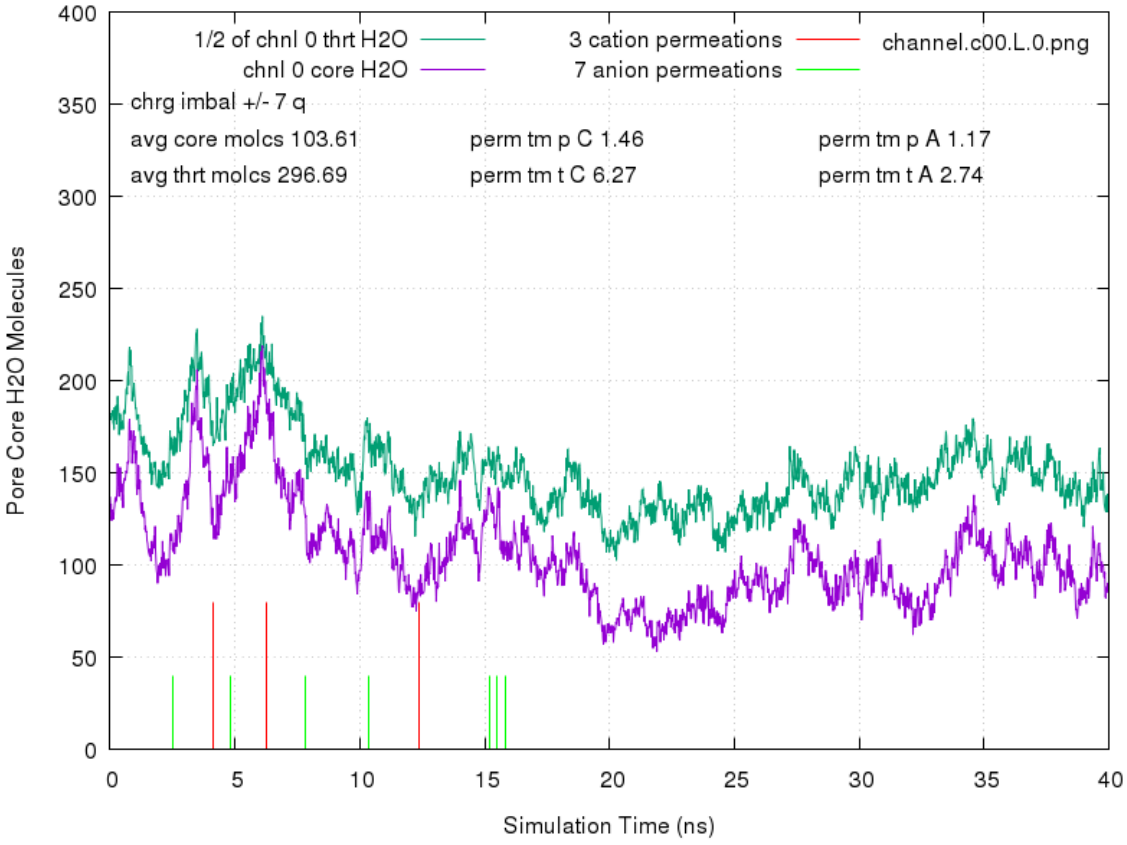

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
3 anion permeations
channel.c07.v.1.png
chrg imbal +/- 7 q
avg core molcs 92.90
avg thrt molcs 281.25
perm tm p C 4.32
perm tm t C 6.93
perm tm p A 0.51
perm tm t A 2.92
Pore Core H2O Molecules
Simulation Time (ns)

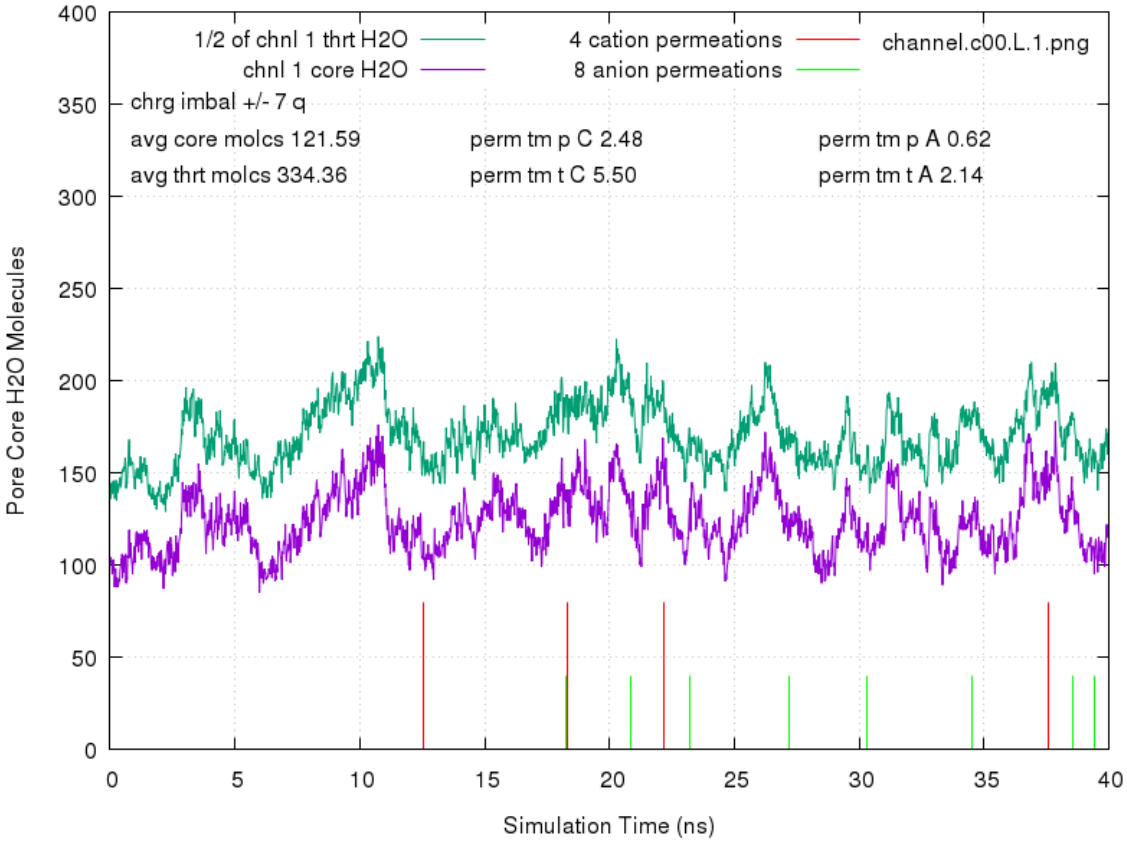

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
4 cation permeations
7 anion permeations
channel.c07.w.0.png
chrg imbal +/- 7 q
avg core molcs 111.44
avg thrt molcs 316.95
perm tm p C 1.32
perm tm t C 3.79
perm tm p A 0.52
perm tm t A 1.65
Pore Core H2O Molecules
Simulation Time (ns)

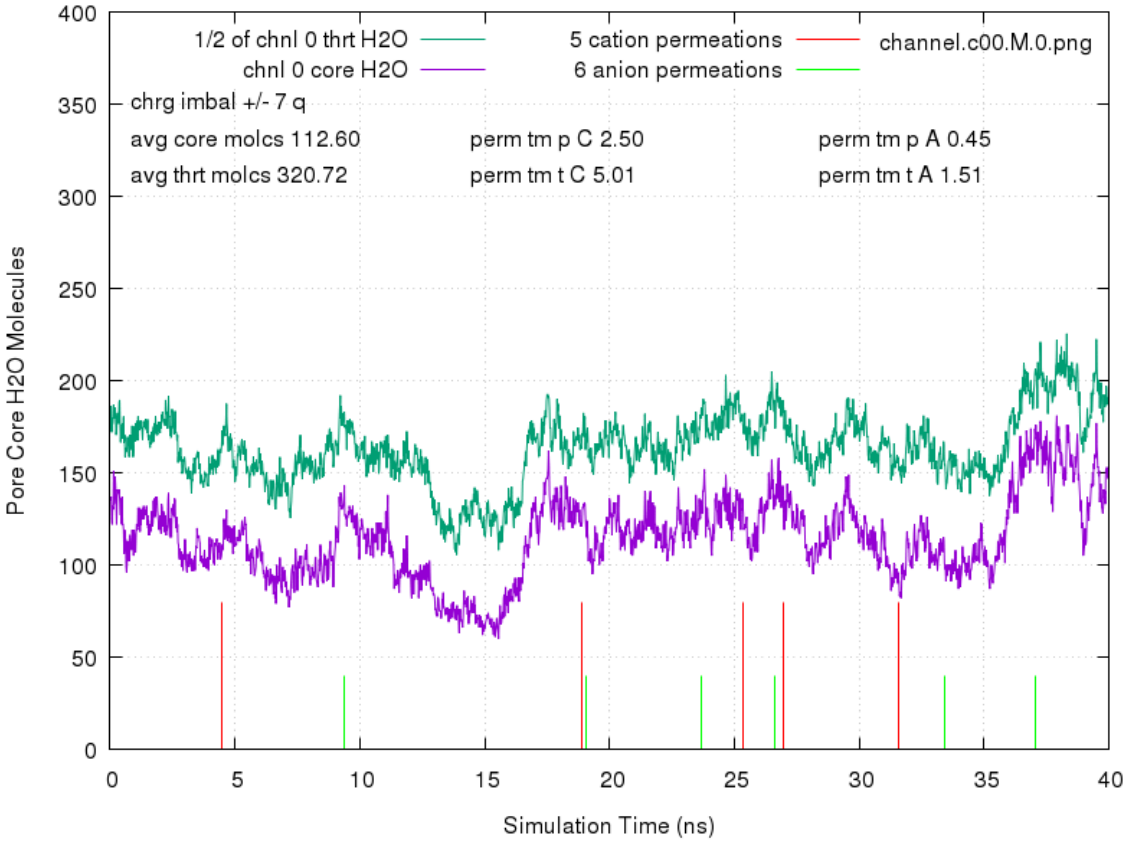

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
5 cation permeations
8 anion permeations
channel.c07.w.1.png
chrg imbal +/- 7 q
avg core molcs 113.56
avg thrt molcs 320.76
perm tm p C 1.19
perm tm t C 3.92
perm tm p A 0.66
perm tm t A 2.48
Pore Core H2O Molecules
Simulation Time (ns)

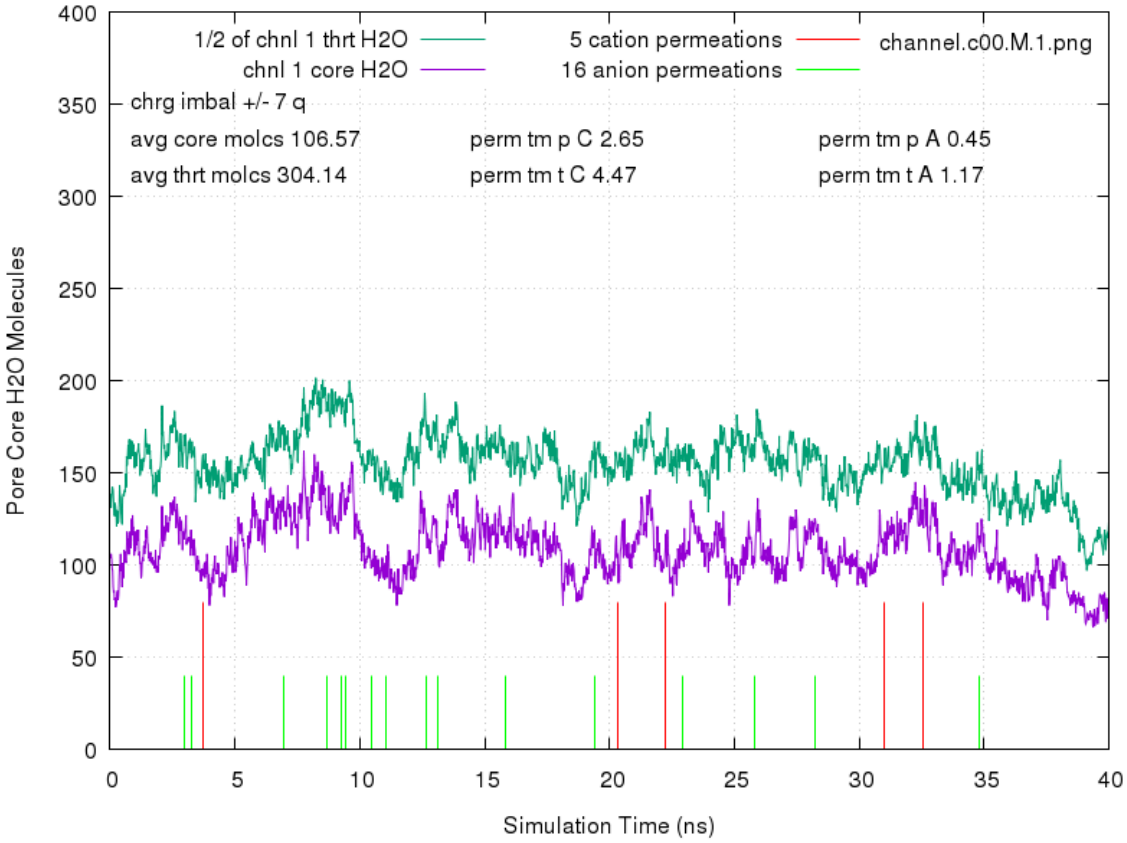

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
7 cation permeations
11 anion permeations
channel.c07.x.0.png
chrg imbal +/- 7 q
avg core molcs 121.31
avg thrt molcs 325.33
perm tm p C 2.41
perm tm t C 4.78
perm tm p A 0.42
perm tm t A 1.20
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

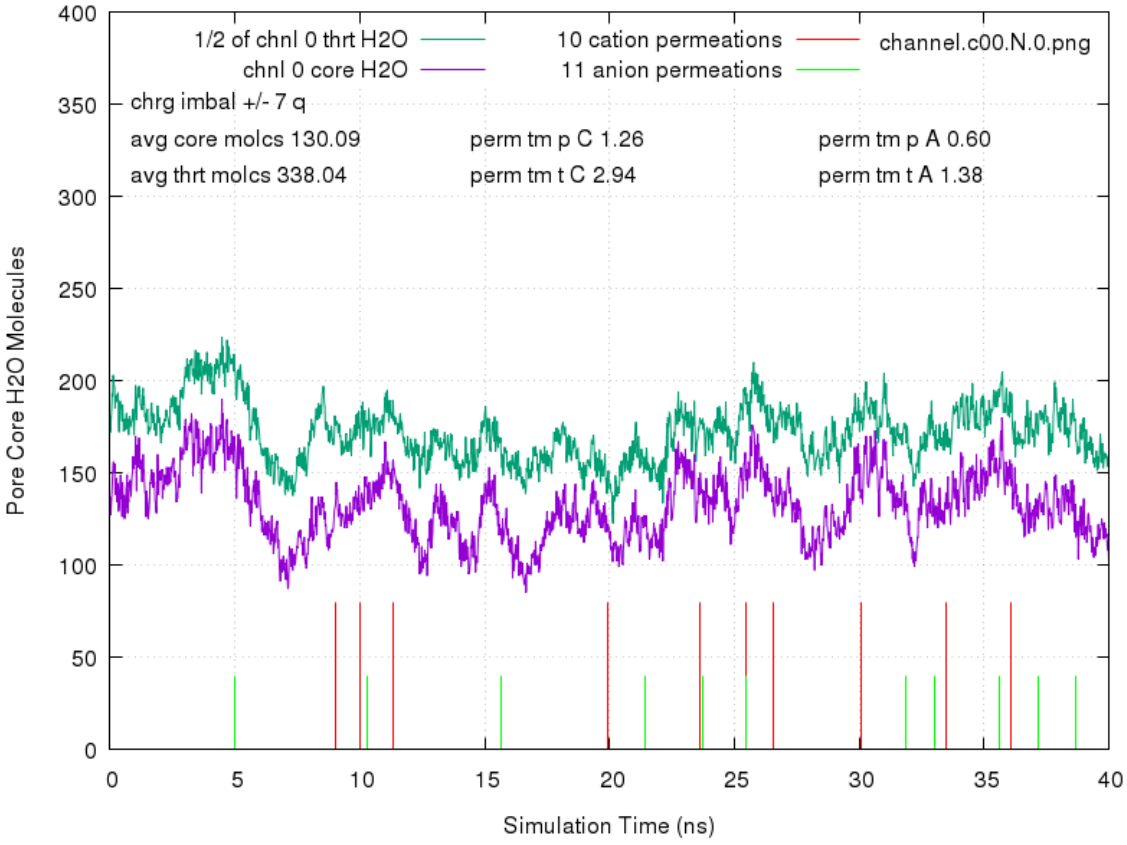

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
1 cation permeations
4 anion permeations
channel.c07.x.1.png
chrg imbal +/- 7 q
avg core molcs 98.10
avg thrt molcs 290.25
perm tm p C 4.40
perm tm t C 10.04
perm tm p A 0.89
perm tm t A 2.41
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

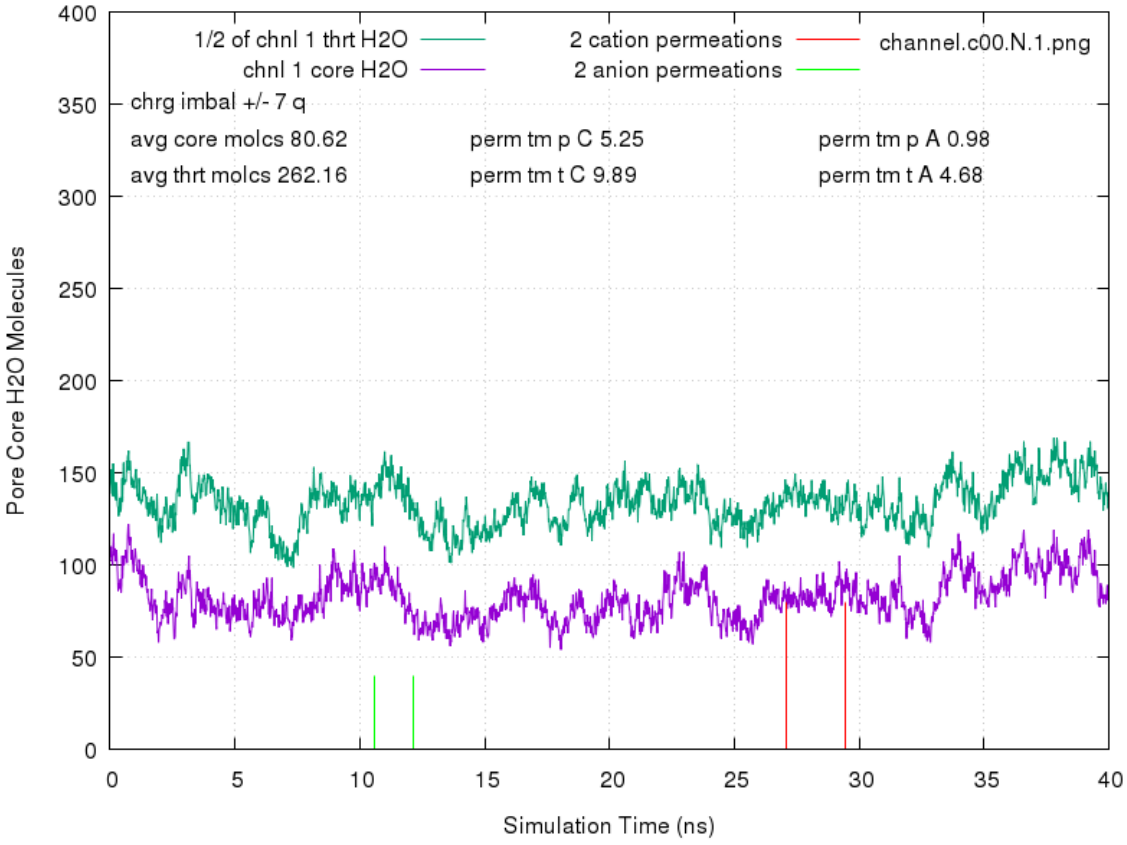

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
7 anion permeations
channel.c00.L.0.png
chrg imbal +/- 7 q
avg core molcs 103.61
avg thrt molcs 296.69
perm tm p C 1.46
perm tm t C 6.27
perm tm p A 1.17
perm tm t A 2.74
Pore Core H2O Molecules
Simulation Time (ns)

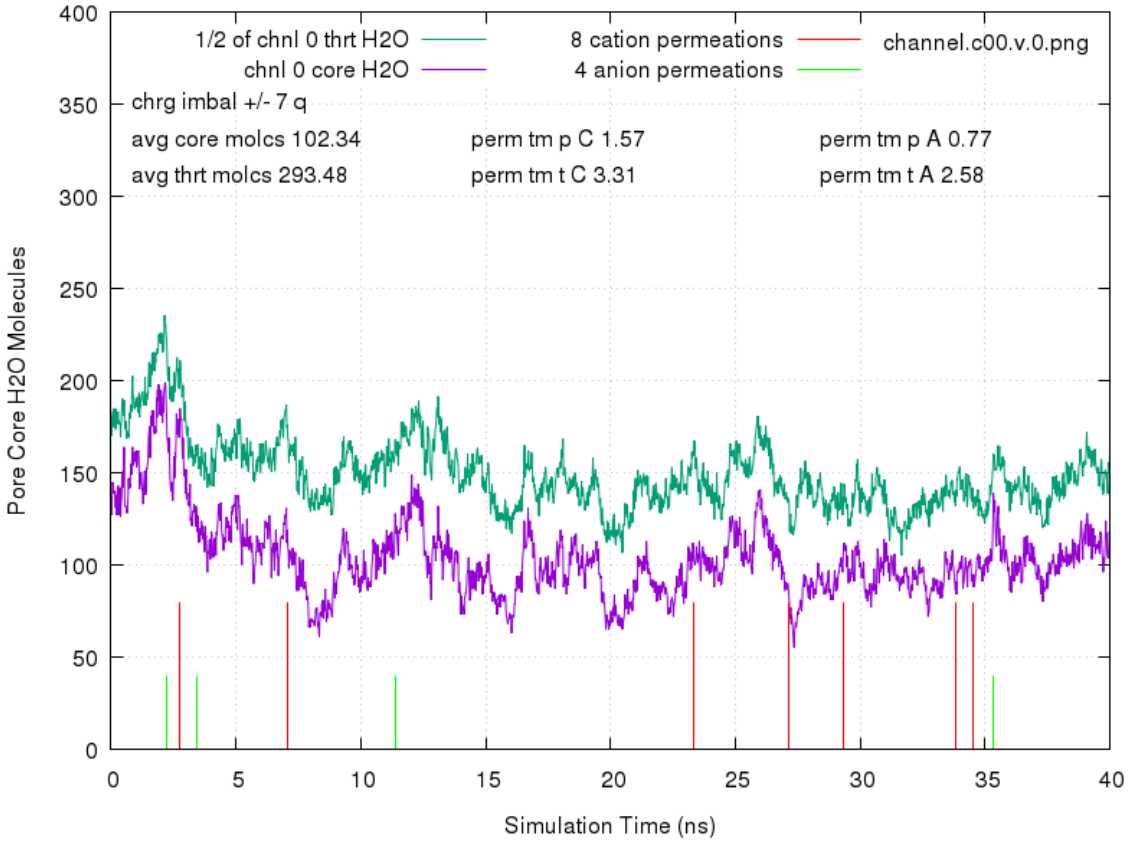

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
8 anion permeations
channel.c00.L.1.png
chrg imbal +/- 7 q
avg core molcs 121.59
avg thrt molcs 334.36
perm tm p C 2.48
perm tm t C 5.50
perm tm p A 0.62
perm tm t A 2.14
Pore Core H2O Molecules
Simulation Time (ns)

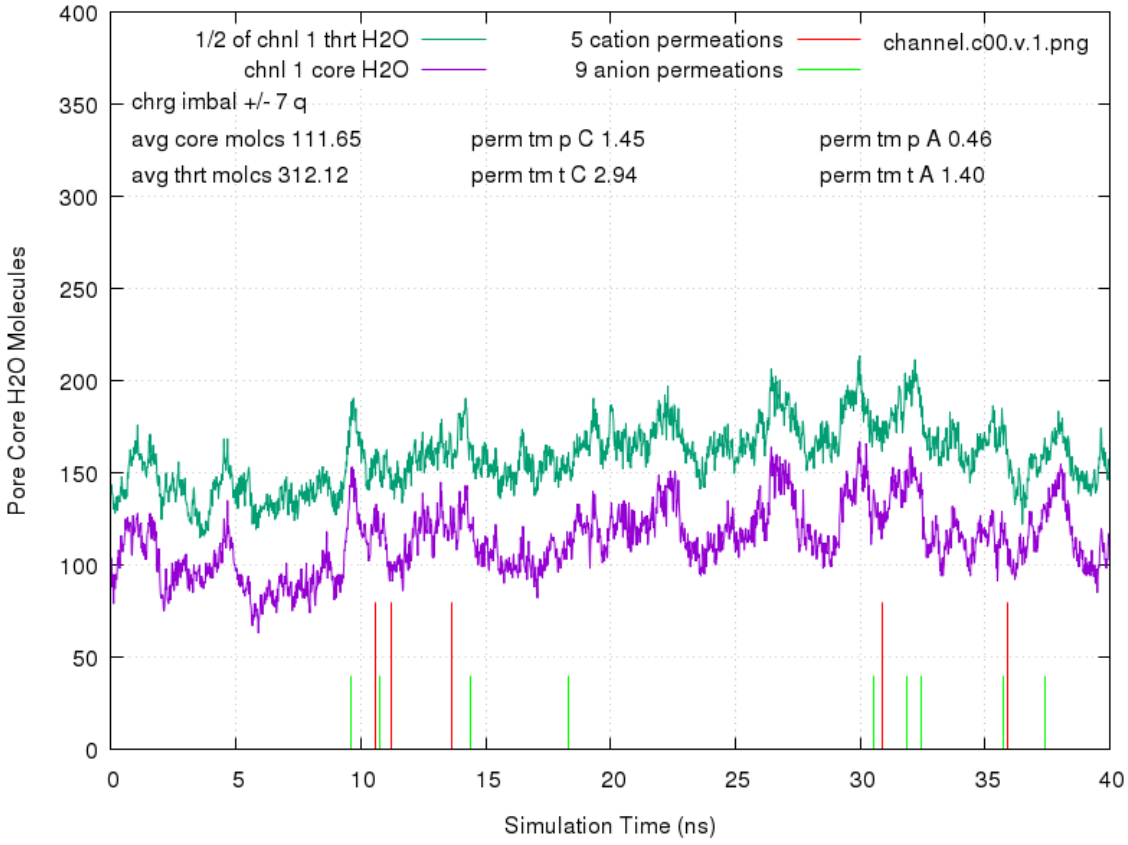

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
6 anion permeations
channel.c00.M.0.png
chrg imbal +/- 7 q
avg core molcs 112.60
avg thrt molcs 320.72
perm tm p C 2.50
perm tm t C 5.01
perm tm p A 0.45
perm tm t A 1.51
Pore Core H2O Molecules
Simulation Time (ns)

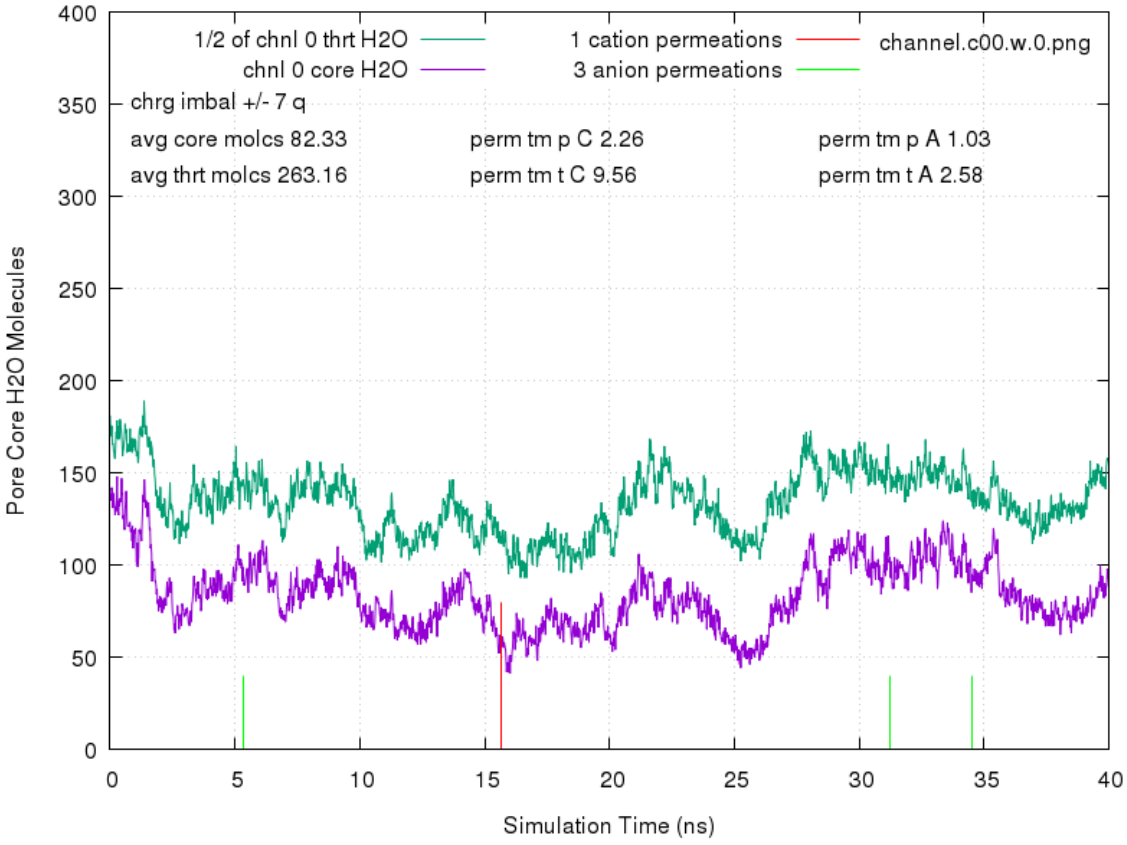

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
5 cation permeations
16 anion permeations
channel.c00.M.1.png
chrg imbal +/- 7 q
avg core molcs 106.57
avg thrt molcs 304.14
perm tm p C 2.65
perm tm t C 4.47
perm tm p A 0.45
perm tm t A 1.17
Pore Core H2O Molecules
Simulation Time (ns)

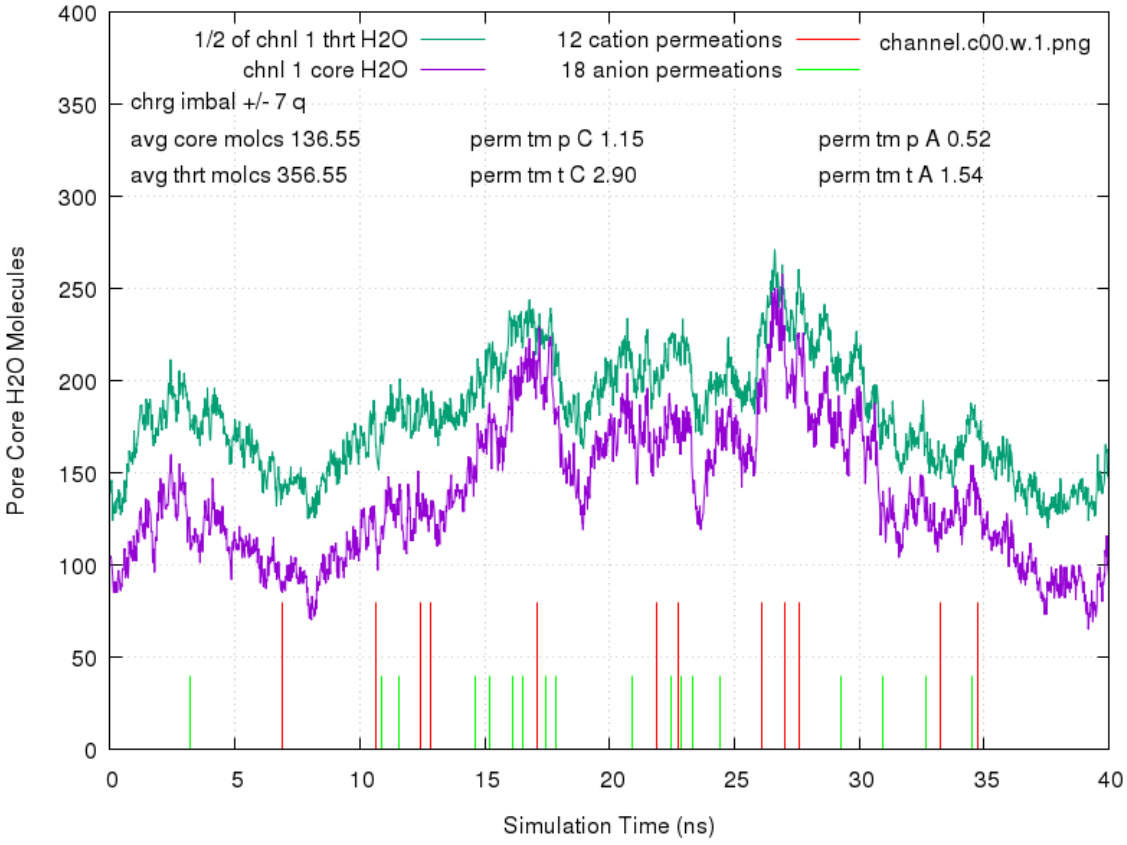

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
10 cation permeations
11 anion permeations
channel.c00.N.0.png
chrg imbal +/- 7 q
avg core molcs 130.09
avg thrt molcs 338.04
perm tm p C 1.26
perm tm t C 2.94
perm tm p A 0.60
perm tm t A 1.38
Pore Core H2O Molecules
Simulation Time (ns)

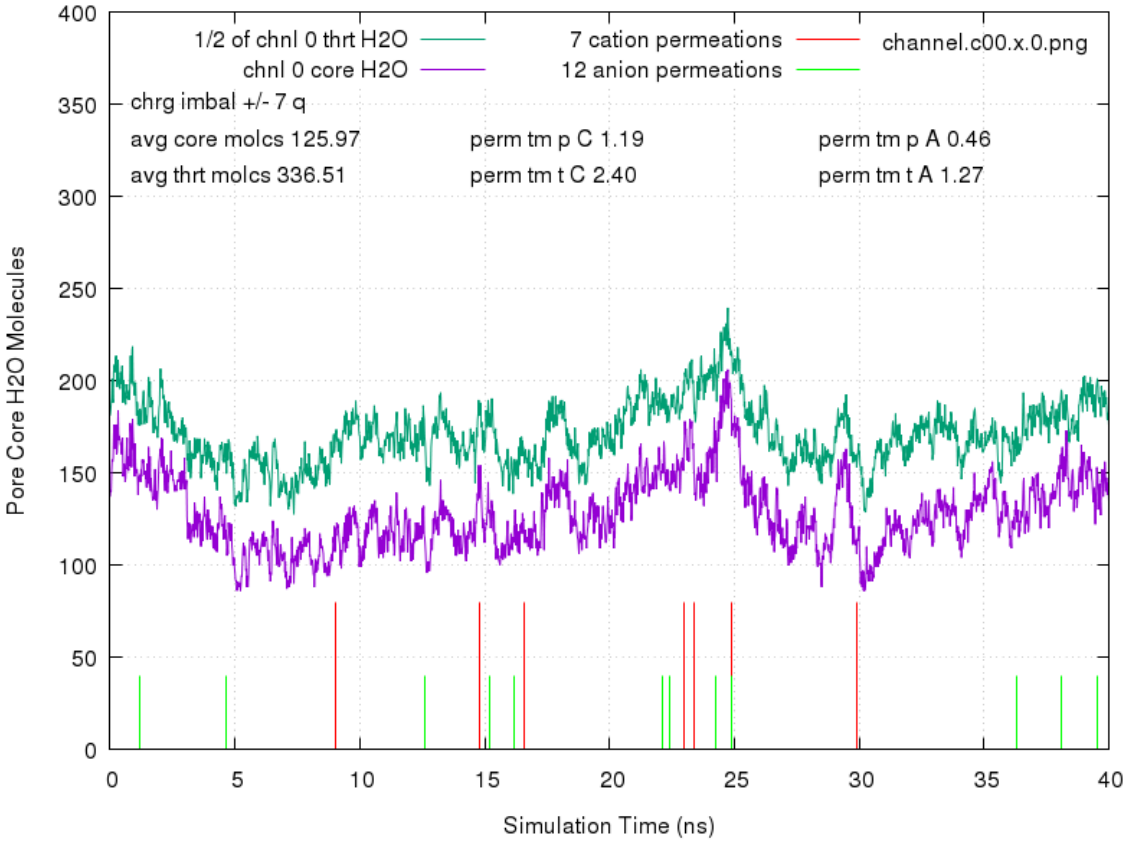

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
2 anion permeations
channel.c00.N.1.png
chrg imbal +/- 7 q
avg core molcs 80.62
avg thrt molcs 262.16
perm tm p C 5.25
perm tm t C 9.89
perm tm p A 0.98
perm tm t A 4.68
Pore Core H2O Molecules
Simulation Time (ns)

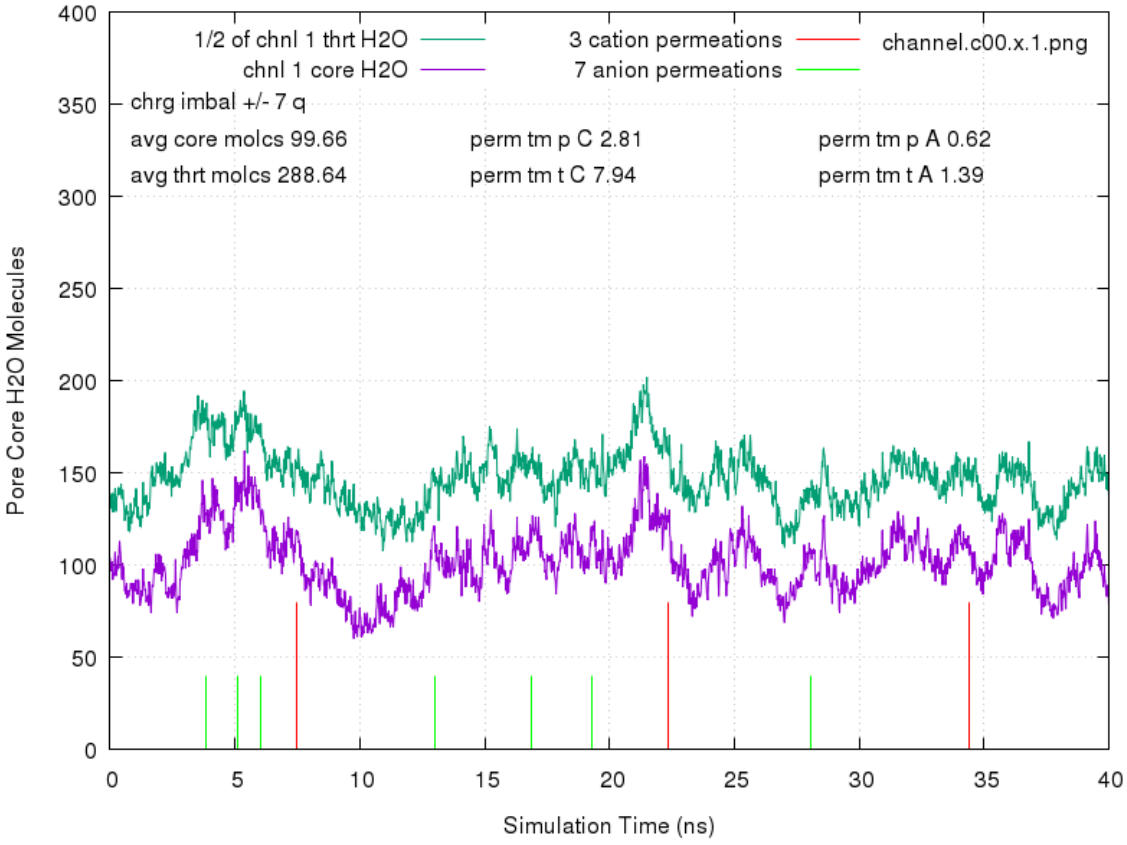

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
4 anion permeations
channel.c00.v.0.png
chrg imbal +/- 7 q
avg core molcs 102.34
avg thrt molcs 293.48
perm tm p C 1.57
perm tm t C 3.31
perm tm p A 0.77
perm tm t A 2.58
Pore Core H2O Molecules
Simulation Time (ns)

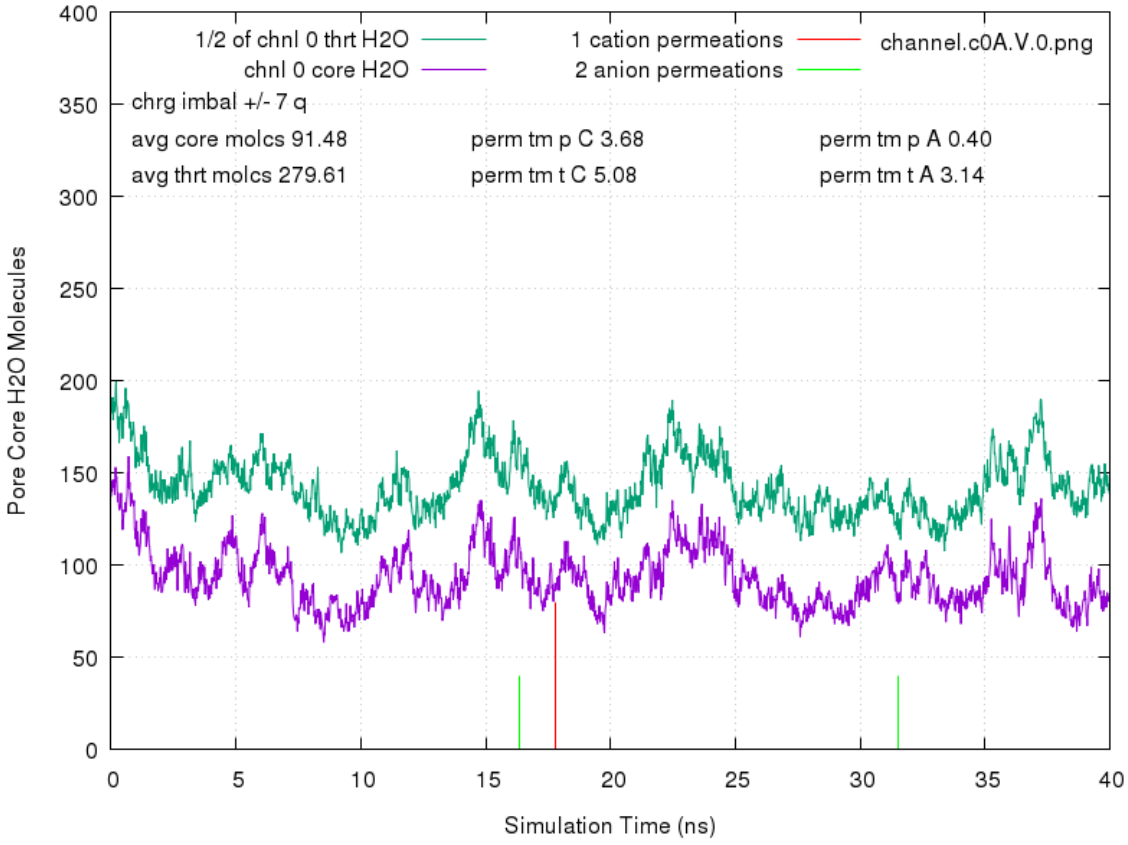

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
5 cation permeations
9 anion permeations
channel.c00.v.1.png
chrg imbal +/- 7 q
avg core molcs 111.65
avg thrt molcs 312.12
perm tm p C 1.45
perm tm t C 2.94
perm tm p A 0.46
perm tm t A 1.40
Pore Core H2O Molecules
Simulation Time (ns)

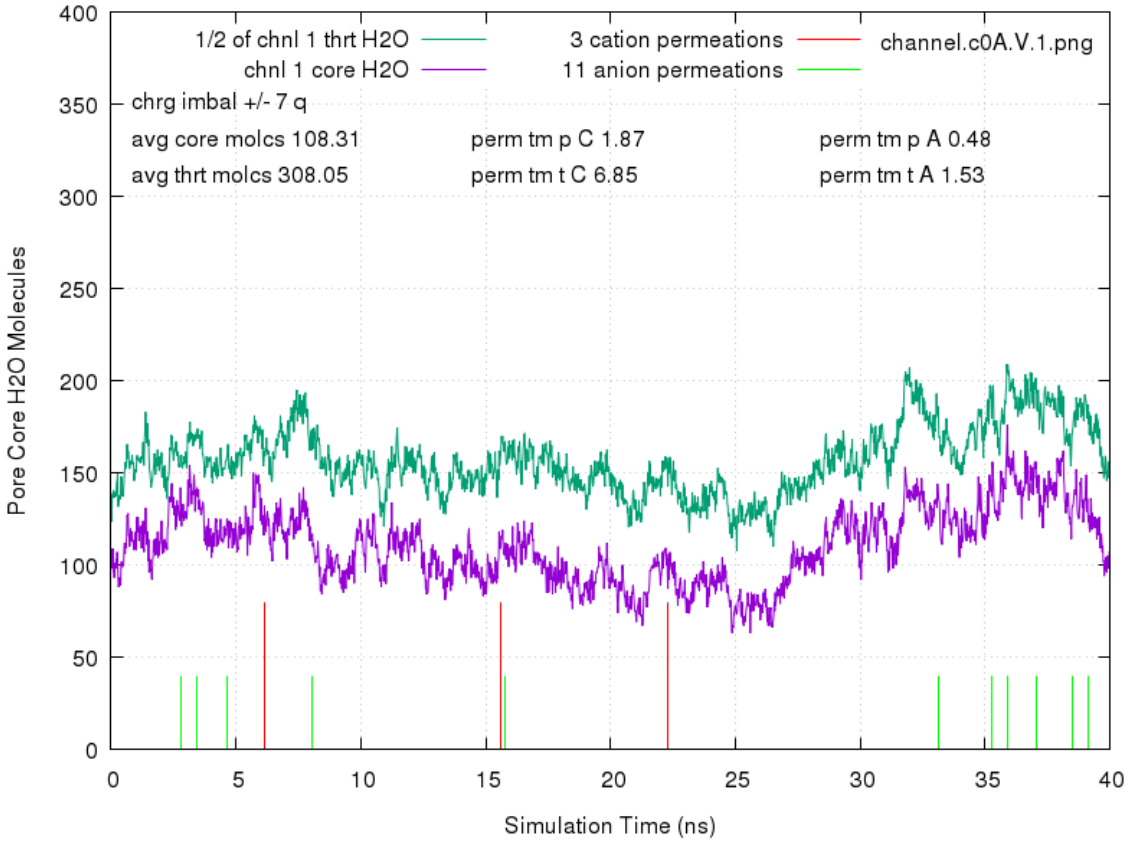

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
3 anion permeations
channel.c00.w.0.png
chrg imbal +/- 7 q
avg core molcs 82.33
avg thrt molcs 263.16
perm tm p C 2.26
perm tm t C 9.56
perm tm p A 1.03
perm tm t A 2.58
Pore Core H2O Molecules
Simulation Time (ns)

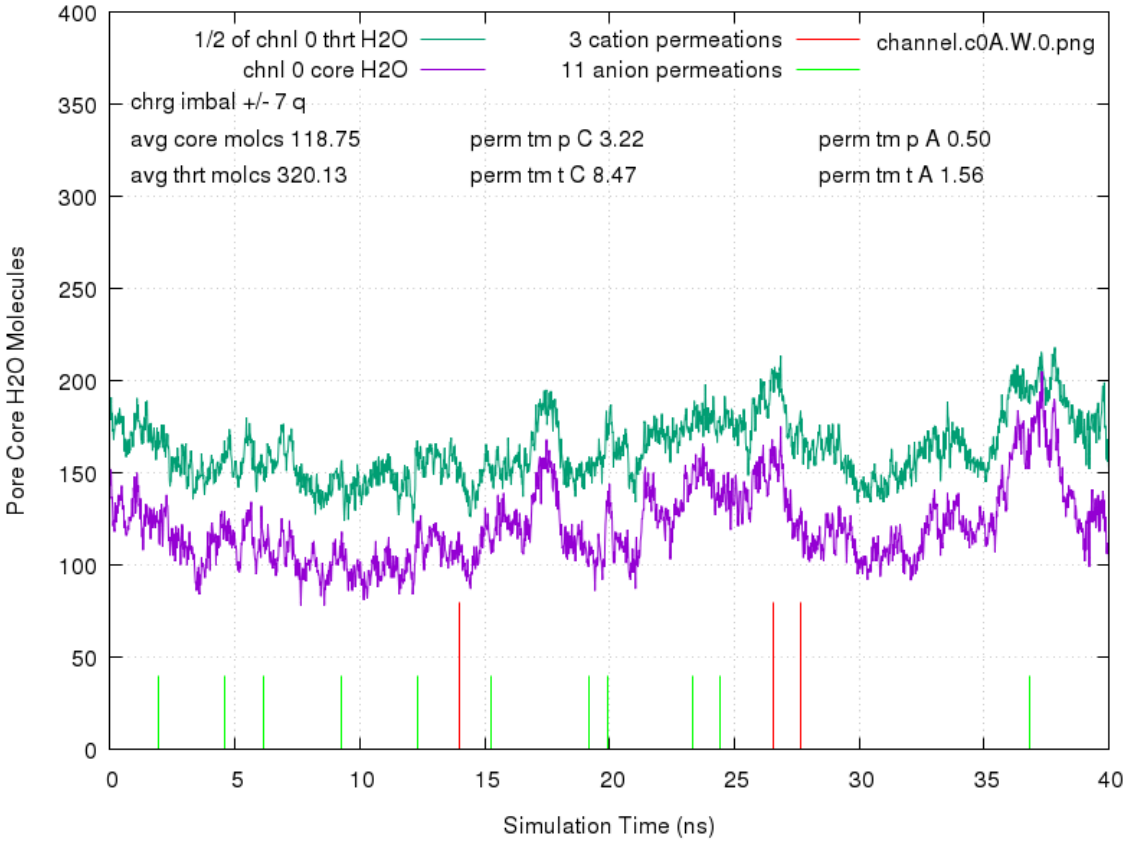

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
12 cation permeations
18 anion permeations
channel.c00.w.1.png
chrg imbal +/- 7 q
avg core molcs 136.55
avg thrt molcs 356.55
perm tm p C 1.15
perm tm t C 2.90
perm tm p A 0.52
perm tm t A 1.54
Pore Core H2O Molecules
Simulation Time (ns)

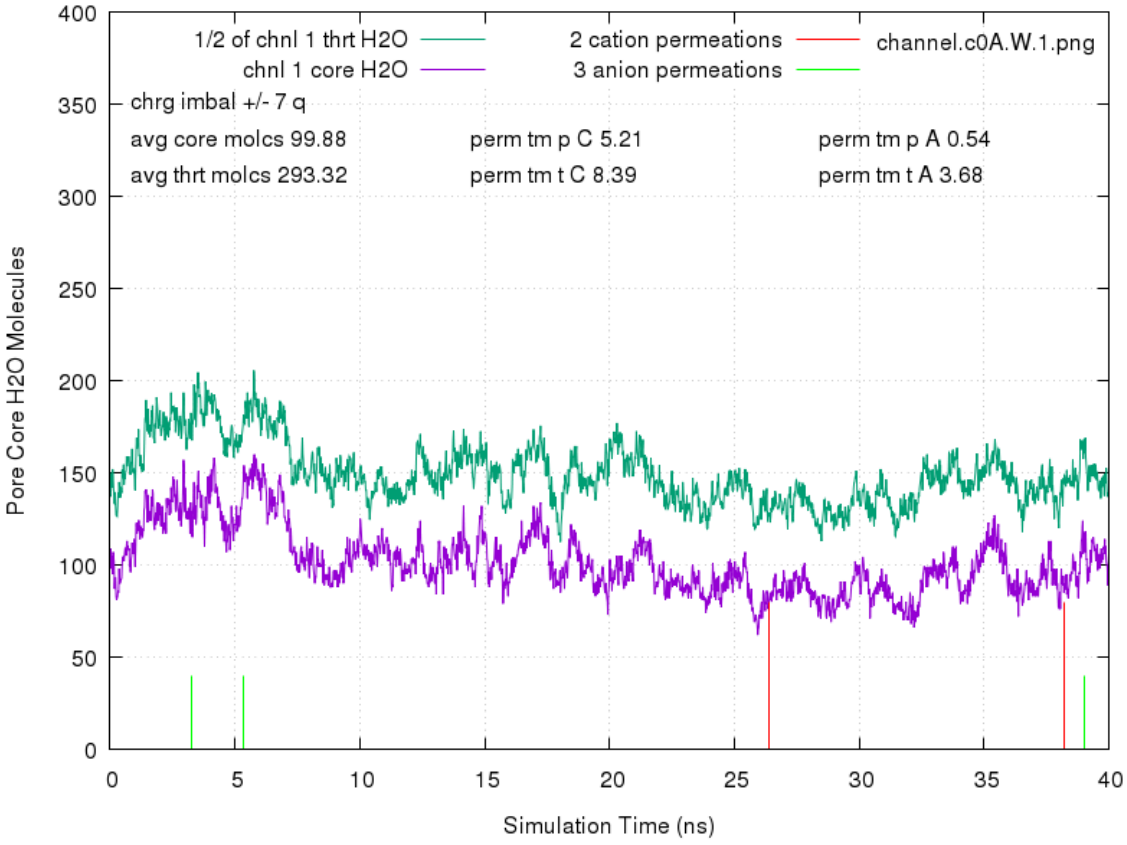

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
7 cation permeations
12 anion permeations
channel.c00.x.0.png
chrg imbal +/- 7 q
avg core molcs 125.97
avg thrt molcs 336.51
perm tm p C 1.19
perm tm t C 2.40
perm tm p A 0.46
perm tm t A 1.27
Pore Core H2O Molecules
Simulation Time (ns)

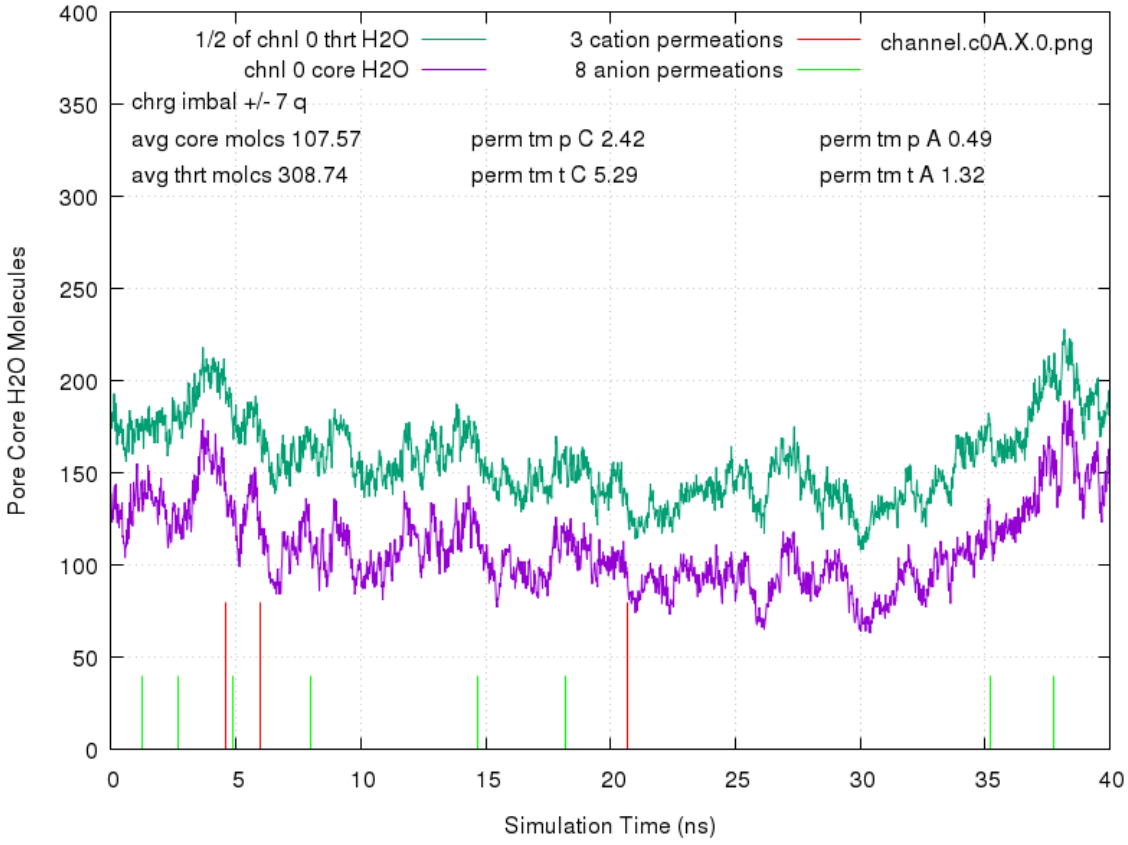

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
7 anion permeations
channel.c00.x.1.png
chrg imbal +/- 7 q
avg core molcs 99.66
avg thrt molcs 288.64
perm tm p C 2.81
perm tm t C 7.94
perm tm p A 0.62
perm tm t A 1.39
Pore Core H2O Molecules
Simulation Time (ns)

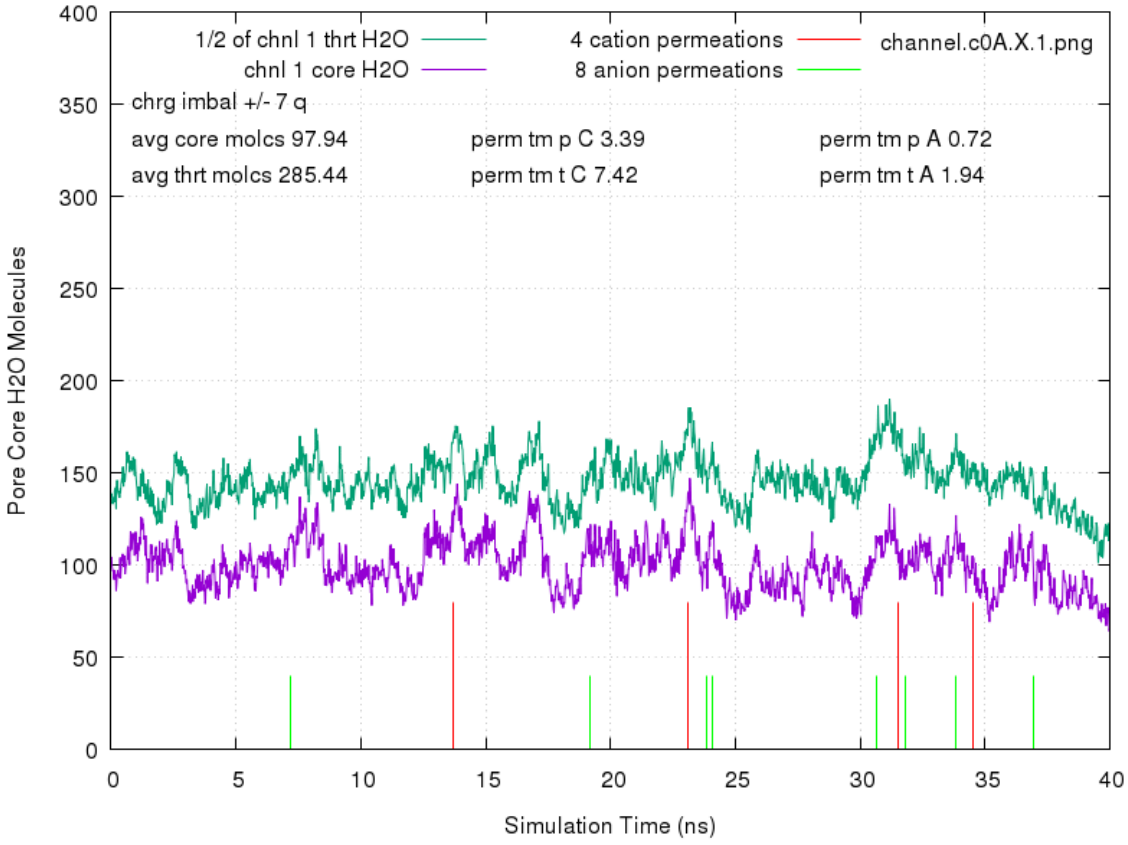

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
1 cation permeations
2 anion permeations
channel.c0A.V.0.png
chrg imbal +/- 7 q
avg core molcs 91.48
avg thrt molcs 279.61
perm tm p C 3.68
perm tm t C 5.08
perm tm p A 0.40
perm tm t A 3.14
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

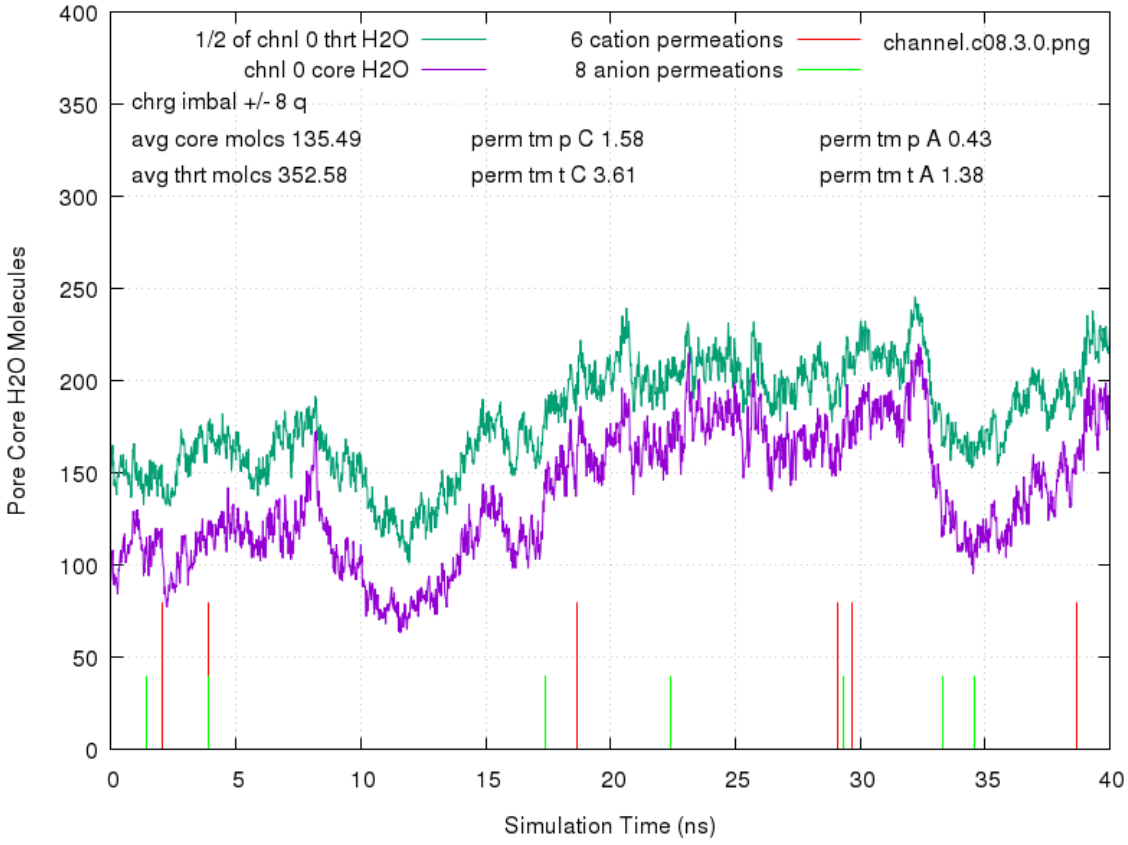

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
11 anion permeations
channel.c0A.V.1.png
chrg imbal +/- 7 q
avg core molcs 108.31
avg thrt molcs 308.05
perm tm p C 1.87
perm tm t C 6.85
perm tm p A 0.48
perm tm t A 1.53
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

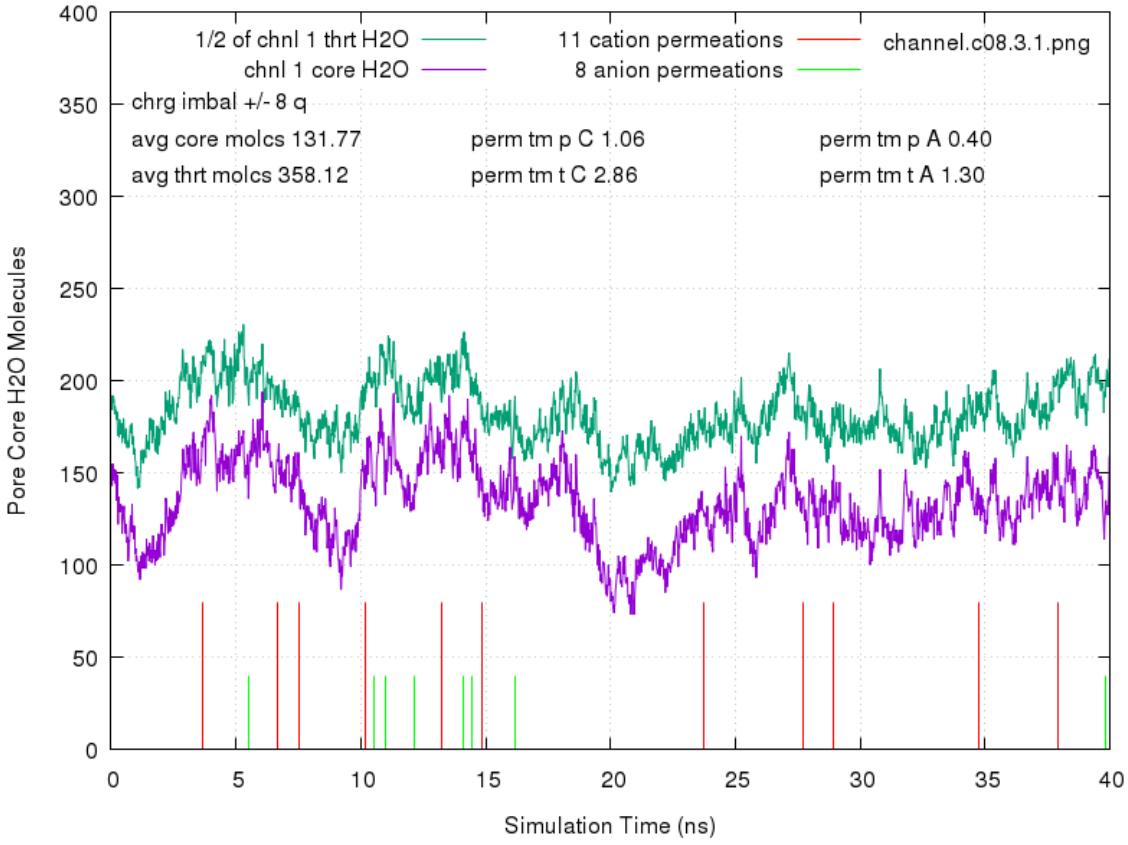

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
11 anion permeations
channel.c0A.W.0.png
chrg imbal +/- 7 q
avg core molcs 118.75
avg thrt molcs 320.13
perm tm p C 3.22
perm tm t C 8.47
perm tm p A 0.50
perm tm t A 1.56
Pore Core H2O Molecules
Simulation Time (ns)

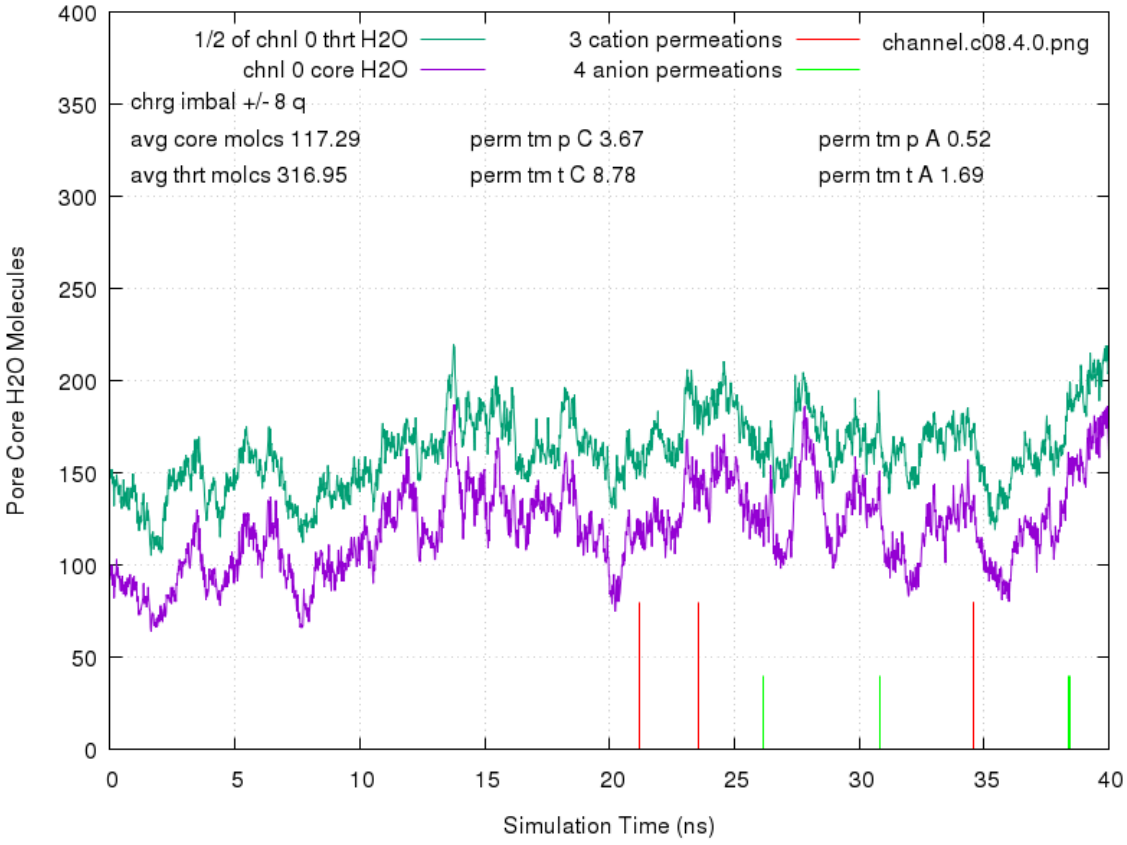

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
3 anion permeations
channel.c0A.W.1.png
chrg imbal +/- 7 q
avg core molcs 99.88
avg thrt molcs 293.32
perm tm p C 5.21
perm tm t C 8.39
perm tm p A 0.54
perm tm t A 3.68
Pore Core H2O Molecules
Simulation Time (ns)

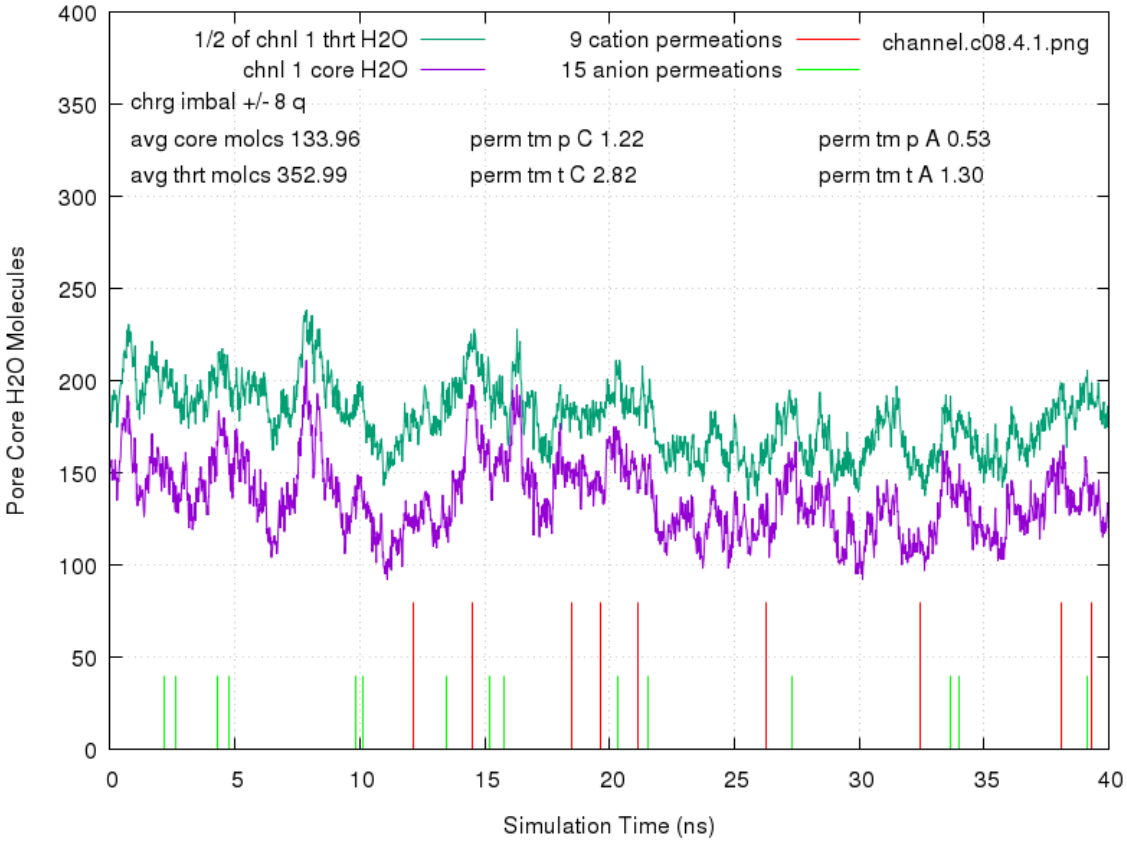

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
8 anion permeations
channel.c0A.X.0.png
chrg imbal +/- 7 q
avg core molcs 107.57
avg thrt molcs 308.74
perm tm p C 2.42
perm tm t C 5.29
perm tm p A 0.49
perm tm t A 1.32
Pore Core H2O Molecules
Simulation Time (ns)

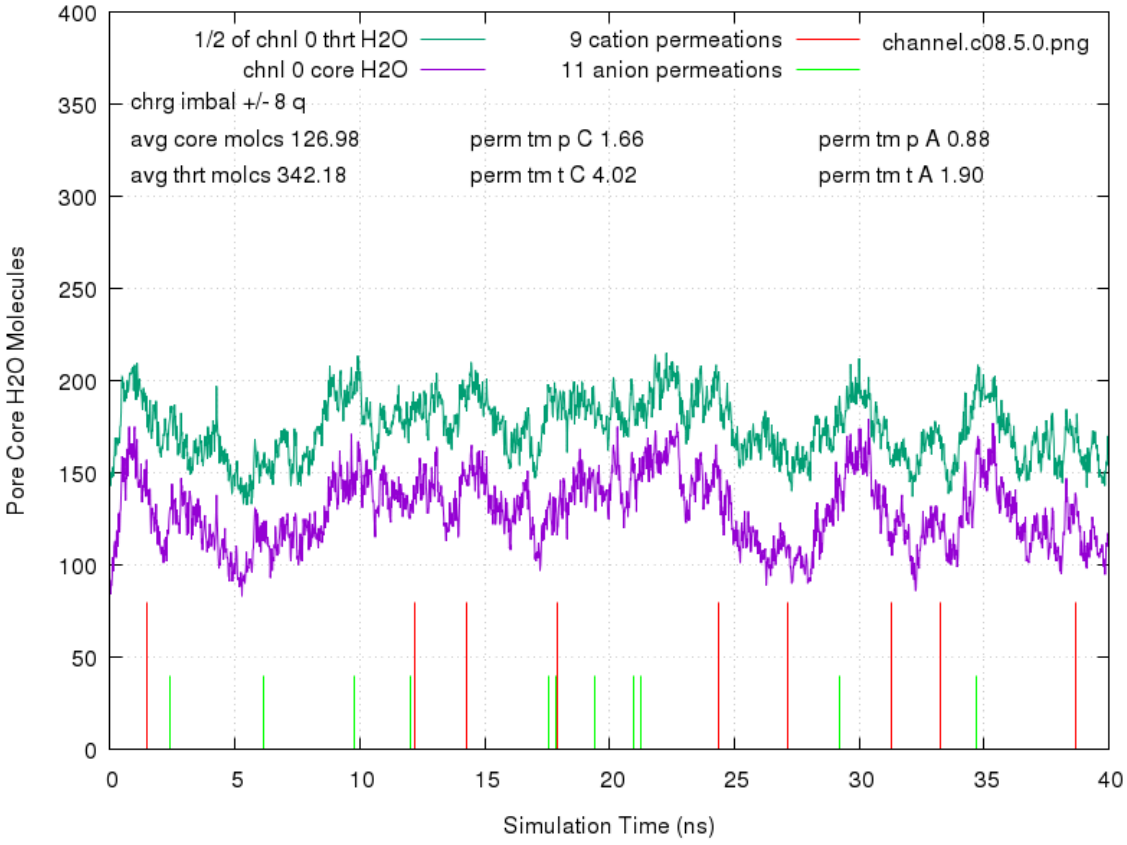

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
8 anion permeations
channel.c0A.X.1.png
chrg imbal +/- 7 q
avg core molcs 97.94
avg thrt molcs 285.44
perm tm p C 3.39
perm tm t C 7.42
perm tm p A 0.72
perm tm t A 1.94
Pore Core H2O Molecules
Simulation Time (ns)

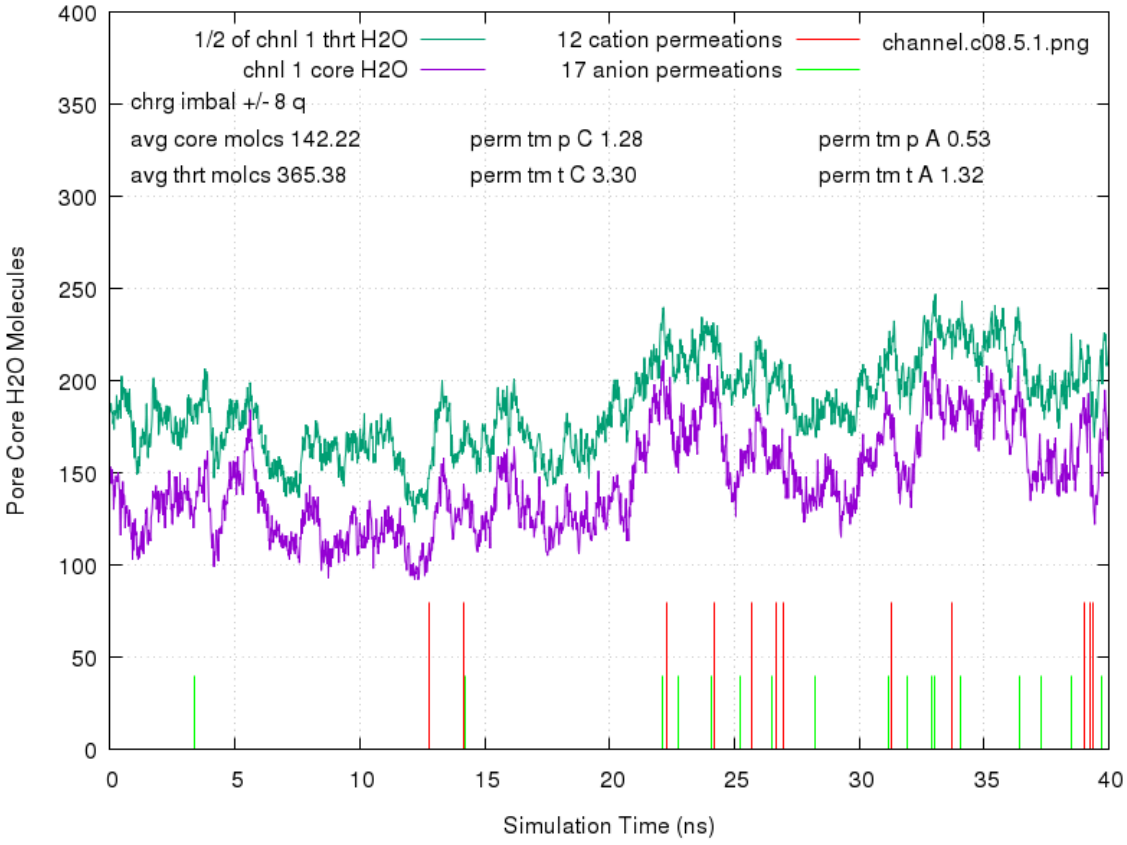

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
6 cation permeations
8 anion permeations
channel.c08.3.0.png
chrg imbal +/- 8 q
avg core molcs 135.49
avg thrt molcs 352.58
perm tm p C 1.58
perm tm t C 3.61
perm tm p A 0.43
perm tm t A 1.38
Pore Core H2O Molecules
Simulation Time (ns)

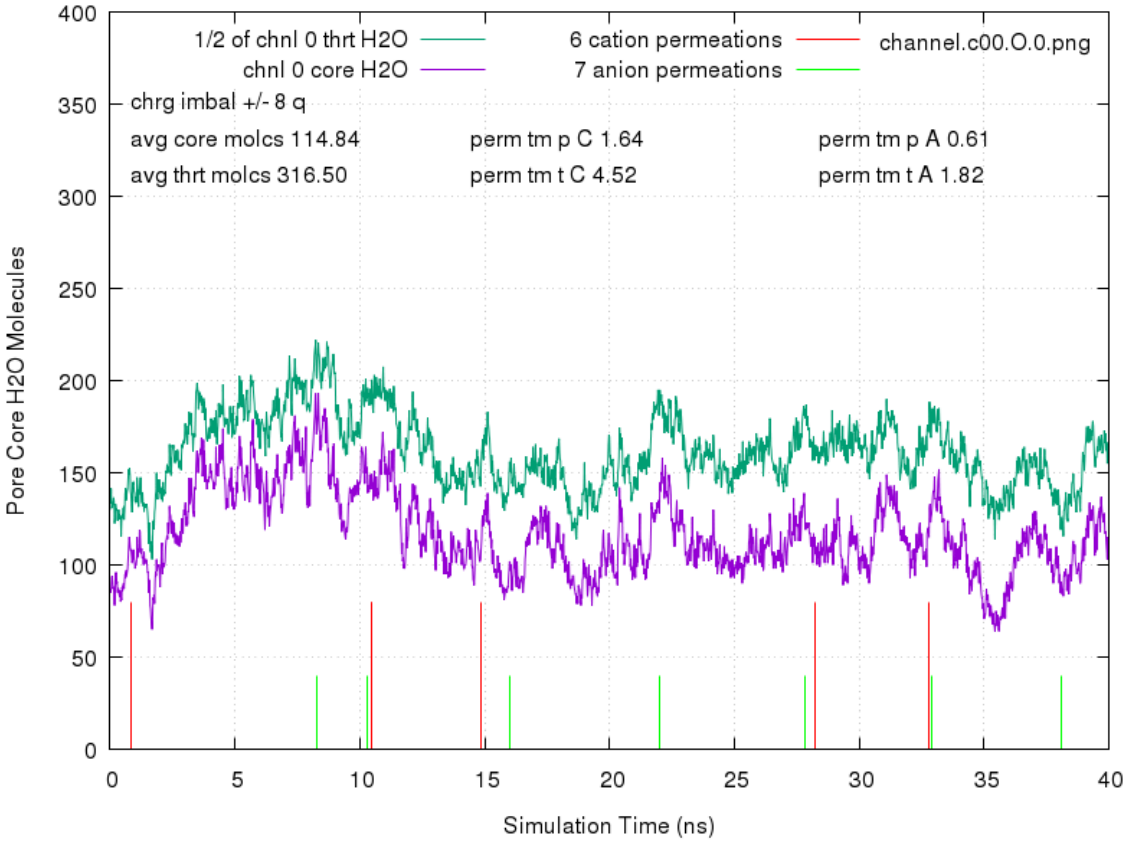

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
11 cation permeations
8 anion permeations
channel.c08.3.1.png
chrg imbal +/- 8 q
avg core molcs 131.77
avg thrt molcs 358.12
perm tm p C 1.06
perm tm t C 2.86
perm tm p A 0.40
perm tm t A 1.30
Pore Core H2O Molecules
Simulation Time (ns)

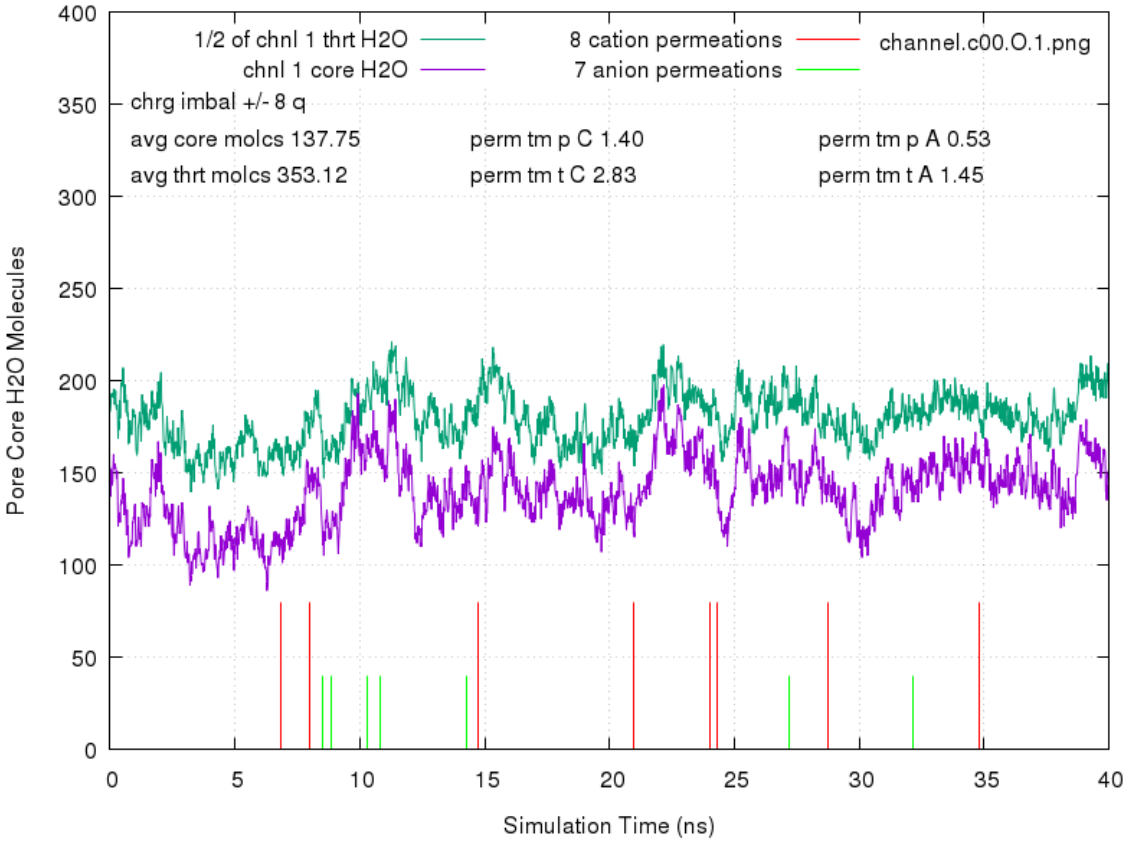

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
3 cation permeations
4 anion permeations
channel.c08.4.0.png
chrg imbal +/- 8 q
avg core molcs 117.29
avg thrt molcs 316.95
perm tm p C 3.67
perm tm t C 8.78
perm tm p A 0.52
perm tm t A 1.69
Pore Core H2O Molecules
Simulation Time (ns)

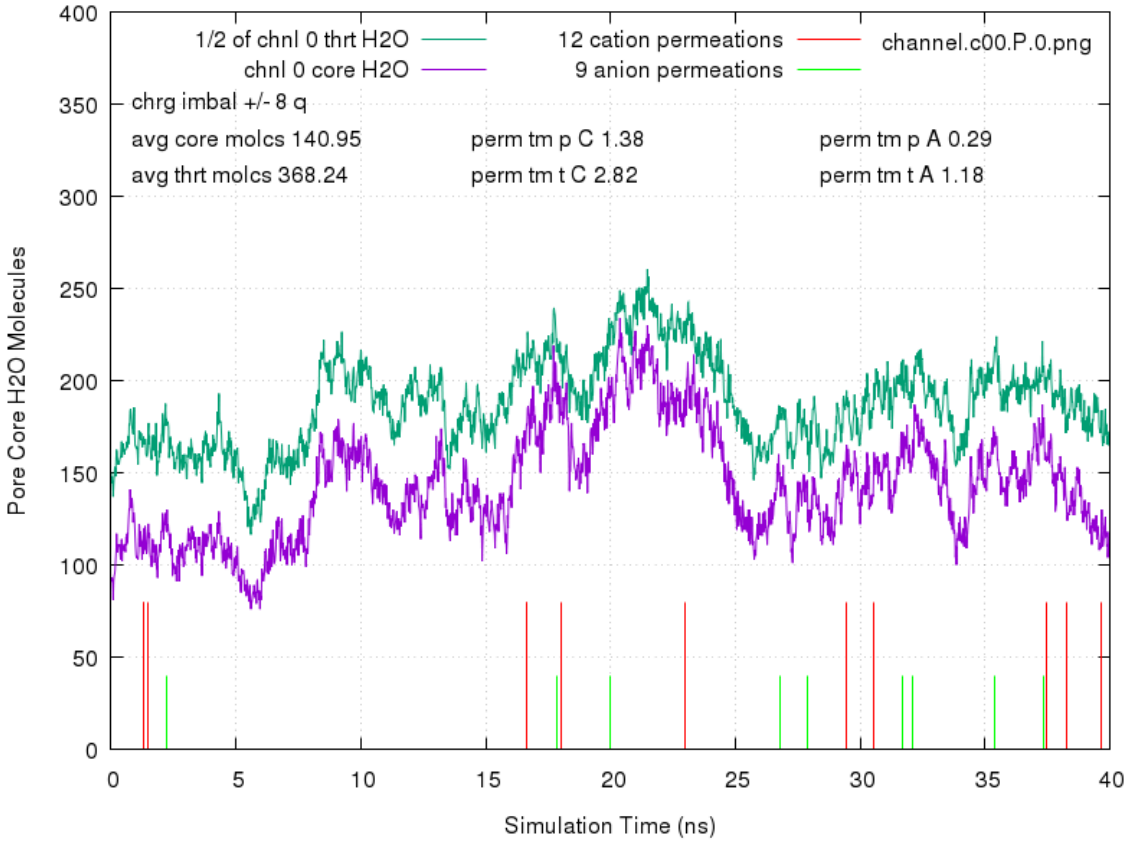

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
9 cation permeations
15 anion permeations
channel.c08.4.1.png
chrg imbal +/- 8 q
avg core molcs 133.96
avg thrt molcs 352.99
perm tm p C 1.22
perm tm t C 2.82
perm tm p A 0.53
perm tm t A 1.30
Pore Core H2O Molecules
Simulation Time (ns)

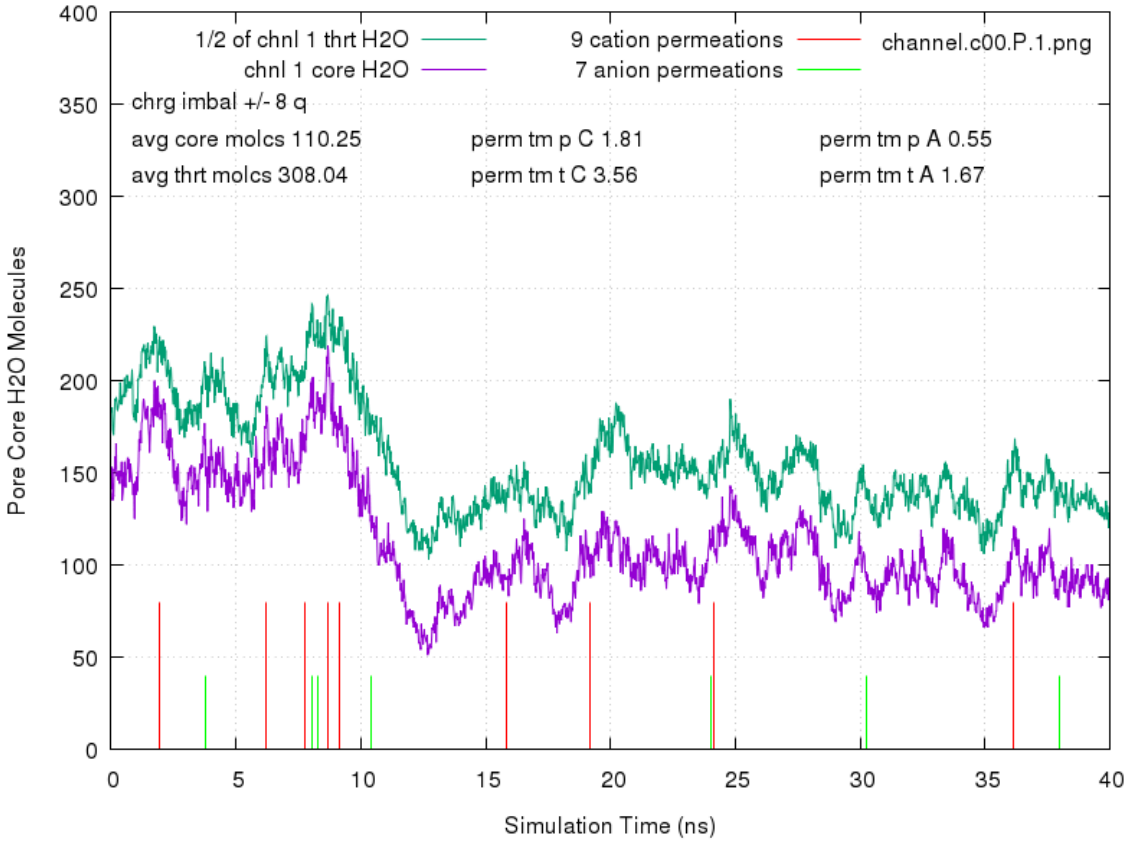

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
9 cation permeations
11 anion permeations
channel.c08.5.0.png
chrg imbal +/- 8 q
avg core molcs 126.98
avg thrt molcs 342.18
perm tm p C 1.66
perm tm t C 4.02
perm tm p A 0.88
perm tm t A 1.90
Pore Core H2O Molecules
Simulation Time (ns)

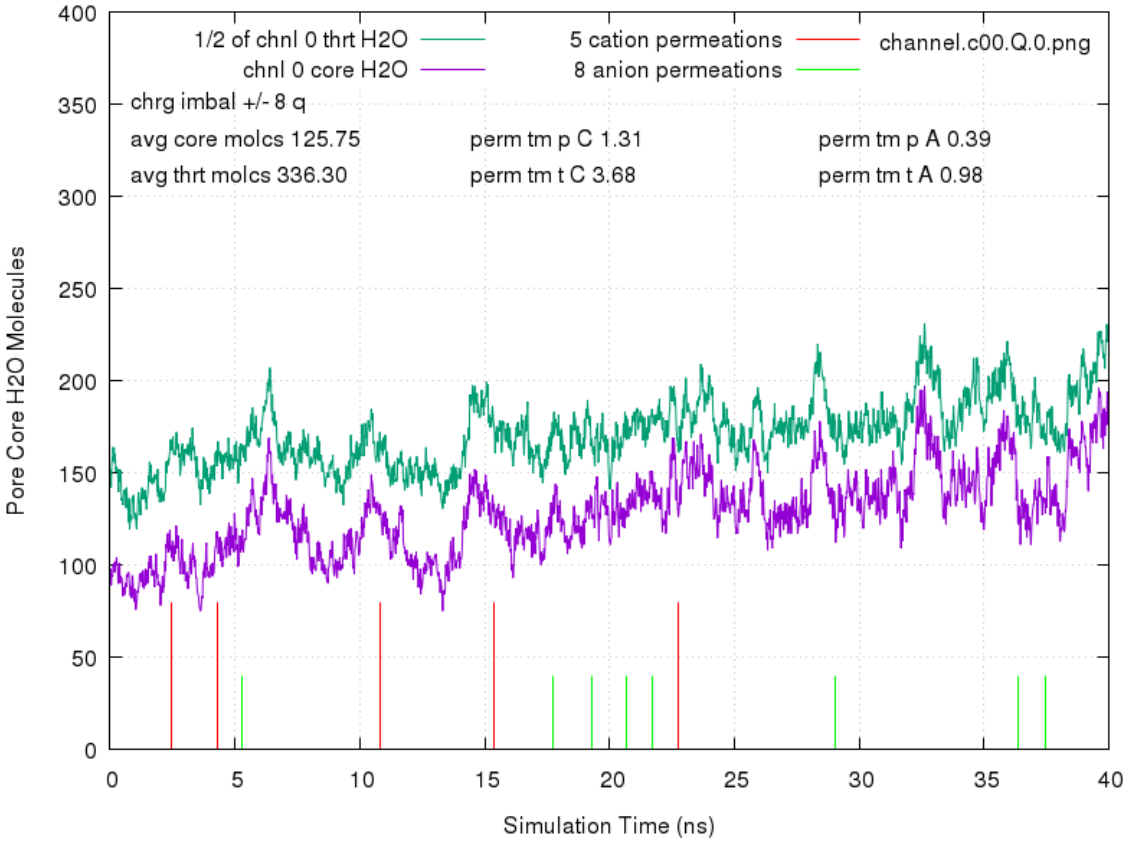

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
12 cation permeations
17 anion permeations
channel.c08.5.1.png
chrg imbal +/- 8 q
avg core molcs 142.22
avg thrt molcs 365.38
perm tm p C 1.28
perm tm t C 3.30
perm tm p A 0.53
perm tm t A 1.32
Pore Core H2O Molecules
Simulation Time (ns)

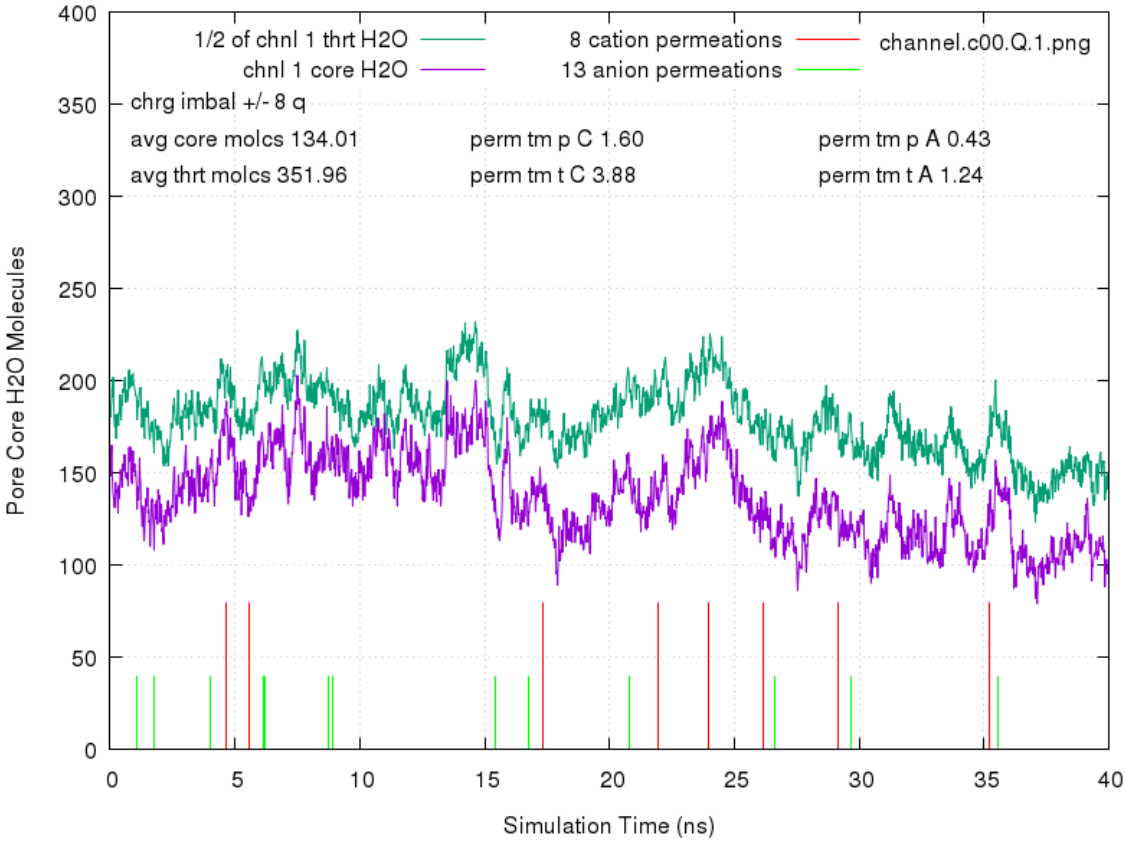

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
6 cation permeations
7 anion permeations
channel.c00.O.0.png
chrg imbal +/- 8 q
avg core molcs 114.84
avg thrt molcs 316.50
perm tm p C 1.64
perm tm t C 4.52
perm tm p A 0.61
perm tm t A 1.82
Pore Core H2O Molecules
Simulation Time (ns)

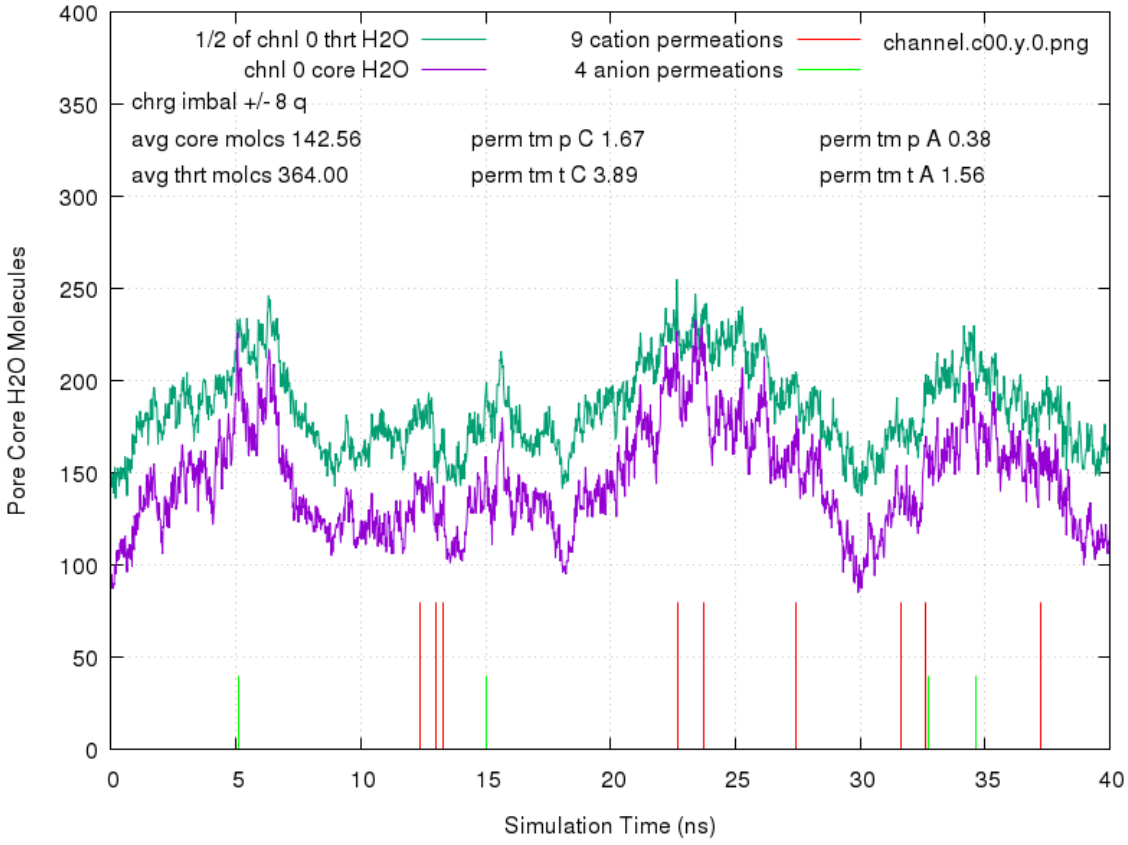

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
8 cation permeations
7 anion permeations
channel.c00.O.1.png
chrg imbal +/- 8 q
avg core molcs 137.75
avg thrt molcs 353.12
perm tm p C 1.40
perm tm t C 2.83
perm tm p A 0.53
perm tm t A 1.45
Pore Core H2O Molecules
Simulation Time (ns)

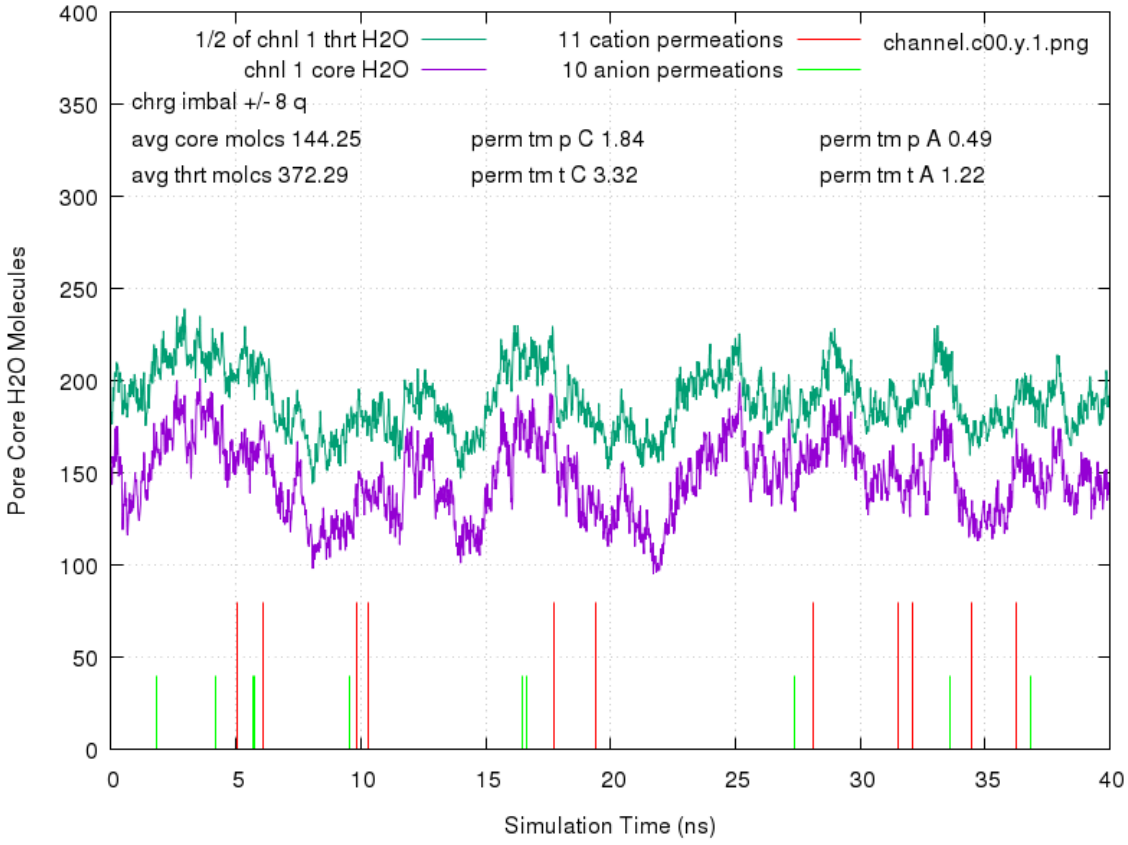

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
12 cation permeations
9 anion permeations
channel.c00.P.0.png
chrg imbal +/- 8 q
avg core molcs 140.95
avg thrt molcs 368.24
perm tm p C 1.38
perm tm t C 2.82
perm tm p A 0.29
perm tm t A 1.18
Pore Core H2O Molecules
Simulation Time (ns)

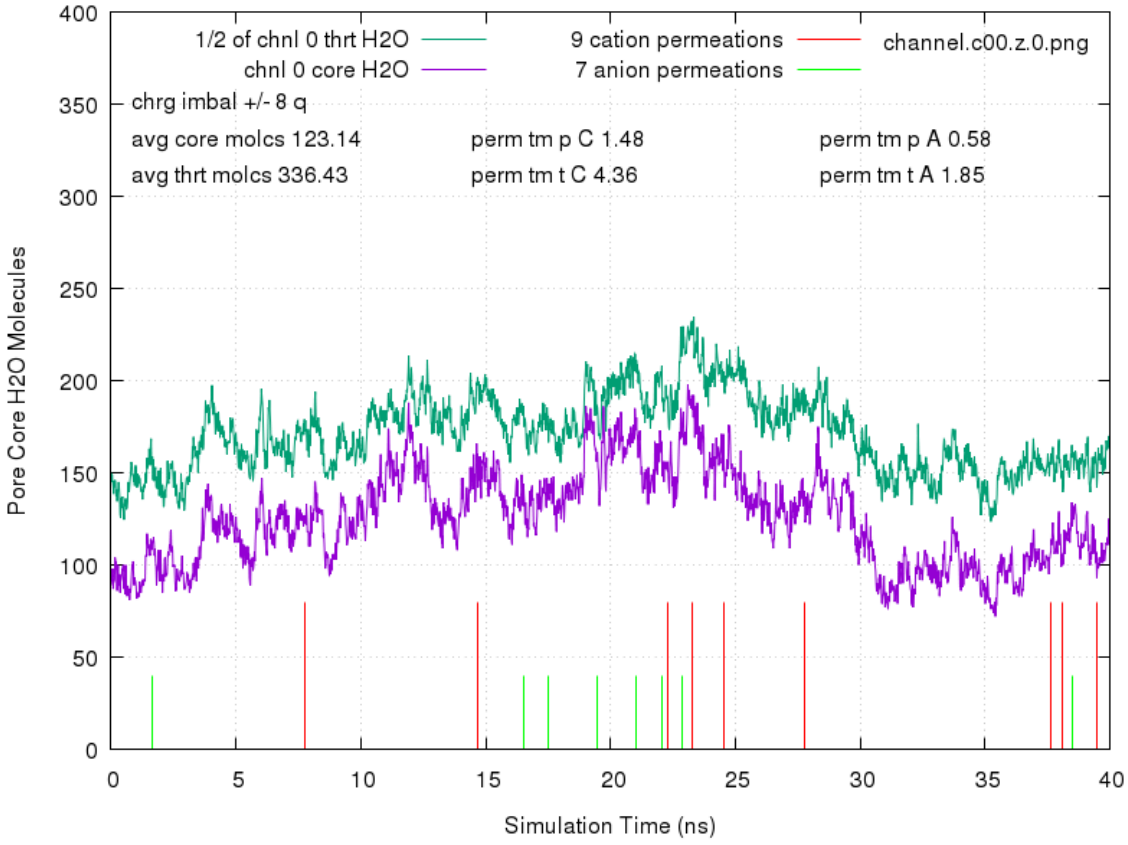

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
9 cation permeations
7 anion permeations
channel.c00.P.1.png
chrg imbal +/- 8 q
avg core molcs 110.25
avg thrt molcs 308.04
perm tm p C 1.81
perm tm t C 3.56
perm tm p A 0.55
perm tm t A 1.67
Pore Core H2O Molecules
Simulation Time (ns)

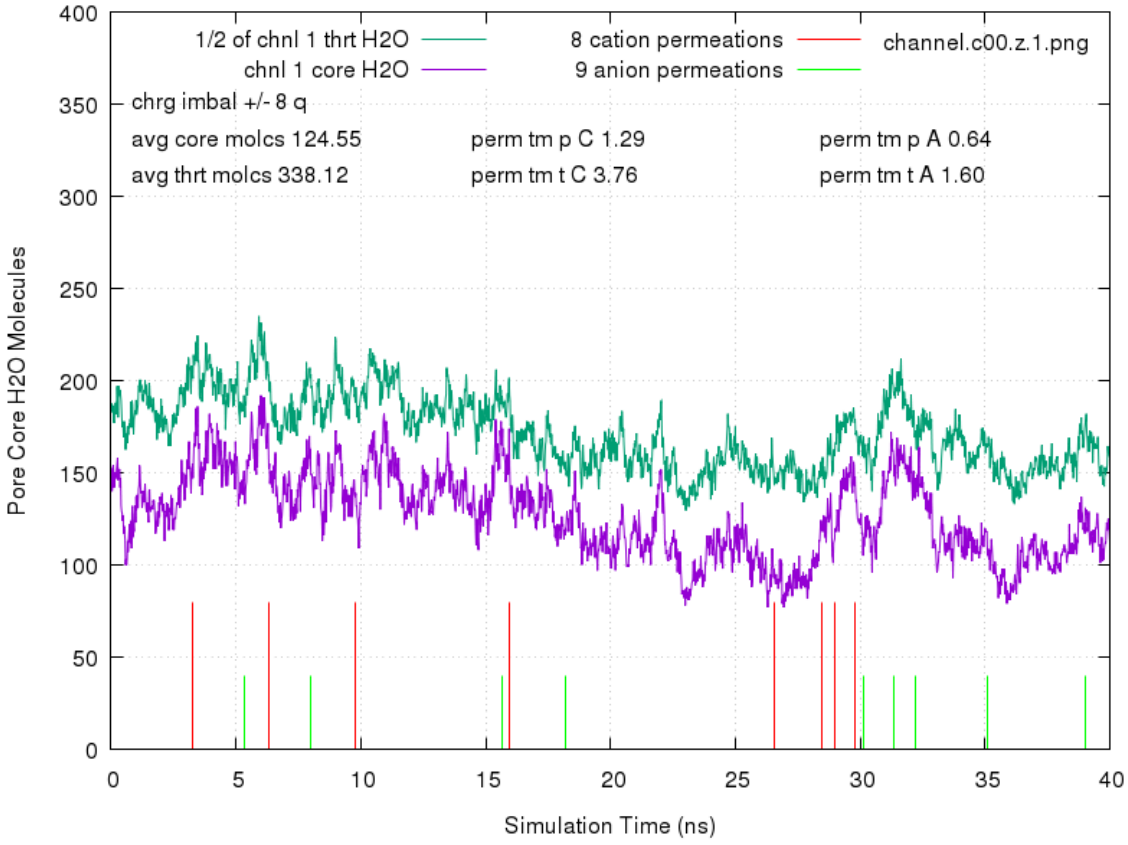

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
8 anion permeations
channel.c00.Q.0.png
chrg imbal +/- 8 q
avg core molcs 125.75
avg thrt molcs 336.30
perm tm p C 1.31
perm tm t C 3.68
perm tm p A 0.39
perm tm t A 0.98
Pore Core H2O Molecules
Simulation Time (ns)

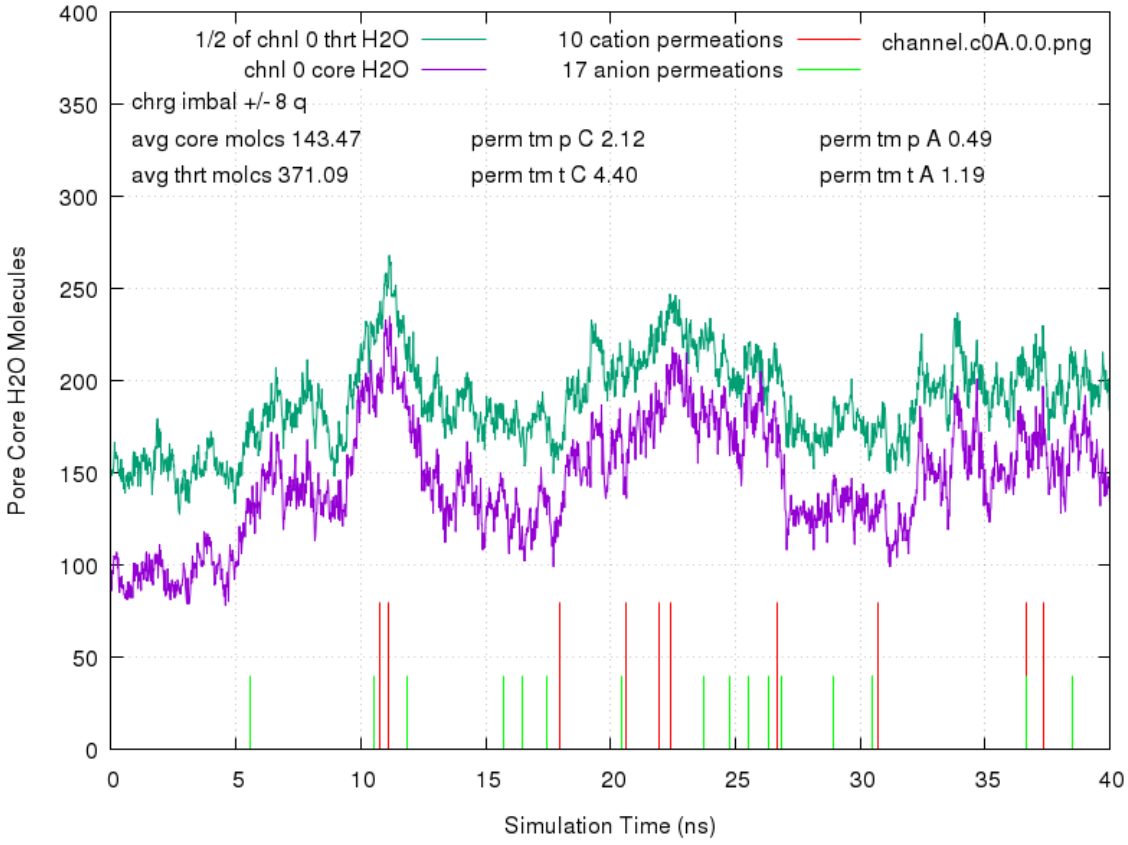

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
8 cation permeations
13 anion permeations
channel.c00.Q.1.png
chrg imbal +/- 8 q
avg core molcs 134.01
avg thrt molcs 351.96
perm tm p C 1.60
perm tm t C 3.88
perm tm p A 0.43
perm tm t A 1.24
Pore Core H2O Molecules
Simulation Time (ns)

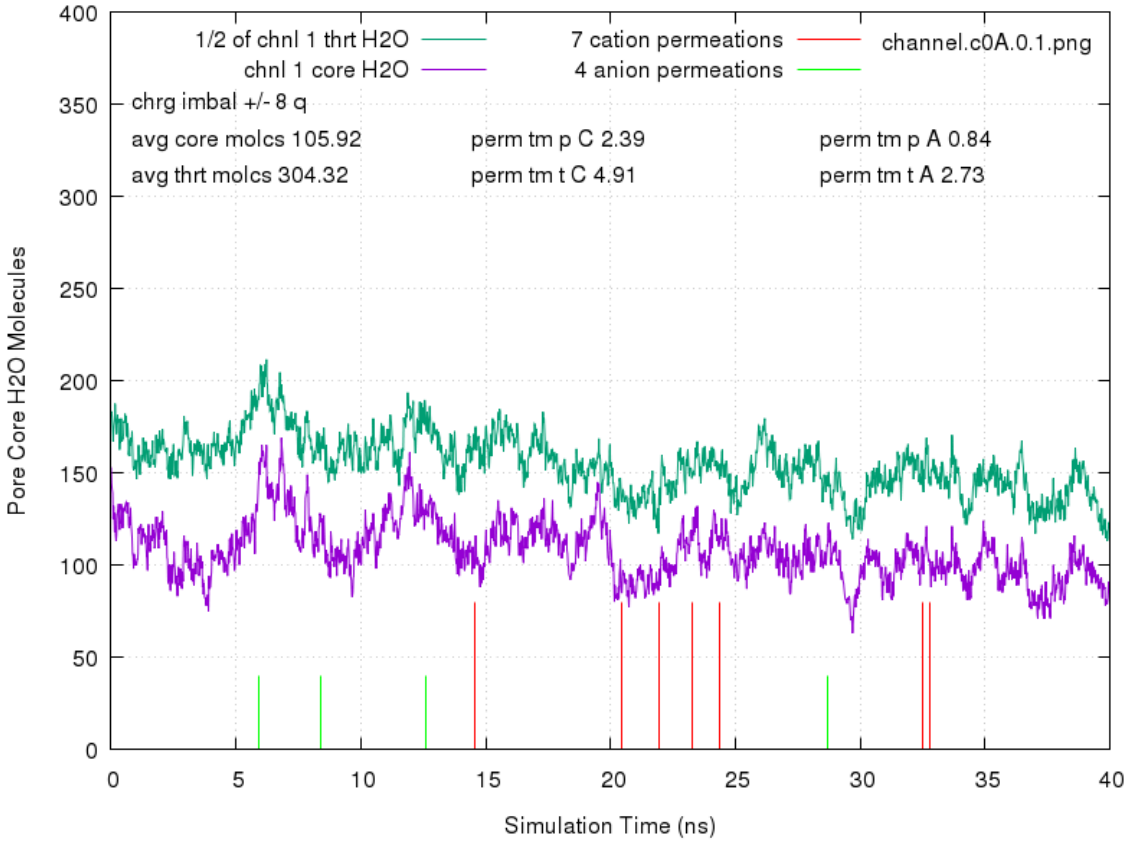

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
9 cation permeations
4 anion permeations
channel.c00.y.0.png
chrg imbal +/- 8 q
avg core molcs 142.56
avg thrt molcs 364.00
perm tm p C 1.67
perm tm t C 3.89
perm tm p A 0.38
perm tm t A 1.56
Pore Core H2O Molecules
Simulation Time (ns)

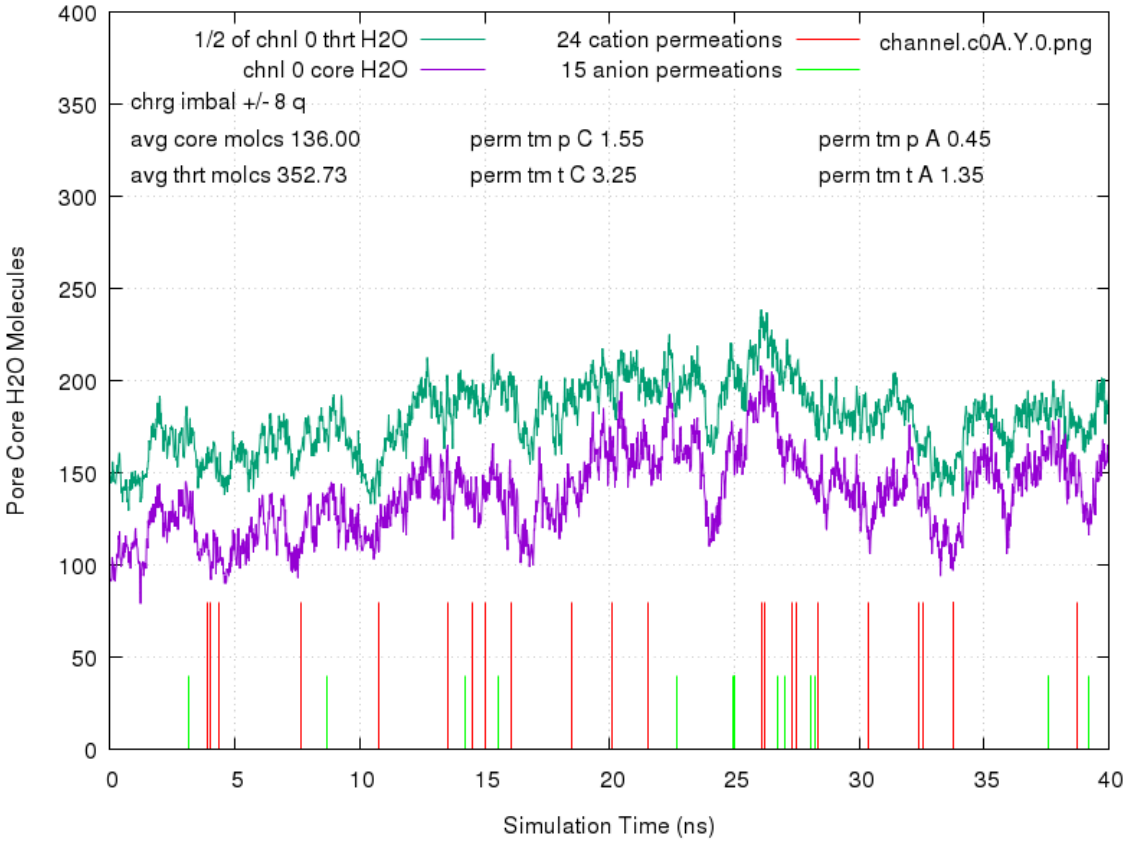

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
11 cation permeations
10 anion permeations
channel.c00.y.1.png
chrg imbal +/- 8 q
avg core molcs 144.25
avg thrt molcs 372.29
perm tm p C 1.84
perm tm t C 3.32
perm tm p A 0.49
perm tm t A 1.22
Pore Core H2O Molecules
Simulation Time (ns)

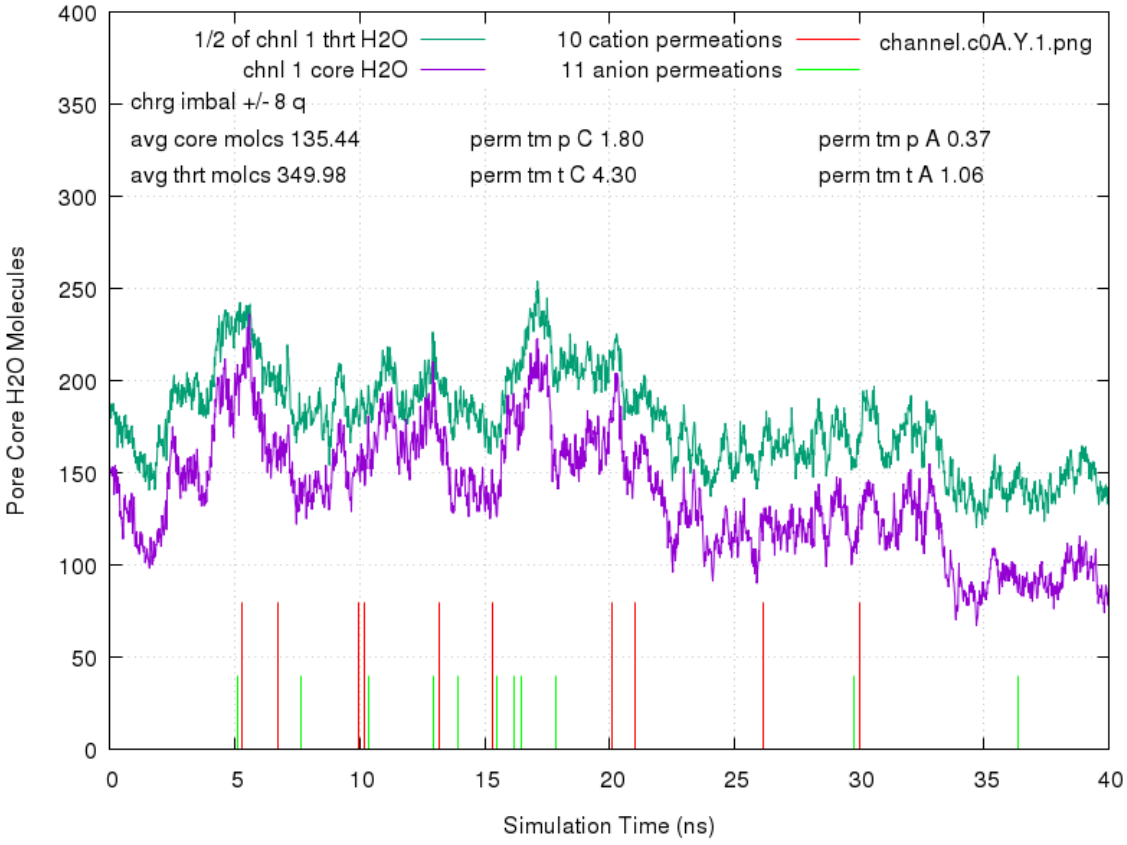

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
9 cation permeations
7 anion permeations
channel.c00.z.0.png
chrg imbal +/- 8 q
avg core molcs 123.14
avg thrt molcs 336.43
perm tm p C 1.48
perm tm t C 4.36
perm tm p A 0.58
perm tm t A 1.85
Pore Core H2O Molecules
Simulation Time (ns)

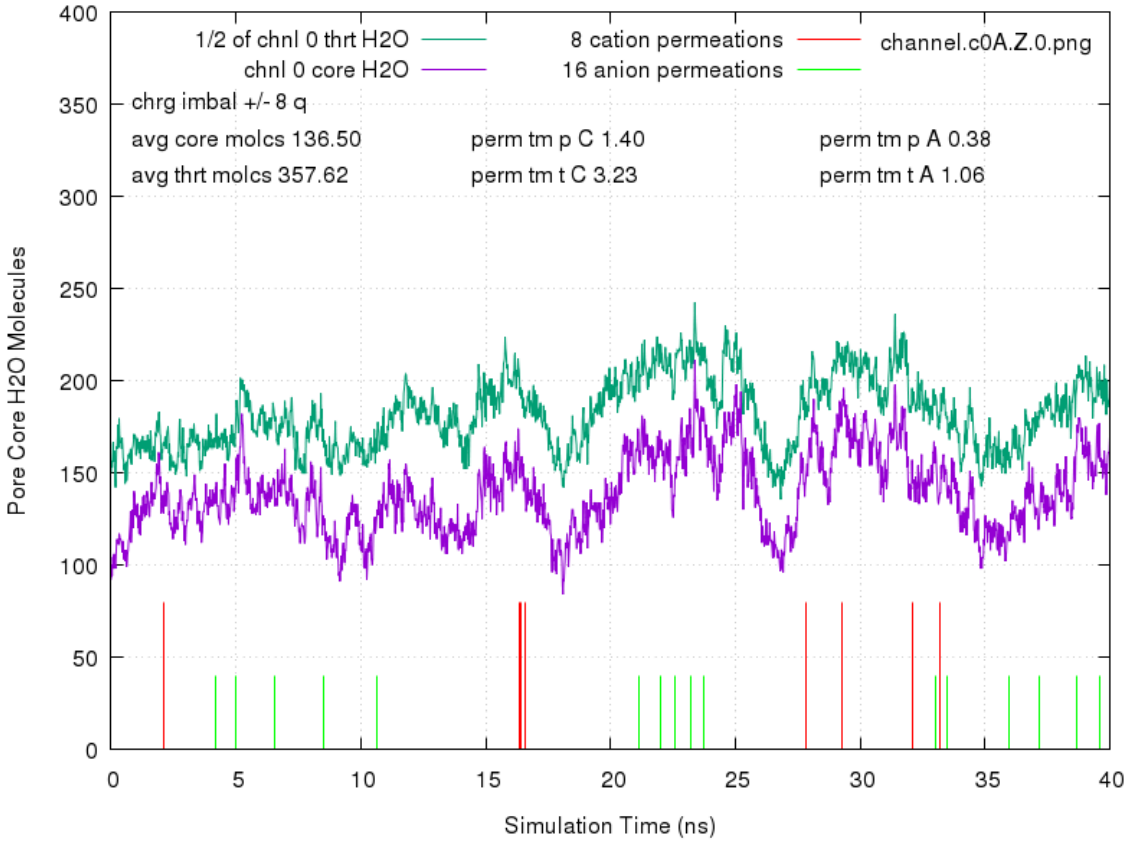

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
8 cation permeations
9 anion permeations
channel.c00.z.1.png
chrg imbal +/- 8 q
avg core molcs 124.55
avg thrt molcs 338.12
perm tm p C 1.29
perm tm t C 3.76
perm tm p A 0.64
perm tm t A 1.60
Pore Core H2O Molecules
Simulation Time (ns)

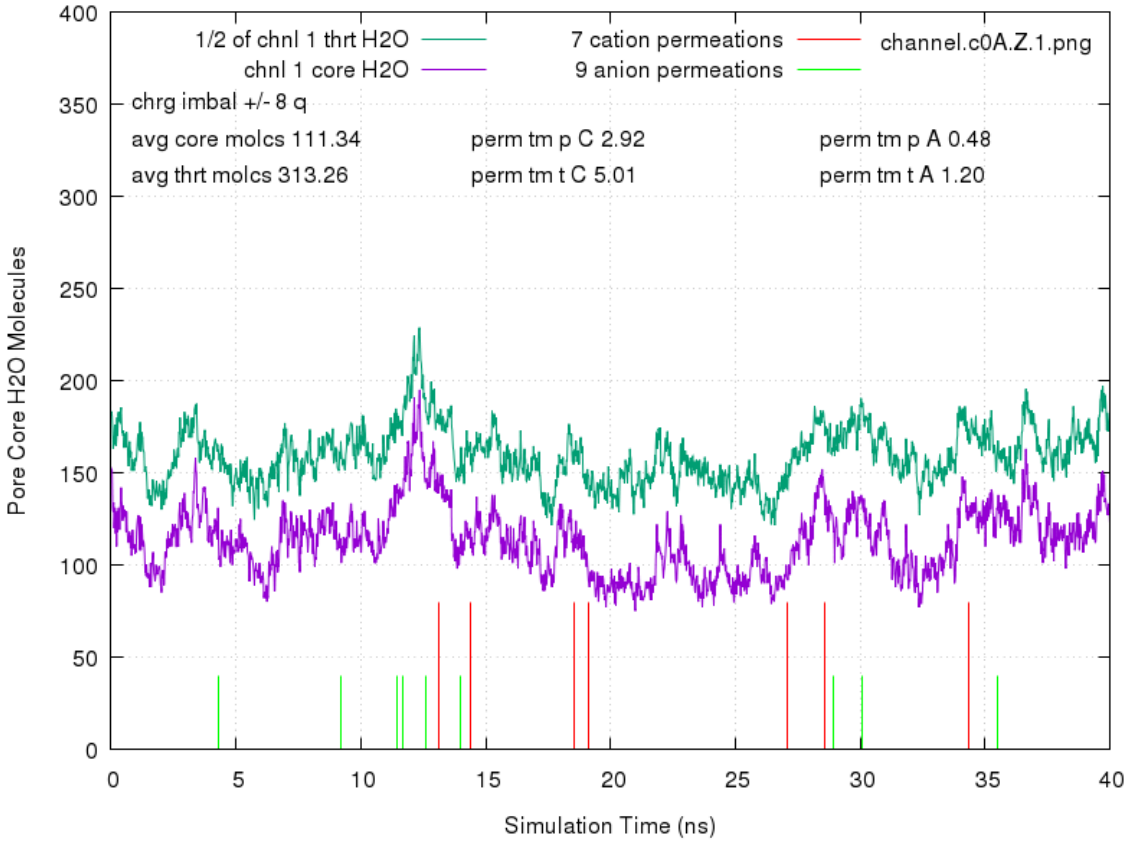

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
10 cation permeations
17 anion permeations
channel.c0A.0.0.png
chrg imbal +/- 8 q
avg core molcs 143.47
avg thrt molcs 371.09
perm tm p C 2.12
perm tm t C 4.40
perm tm p A 0.49
perm tm t A 1.19
Pore Core H2O Molecules
Simulation Time (ns)

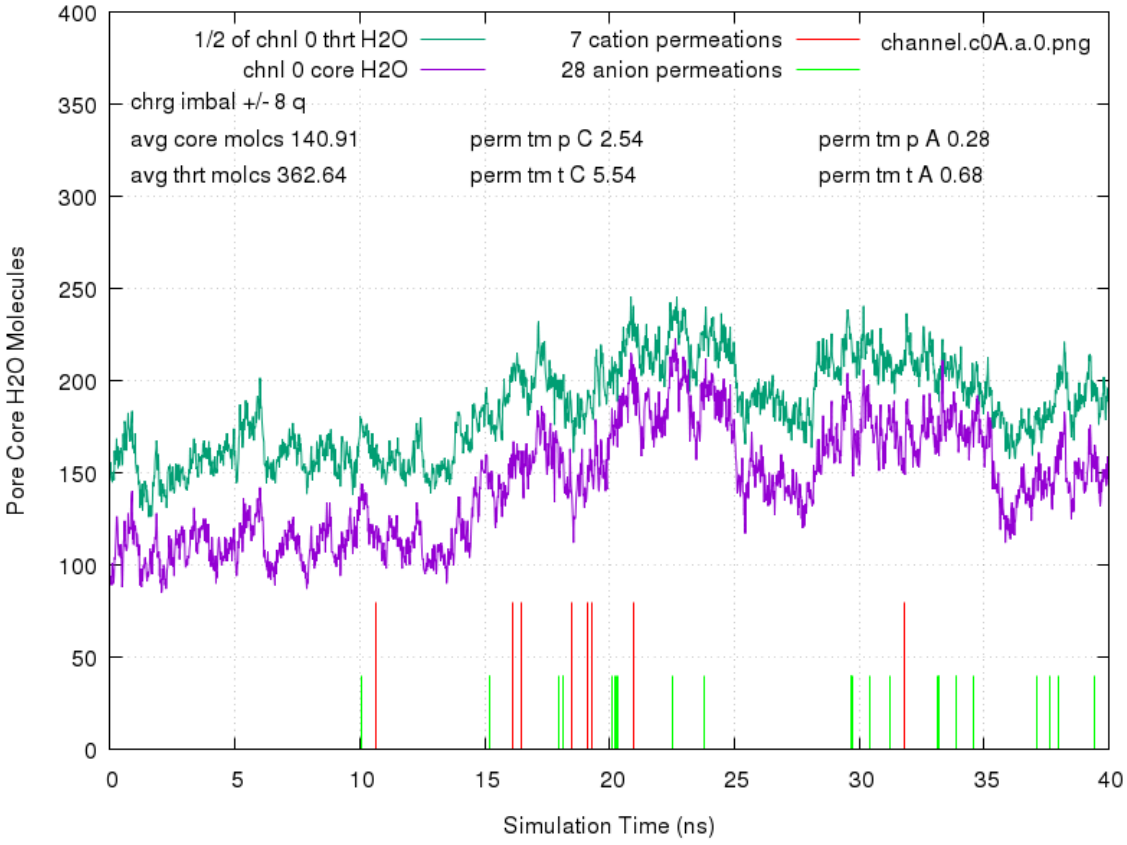

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
7 cation permeations
4 anion permeations
channel.c0A.0.1.png
chrg imbal +/- 8 q
avg core molcs 105.92
avg thrt molcs 304.32
perm tm p C 2.39
perm tm t C 4.91
perm tm p A 0.84
perm tm t A 2.73
Pore Core H2O Molecules
Simulation Time (ns)

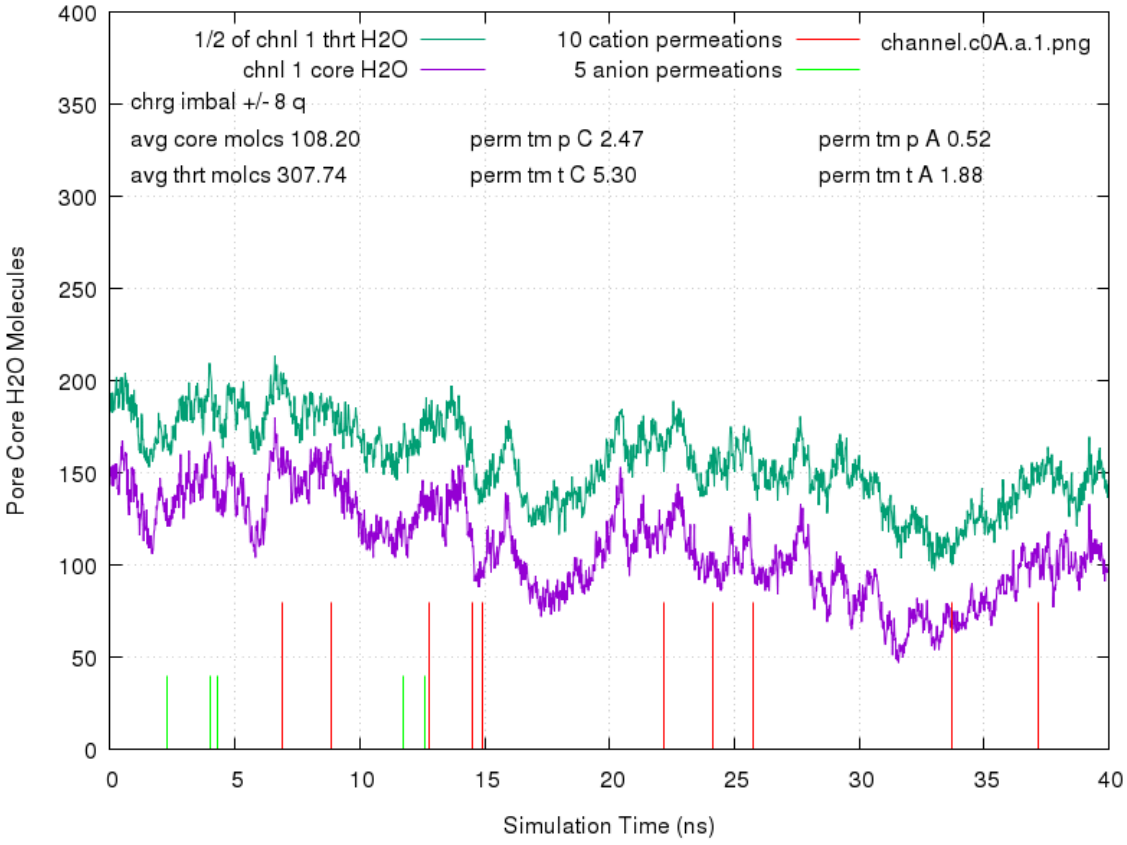

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
24 cation permeations
15 anion permeations
channel.c0A.Y.0.png
chrg imbal +/- 8 q
avg core molcs 136.00
avg thrt molcs 352.73
perm tm p C 1.55
perm tm t C 3.25
perm tm p A 0.45
perm tm t A 1.35
Pore Core H2O Molecules
Simulation Time (ns)

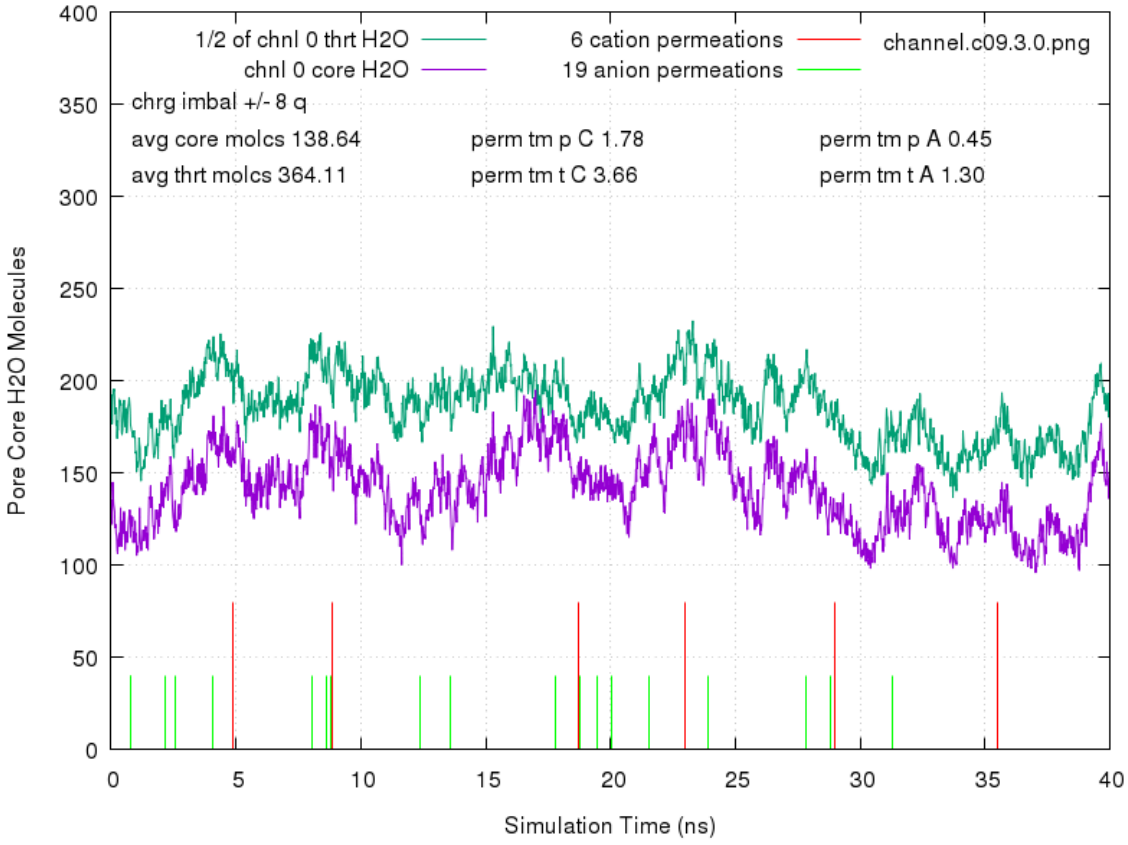

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
10 cation permeations
11 anion permeations
channel.c0A.Y.1.png
chrg imbal +/- 8 q
avg core molcs 135.44
avg thrt molcs 349.98
perm tm p C 1.80
perm tm t C 4.30
perm tm p A 0.37
perm tm t A 1.06
Pore Core H2O Molecules
Simulation Time (ns)

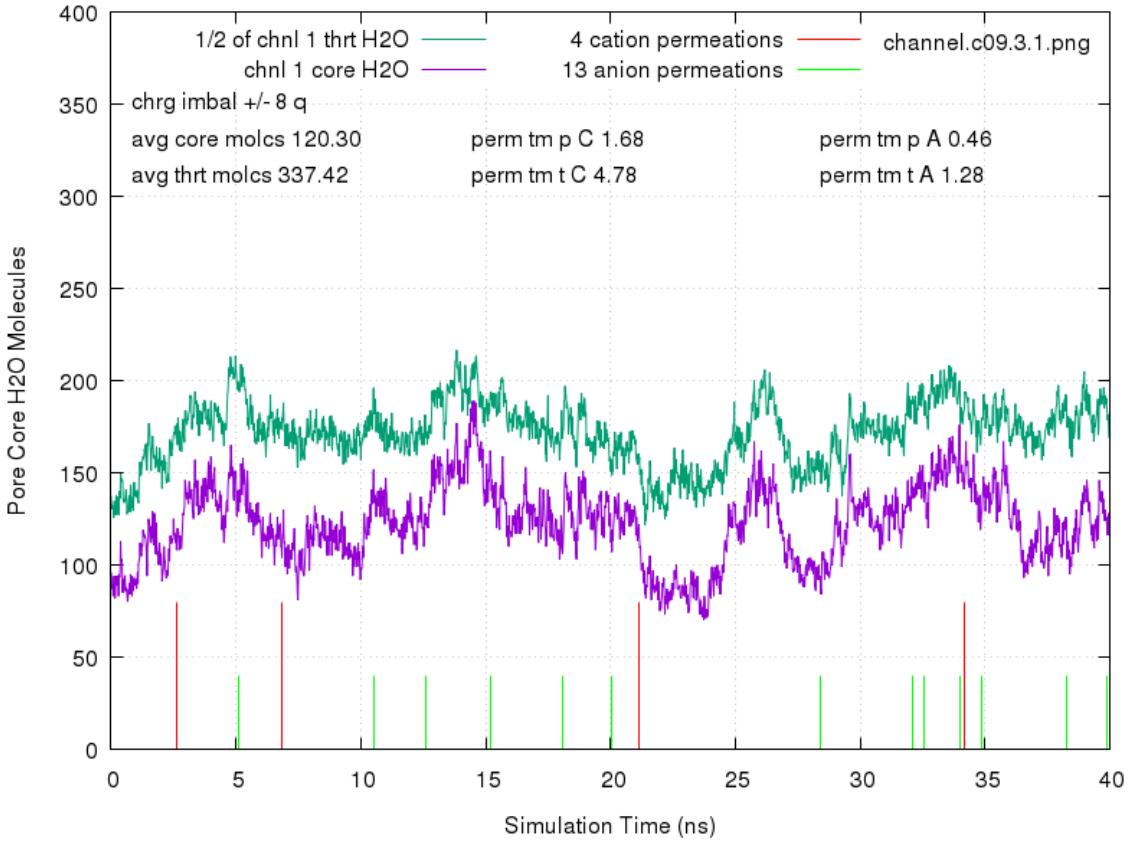

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
16 anion permeations
channel.c0A.Z.0.png
chrg imbal +/- 8 q
avg core molcs 136.50
avg thrt molcs 357.62
perm tm p C 1.40
perm tm t C 3.23
perm tm p A 0.38
perm tm t A 1.06
Pore Core H2O Molecules
Simulation Time (ns)

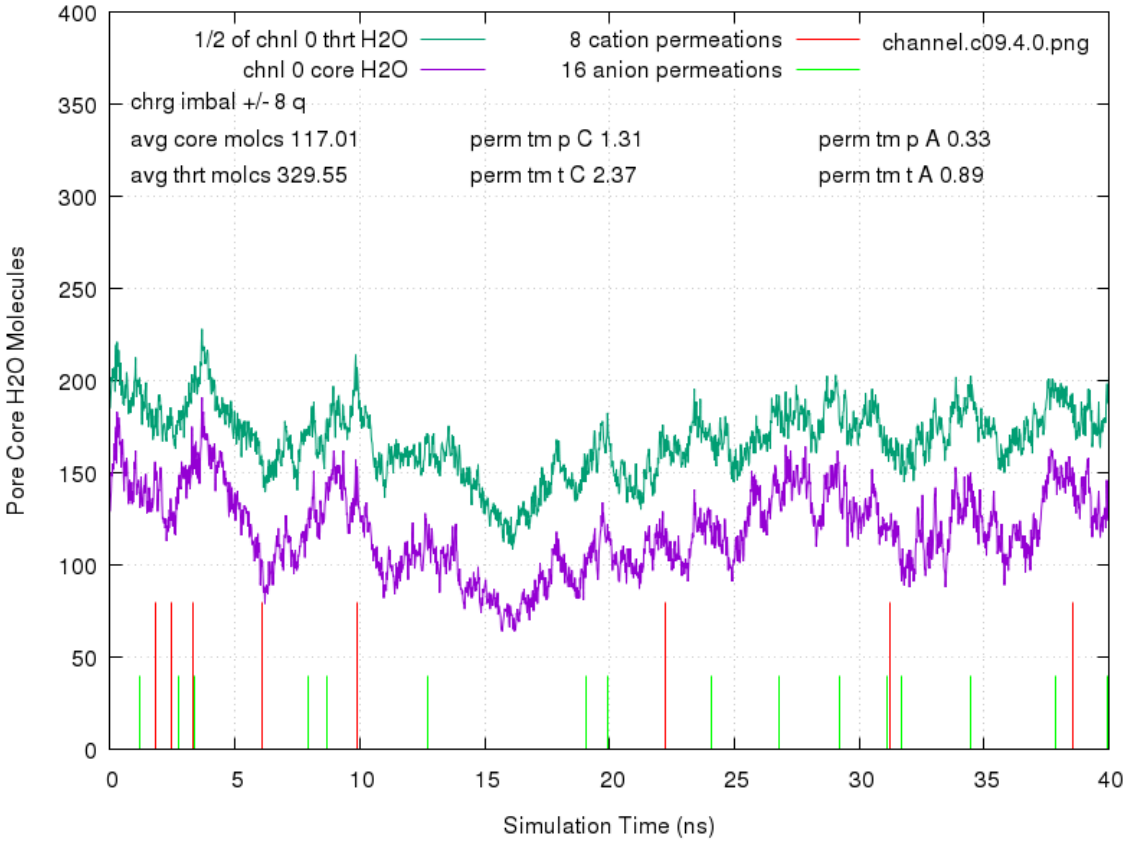

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
7 cation permeations
9 anion permeations
channel.c0A.Z.1.png
chrg imbal +/- 8 q
avg core molcs 111.34
avg thrt molcs 313.26
perm tm p C 2.92
perm tm t C 5.01
perm tm p A 0.48
perm tm t A 1.20
Pore Core H2O Molecules
Simulation Time (ns)

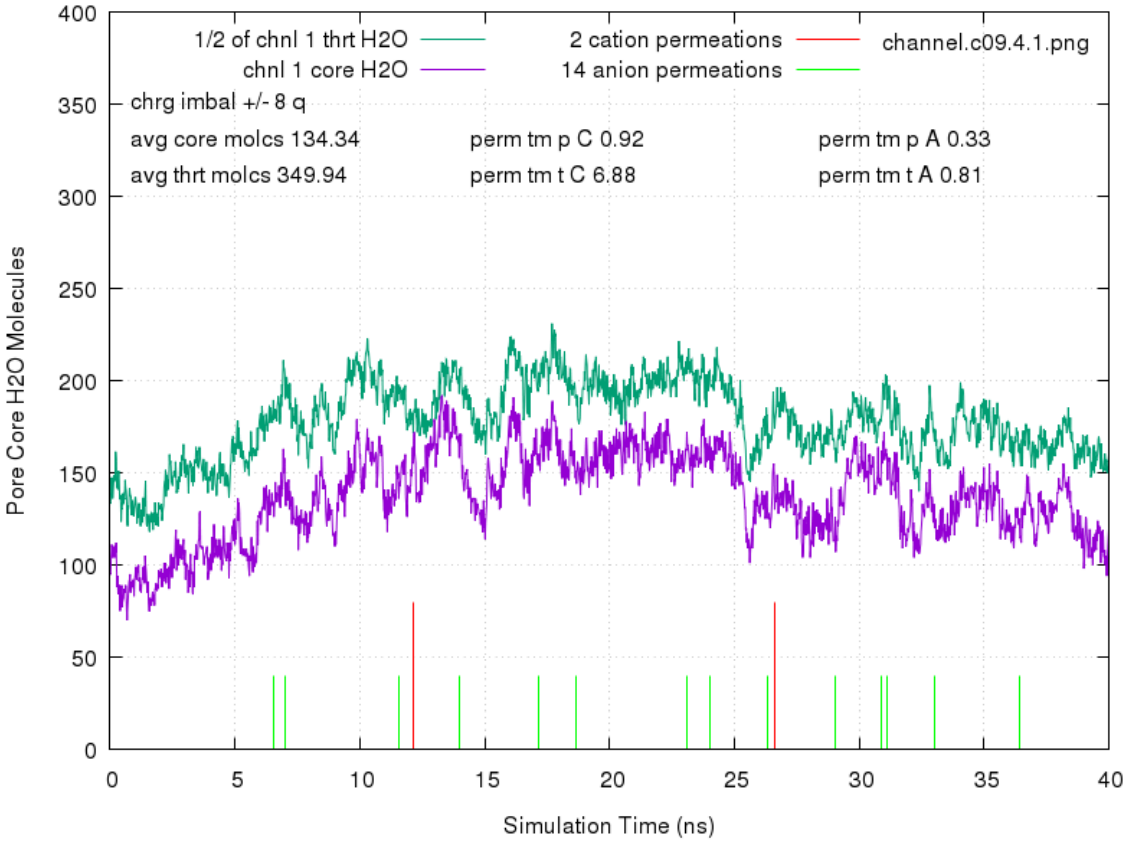

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
7 cation permeations
28 anion permeations
channel.c0A.a.0.png
chrg imbal +/- 8 q
avg core molcs 140.91
avg thrt molcs 362.64
perm tm p C 2.54
perm tm t C 5.54
perm tm p A 0.28
perm tm t A 0.68
Pore Core H2O Molecules
Simulation Time (ns)

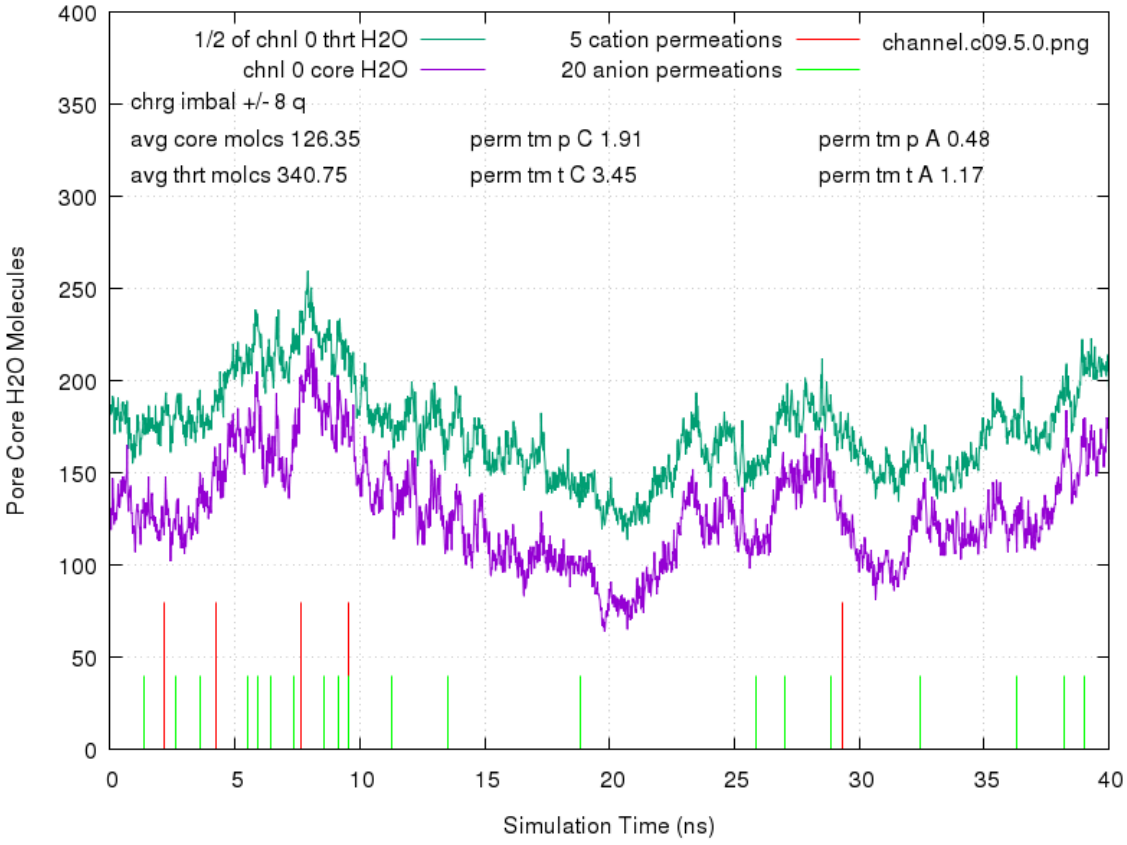

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
10 cation permeations
5 anion permeations
channel.c0A.a.1.png
chrg imbal +/- 8 q
avg core molcs 108.20
avg thrt molcs 307.74
perm tm p C 2.47
perm tm t C 5.30
perm tm p A 0.52
perm tm t A 1.88
Pore Core H2O Molecules
Simulation Time (ns)

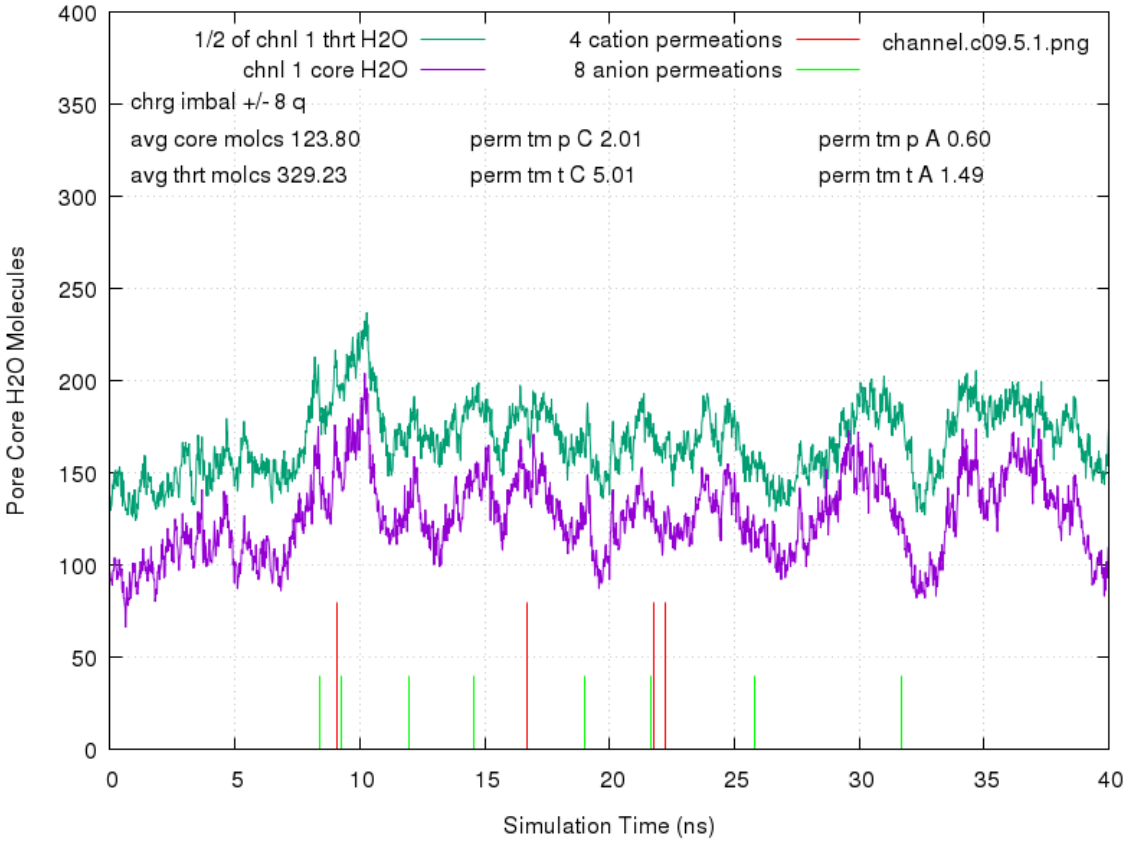

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
6 cation permeations
19 anion permeations
channel.c09.3.0.png
chrg imbal +/- 8 q
avg core molcs 138.64
avg thrt molcs 364.11
perm tm p C 1.78
perm tm t C 3.66
perm tm p A 0.45
perm tm t A 1.30
Pore Core H2O Molecules
Simulation Time (ns)

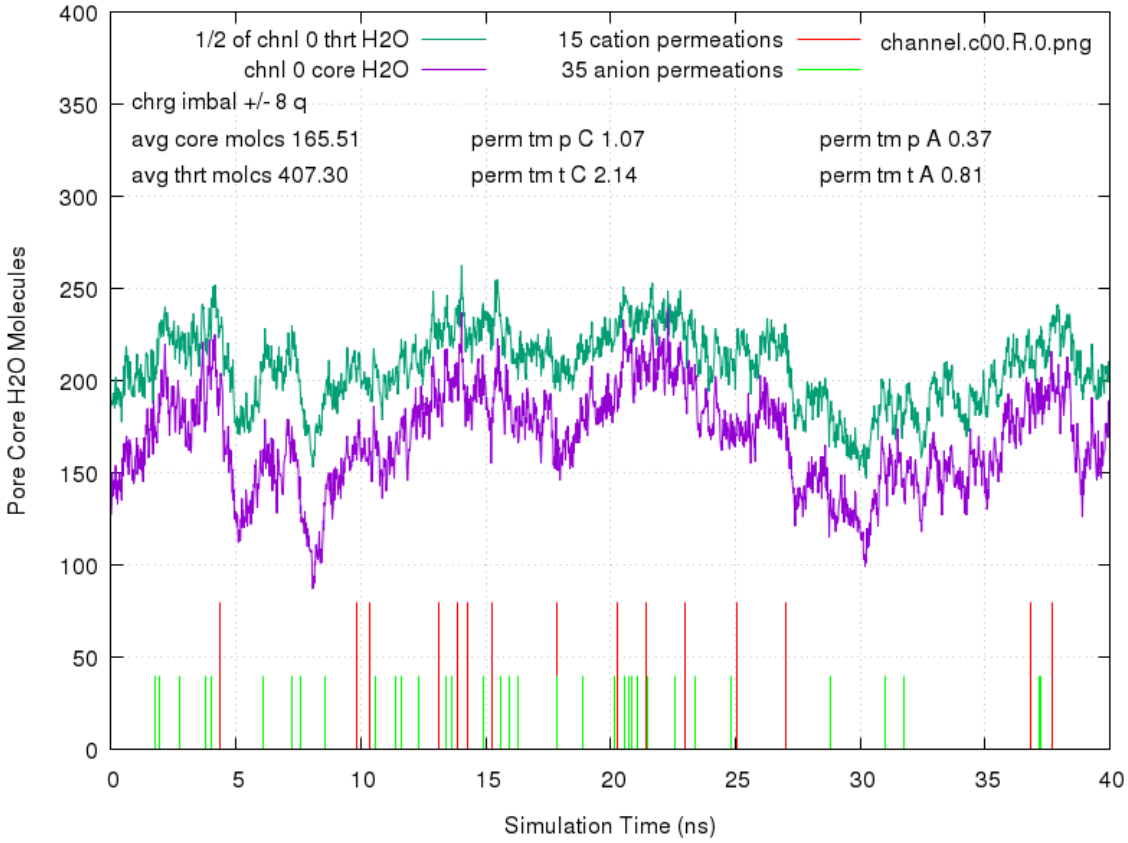

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
13 anion permeations
channel.c09.3.1.png
chrg imbal +/- 8 q
avg core molcs 120.30
avg thrt molcs 337.42
perm tm p C 1.68
perm tm t C 4.78
perm tm p A 0.46
perm tm t A 1.28
Pore Core H2O Molecules
Simulation Time (ns)

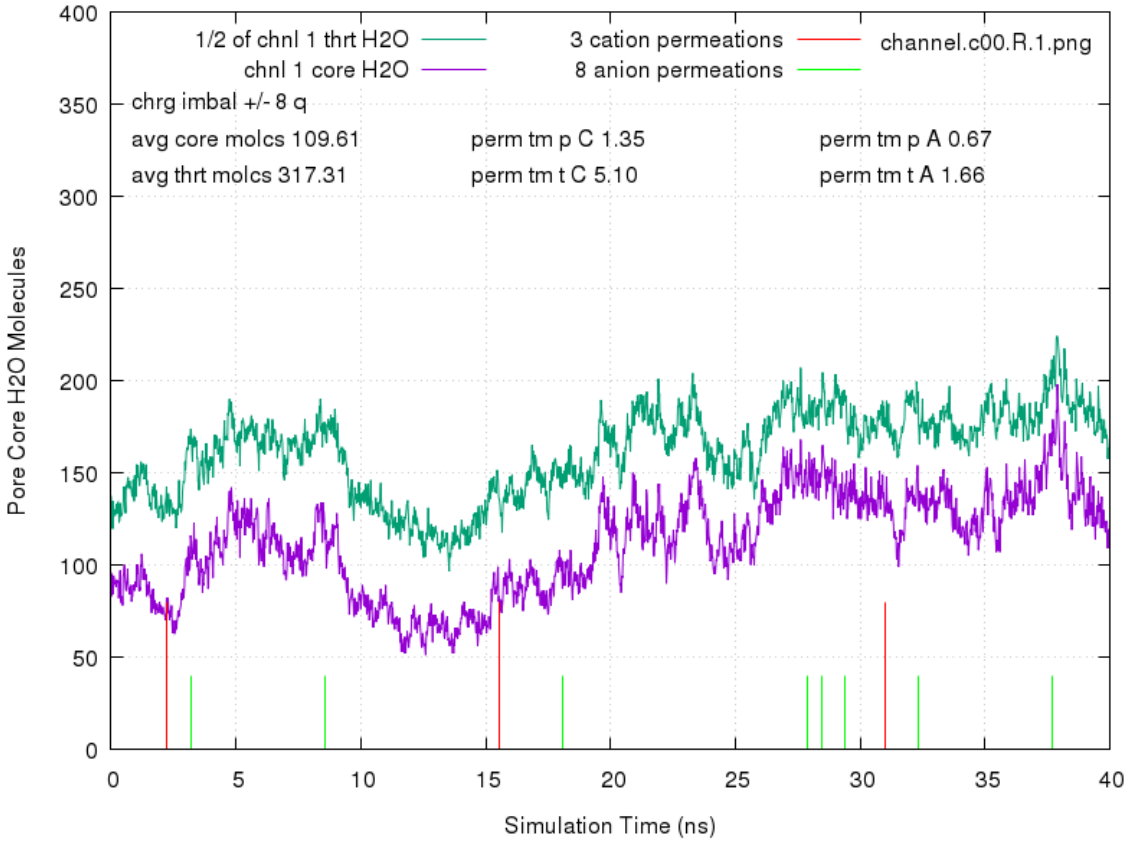

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
16 anion permeations
channel.c09.4.0.png
chrg imbal +/- 8 q
avg core molcs 117.01
avg thrt molcs 329.55
perm tm p C 1.31
perm tm t C 2.37
perm tm p A 0.33
perm tm t A 0.89
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

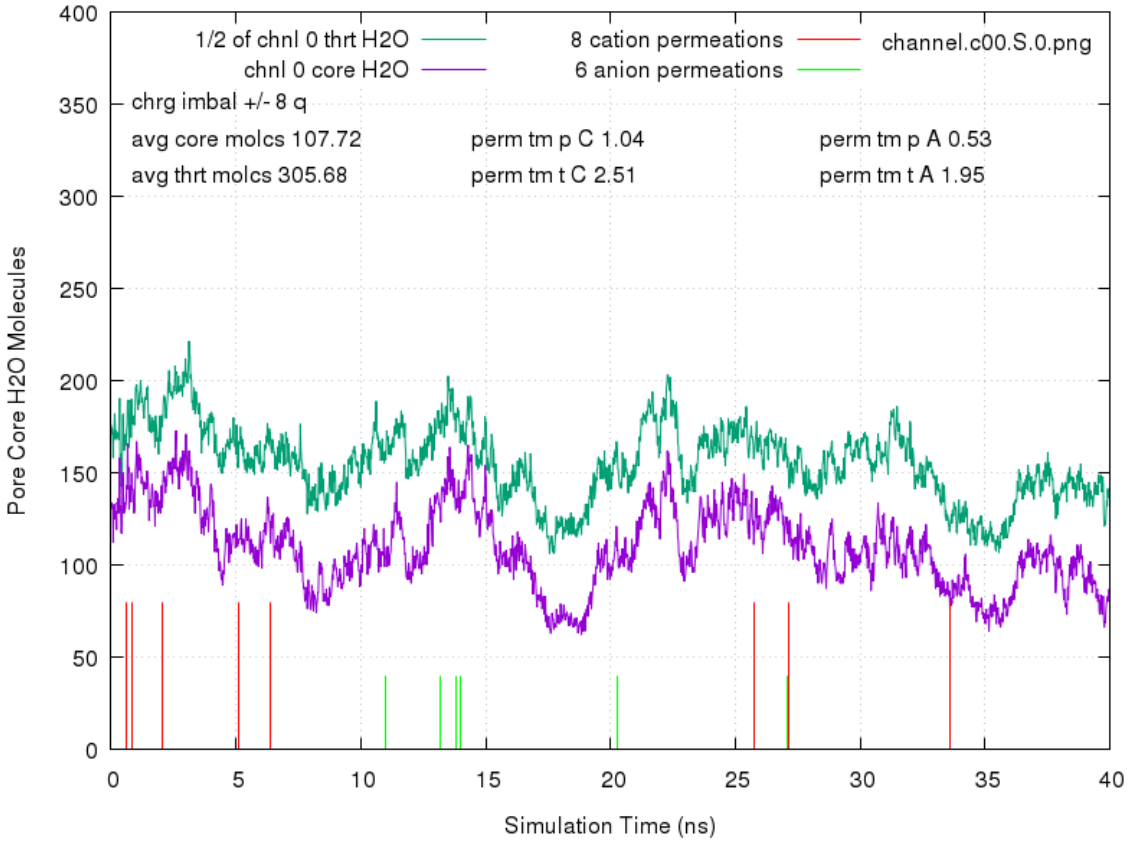

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
2 cation permeations
14 anion permeations
channel.c09.4.1.png
chrg imbal +/- 8 q
avg core molcs 134.34
avg thrt molcs 349.94
perm tm p C 0.92
perm tm t C 6.88
perm tm p A 0.33
perm tm t A 0.81
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

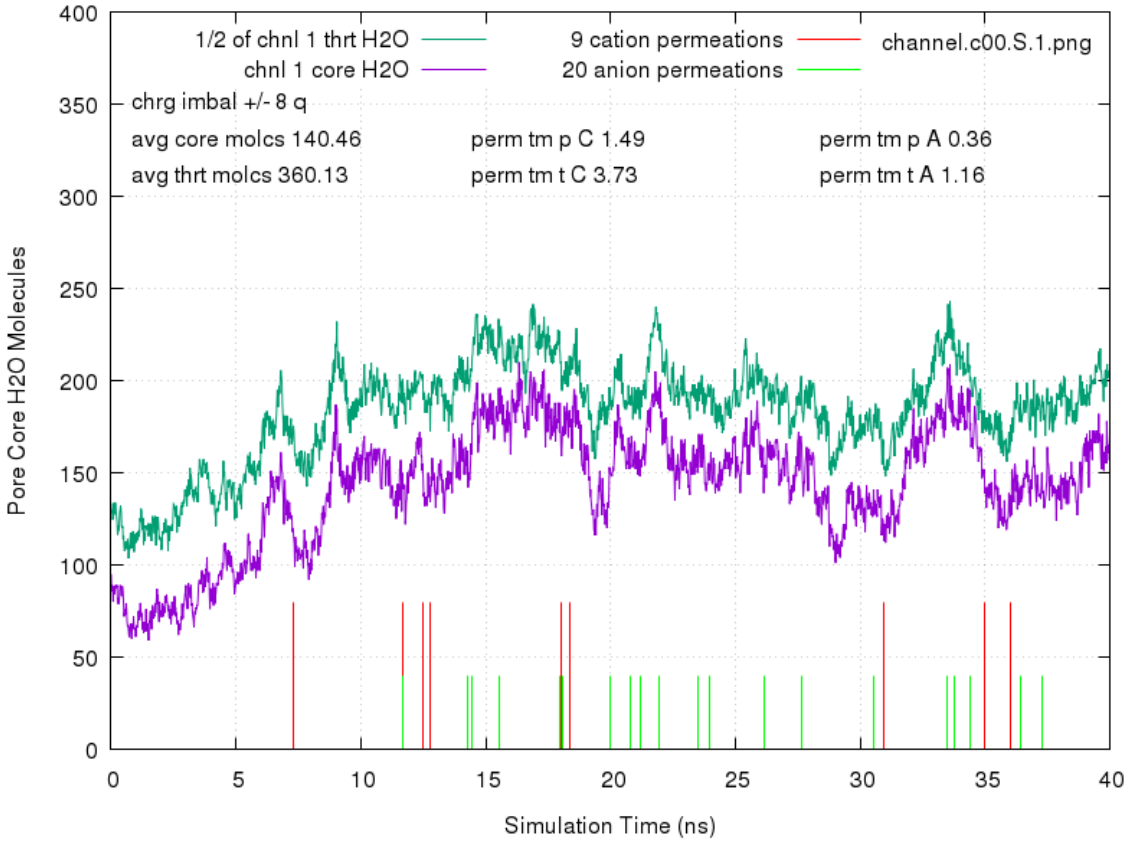

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
20 anion permeations
channel.c09.5.0.png
chrg imbal +/- 8 q
avg core molcs 126.35
avg thrt molcs 340.75
perm tm p C 1.91
perm tm t C 3.45
perm tm p A 0.48
perm tm t A 1.17
Pore Core H2O Molecules
Simulation Time (ns)

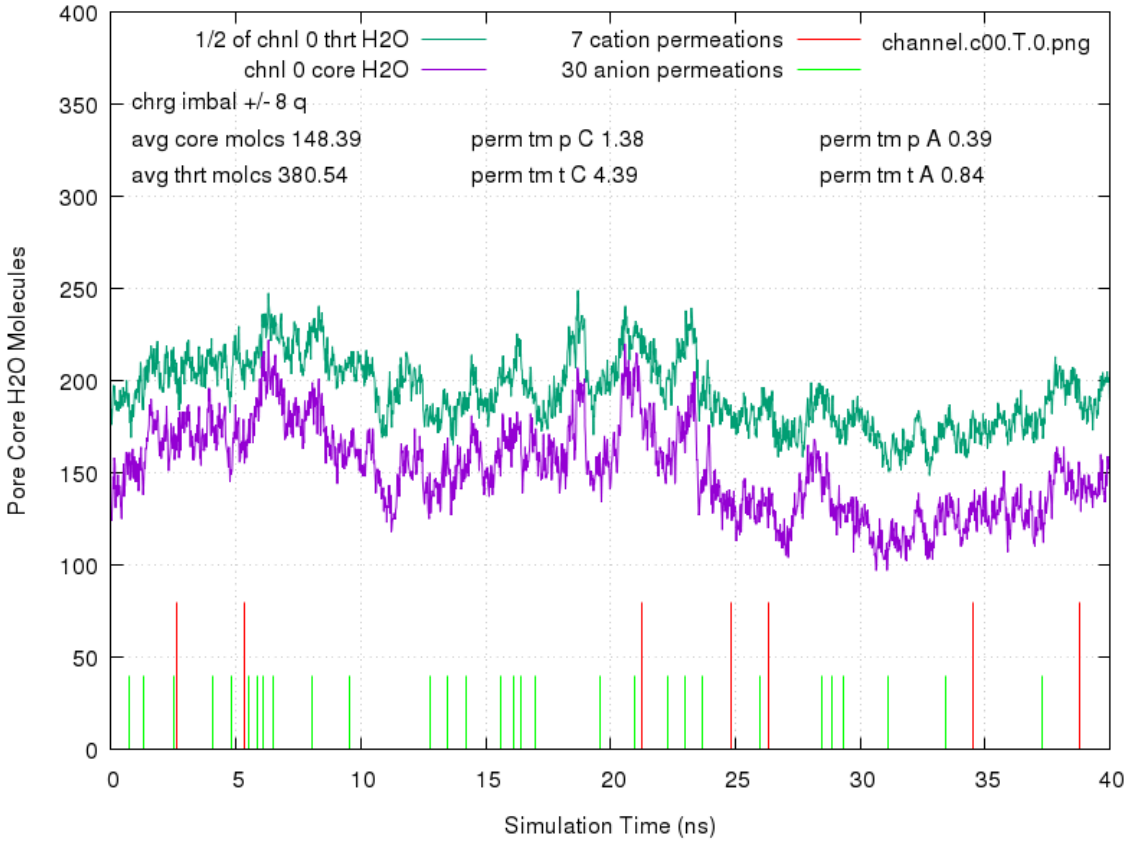

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
8 anion permeations
channel.c09.5.1.png
chrg imbal +/- 8 q
avg core molcs 123.80
avg thrt molcs 329.23
perm tm p C 2.01
perm tm t C 5.01
perm tm p A 0.60
perm tm t A 1.49
Pore Core H2O Molecules
Simulation Time (ns)

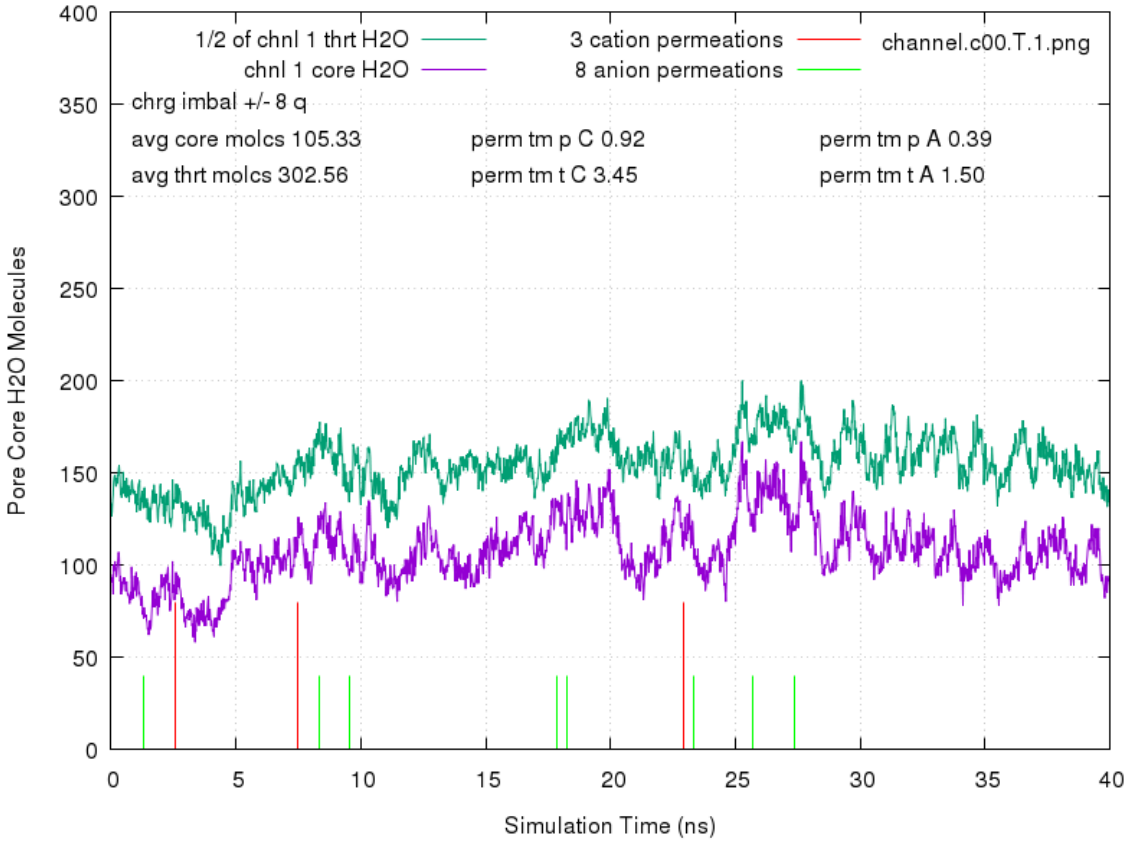

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
15 cation permeations
35 anion permeations
channel.c00.R.0.png
chrg imbal +/- 8 q
avg core molcs 165.51
avg thrt molcs 407.30
perm tm p C 1.07
perm tm t C 2.14
perm tm p A 0.37
perm tm t A 0.81
Pore Core H2O Molecules
Simulation Time (ns)

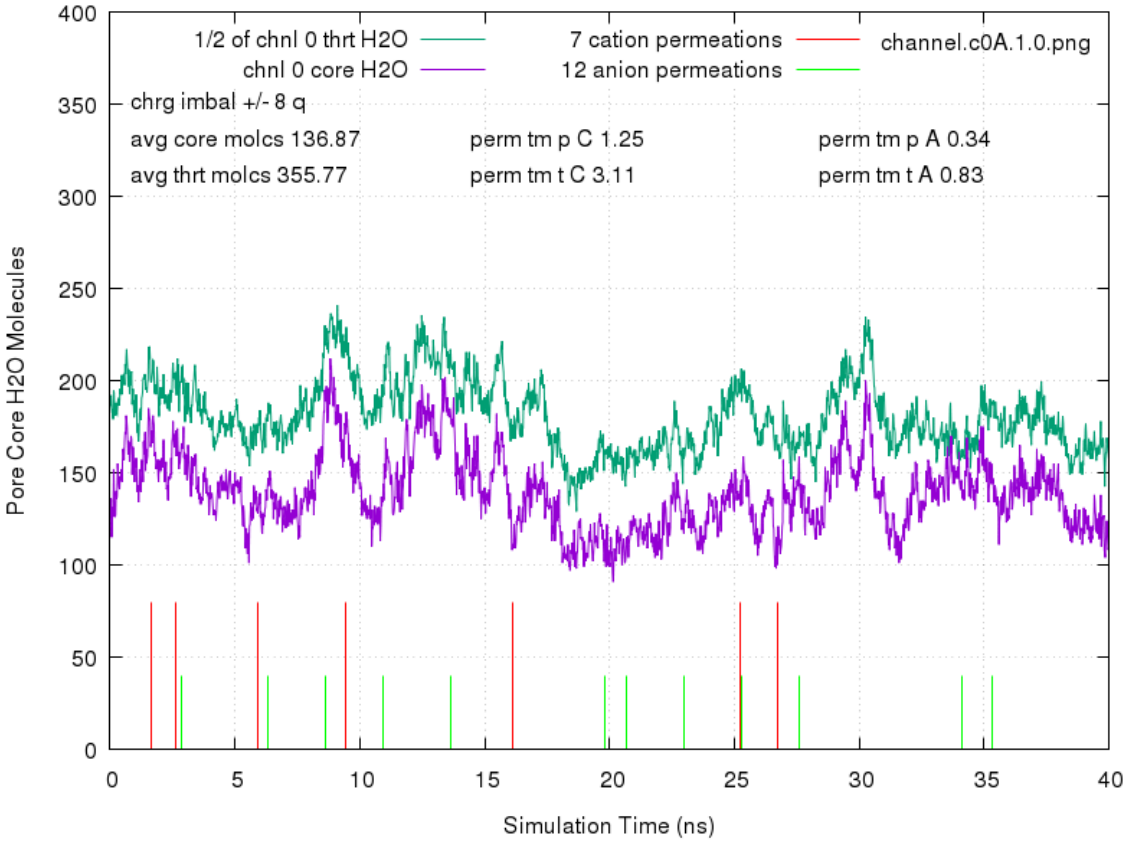

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
8 anion permeations
channel.c00.R.1.png
chrg imbal +/- 8 q
avg core molcs 109.61
avg thrt molcs 317.31
perm tm p C 1.35
perm tm t C 5.10
perm tm p A 0.67
perm tm t A 1.66
Pore Core H2O Molecules
Simulation Time (ns)

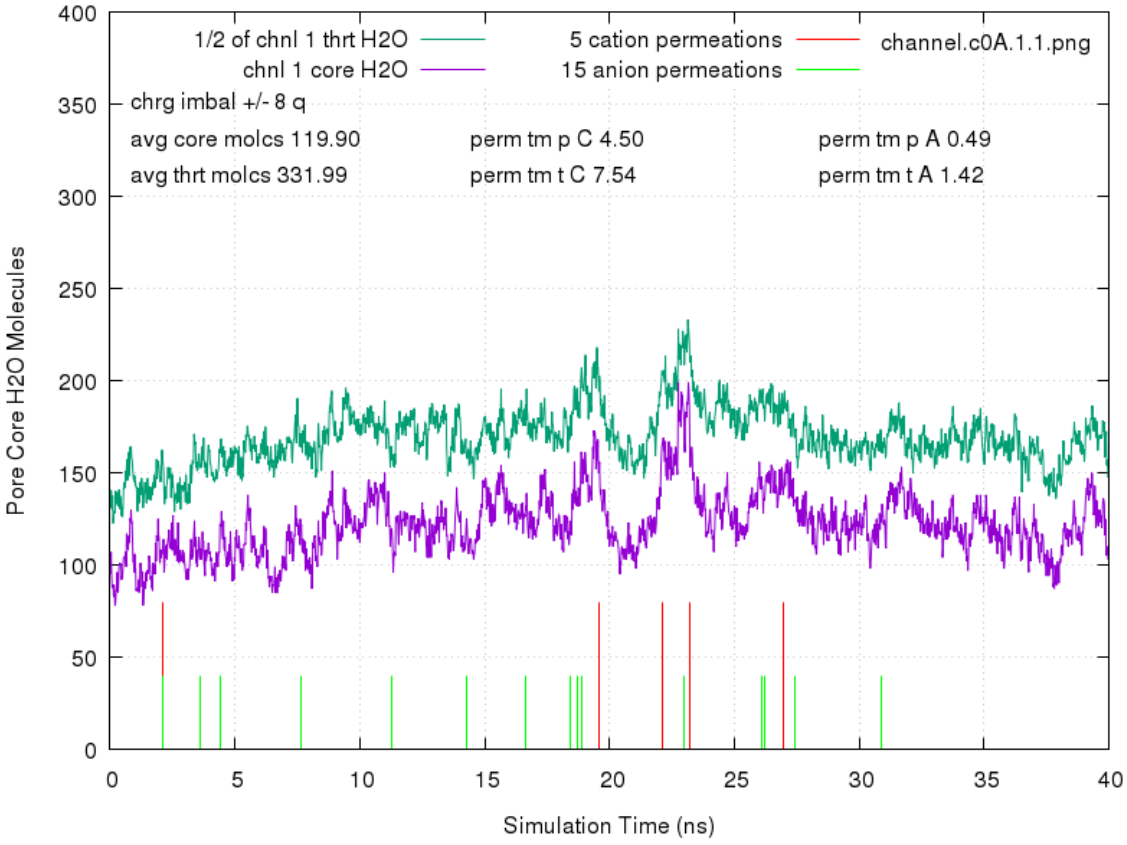

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
6 anion permeations
channel.c00.S.0.png
chrg imbal +/- 8 q
avg core molcs 107.72
avg thrt molcs 305.68
perm tm p C 1.04
perm tm t C 2.51
perm tm p A 0.53
perm tm t A 1.95
Pore Core H2O Molecules
Simulation Time (ns)

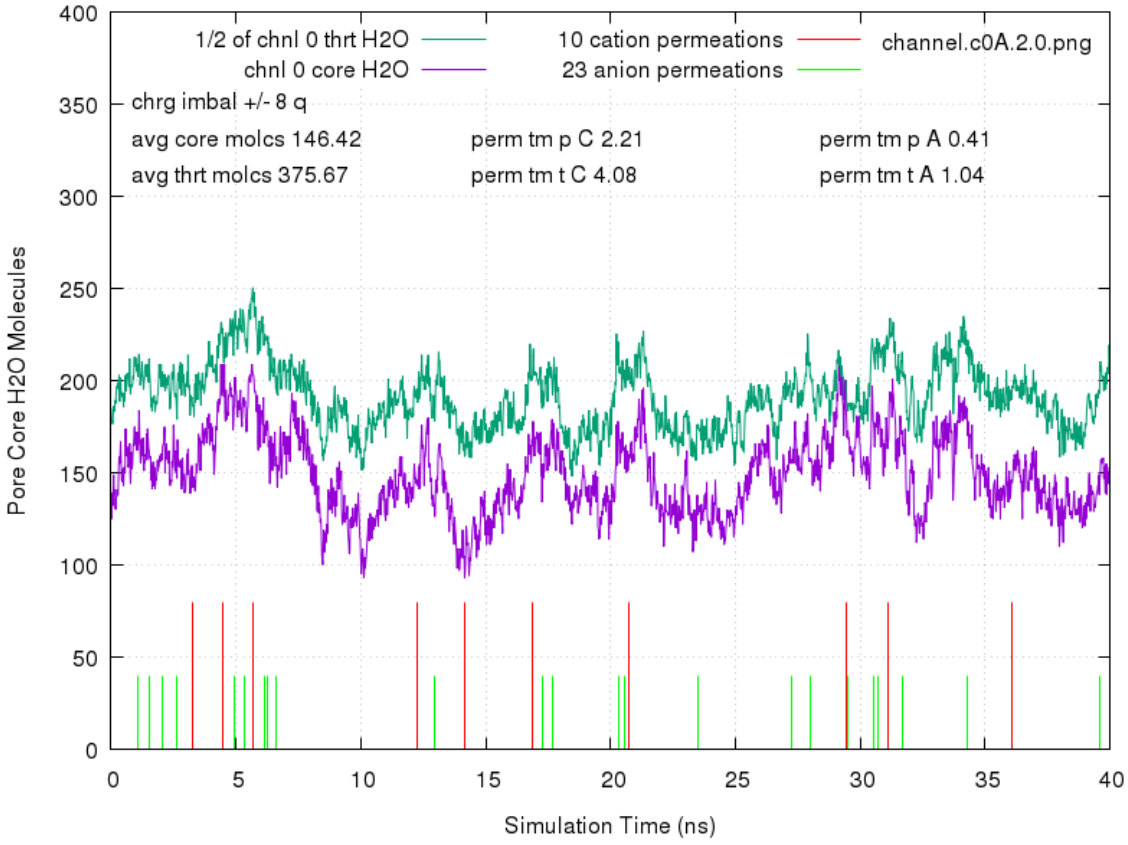

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
9 cation permeations
20 anion permeations
channel.c00.S.1.png
chrg imbal +/- 8 q
avg core molcs 140.46
avg thrt molcs 360.13
perm tm p C 1.49
perm tm t C 3.73
perm tm p A 0.36
perm tm t A 1.16
Pore Core H2O Molecules
Simulation Time (ns)

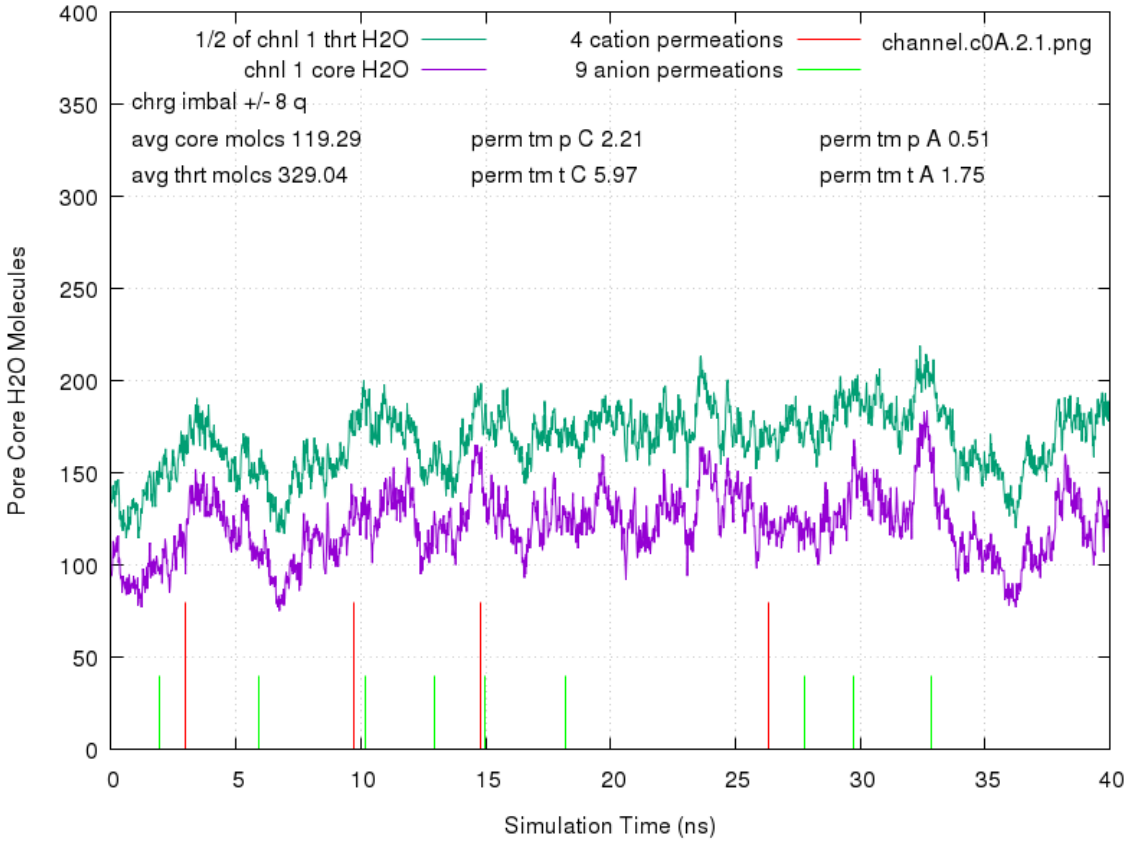

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
7 cation permeations
30 anion permeations
channel.c00.T.0.png
chrg imbal +/- 8 q
avg core molcs 148.39
avg thrt molcs 380.54
perm tm p C 1.38
perm tm t C 4.39
perm tm p A 0.39
perm tm t A 0.84
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

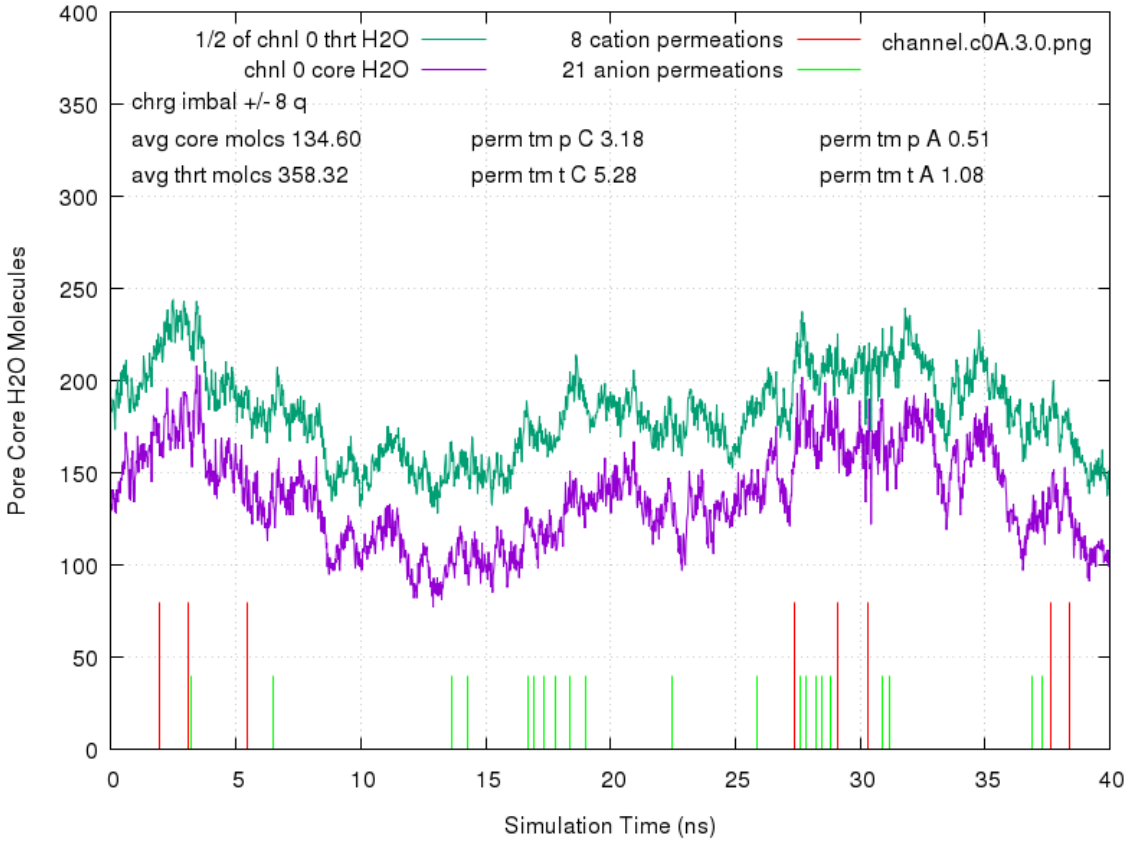

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
3 cation permeations
8 anion permeations
channel.c00.T.1.png
chrg imbal +/- 8 q
avg core molcs 105.33
avg thrt molcs 302.56
perm tm p C 0.92
perm tm t C 3.45
perm tm p A 0.39
perm tm t A 1.50
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

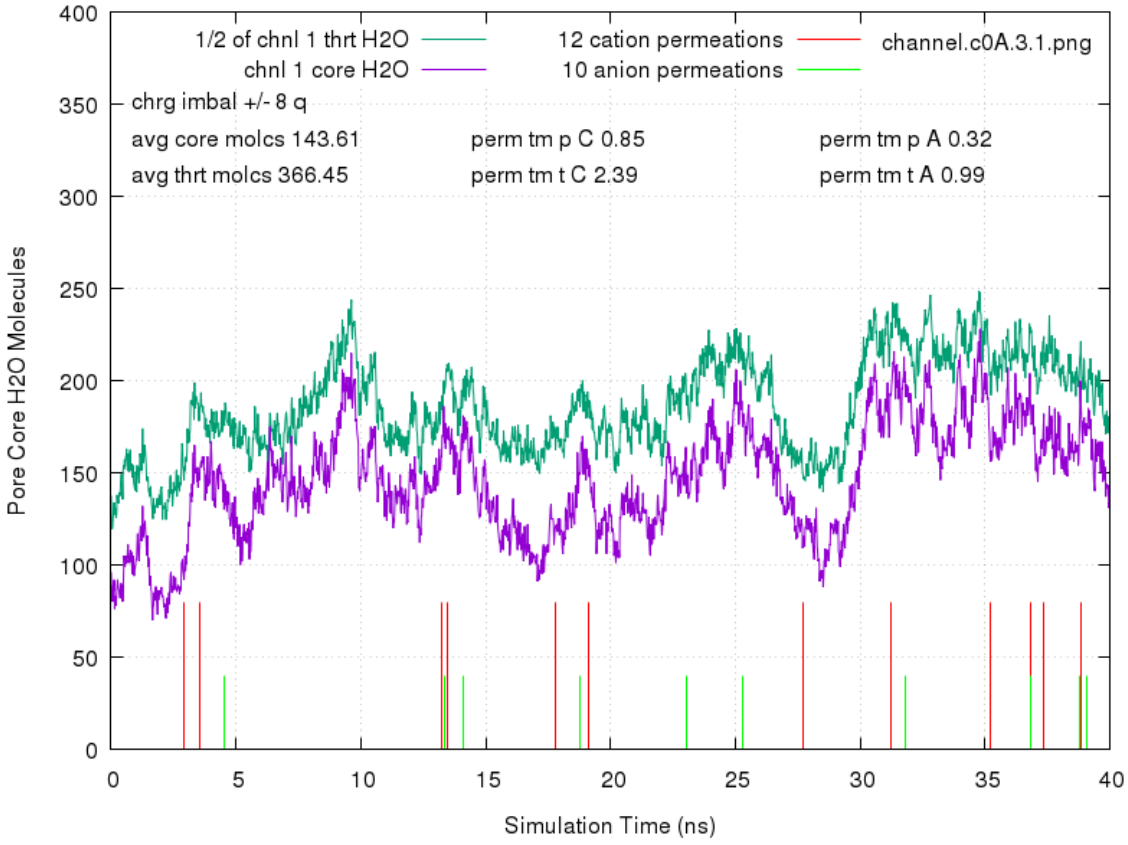

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
7 cation permeations
12 anion permeations
channel.c0A.1.0.png
chrg imbal +/- 8 q
avg core molcs 136.87
avg thrt molcs 355.77
perm tm p C 1.25
perm tm t C 3.11
perm tm p A 0.34
perm tm t A 0.83
Pore Core H2O Molecules
Simulation Time (ns)

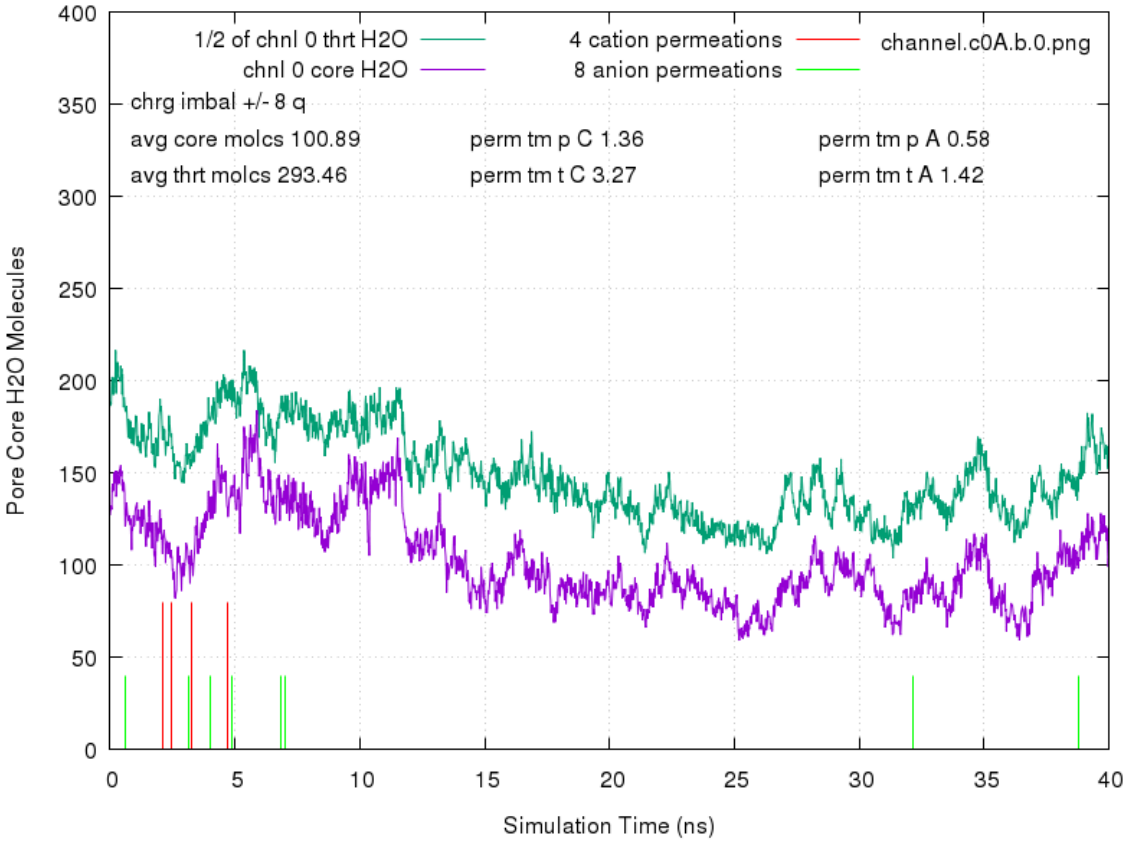

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
5 cation permeations
15 anion permeations
channel.c0A.1.1.png
chrg imbal +/- 8 q
avg core molcs 119.90
avg thrt molcs 331.99
perm tm p C 4.50
perm tm t C 7.54
perm tm p A 0.49
perm tm t A 1.42
Pore Core H2O Molecules
Simulation Time (ns)

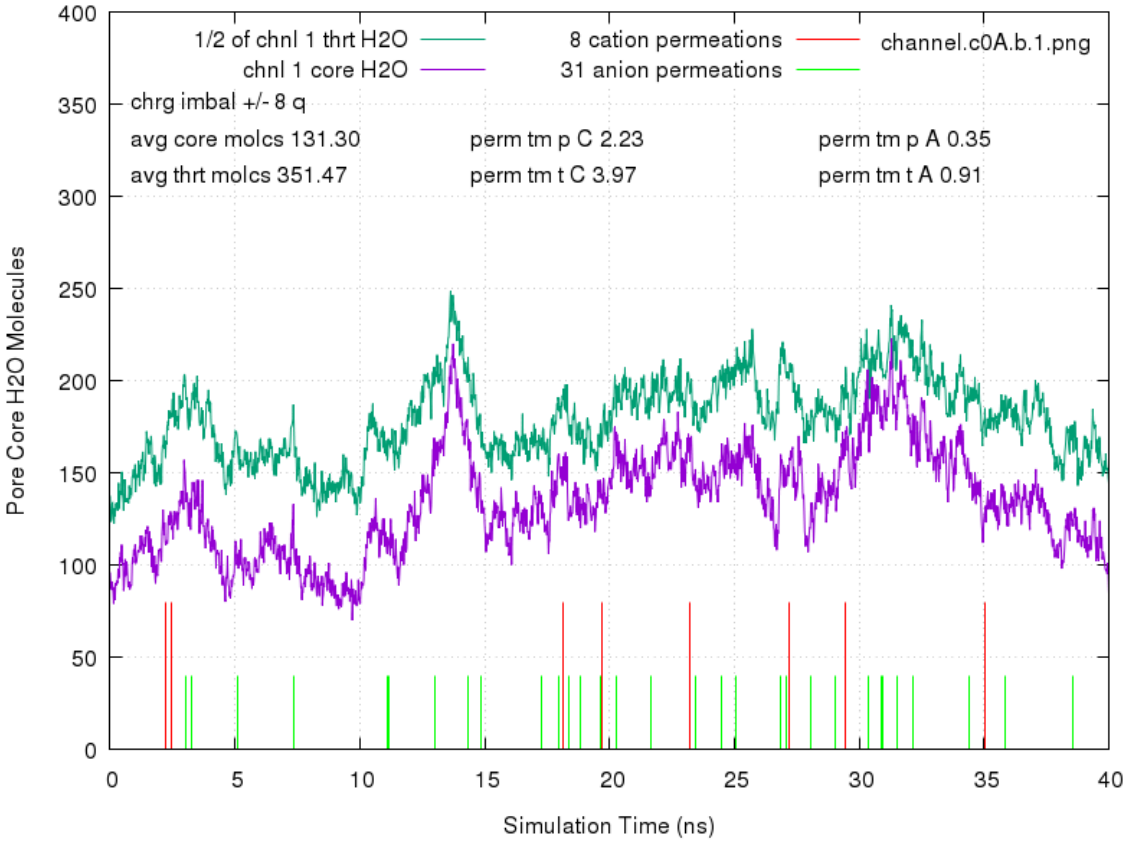

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
10 cation permeations
23 anion permeations
channel.c0A.2.0.png
chrg imbal +/- 8 q
avg core molcs 146.42
avg thrt molcs 375.67
perm tm p C 2.21
perm tm t C 4.08
perm tm p A 0.41
perm tm t A 1.04
Pore Core H2O Molecules
Simulation Time (ns)

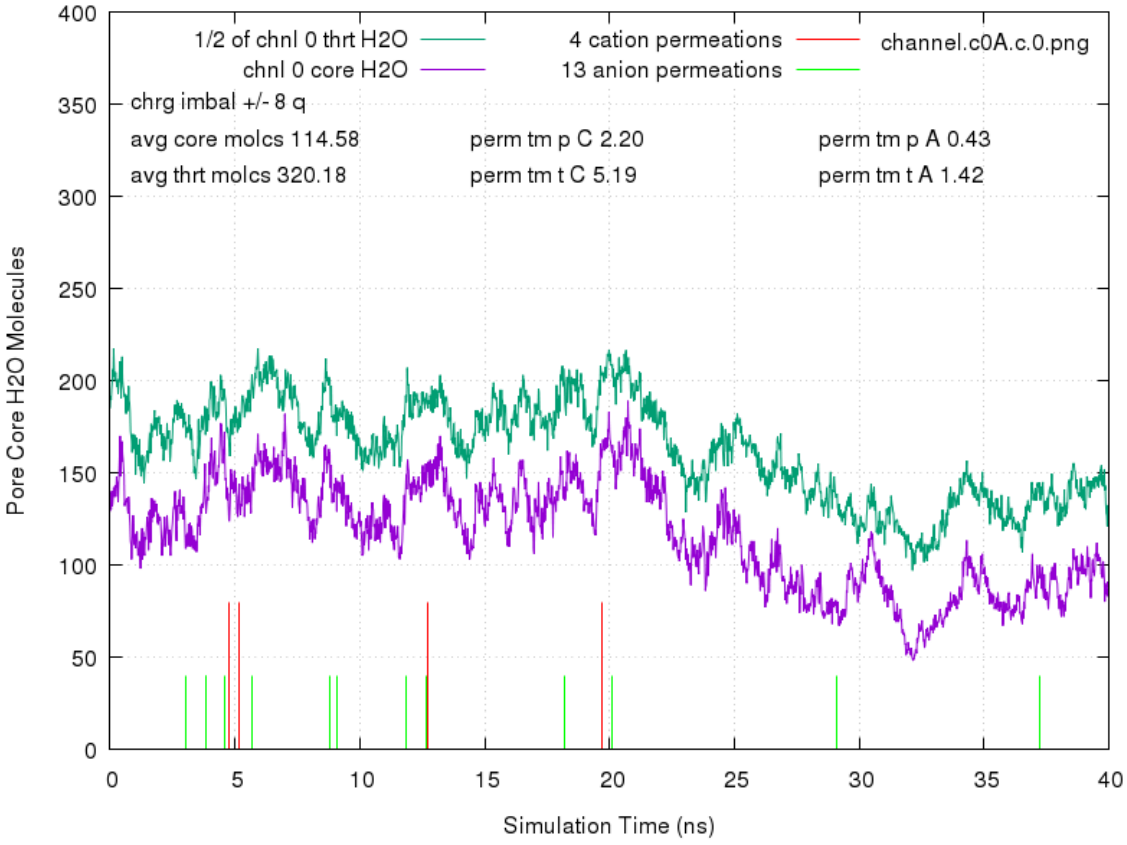

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
9 anion permeations
channel.c0A.2.1.png
chrg imbal +/- 8 q
avg core molcs 119.29
avg thrt molcs 329.04
perm tm p C 2.21
perm tm t C 5.97
perm tm p A 0.51
perm tm t A 1.75
Pore Core H2O Molecules
Simulation Time (ns)

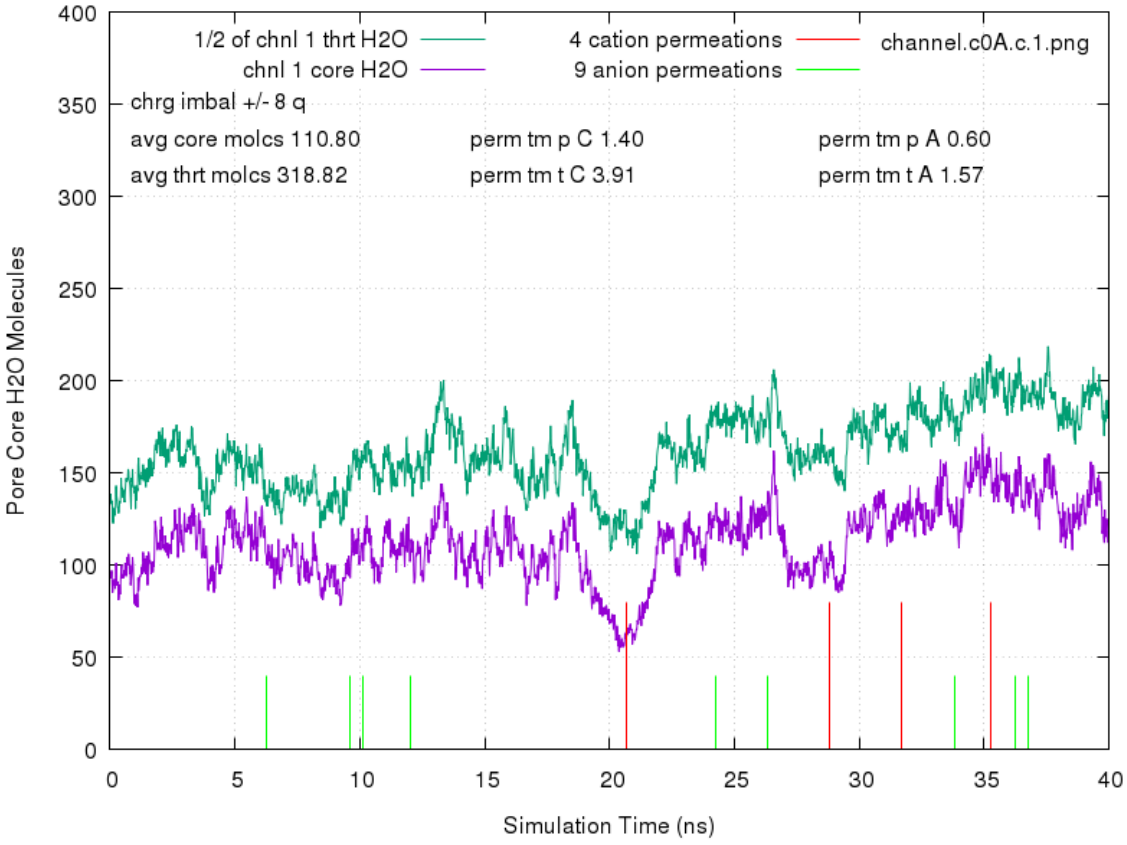

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
21 anion permeations
channel.c0A.3.0.png
chrg imbal +/- 8 q
avg core molcs 134.60
avg thrt molcs 358.32
perm tm p C 3.18
perm tm t C 5.28
perm tm p A 0.51
perm tm t A 1.08
Pore Core H2O Molecules
Simulation Time (ns)

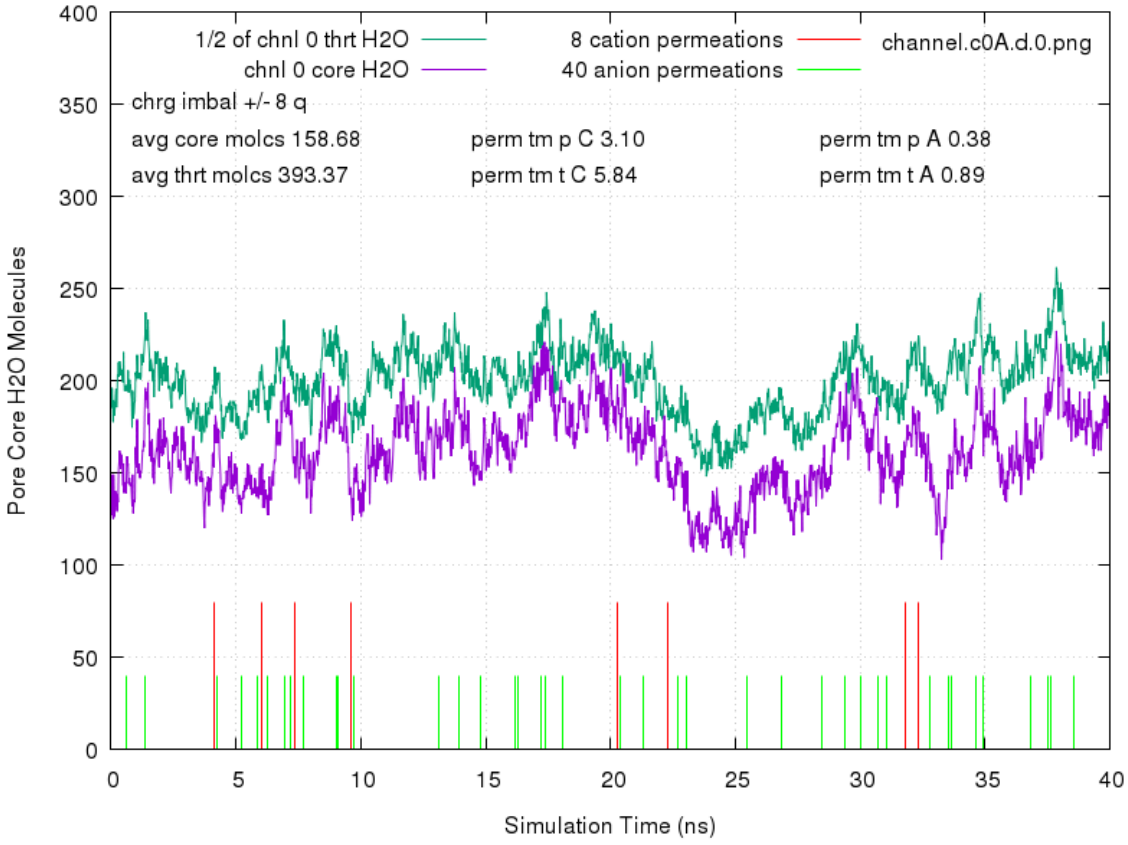

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
12 cation permeations
10 anion permeations
channel.c0A.3.1.png
chrg imbal +/- 8 q
avg core molcs 143.61
avg thrt molcs 366.45
perm tm p C 0.85
perm tm t C 2.39
perm tm p A 0.32
perm tm t A 0.99
Pore Core H2O Molecules
Simulation Time (ns)

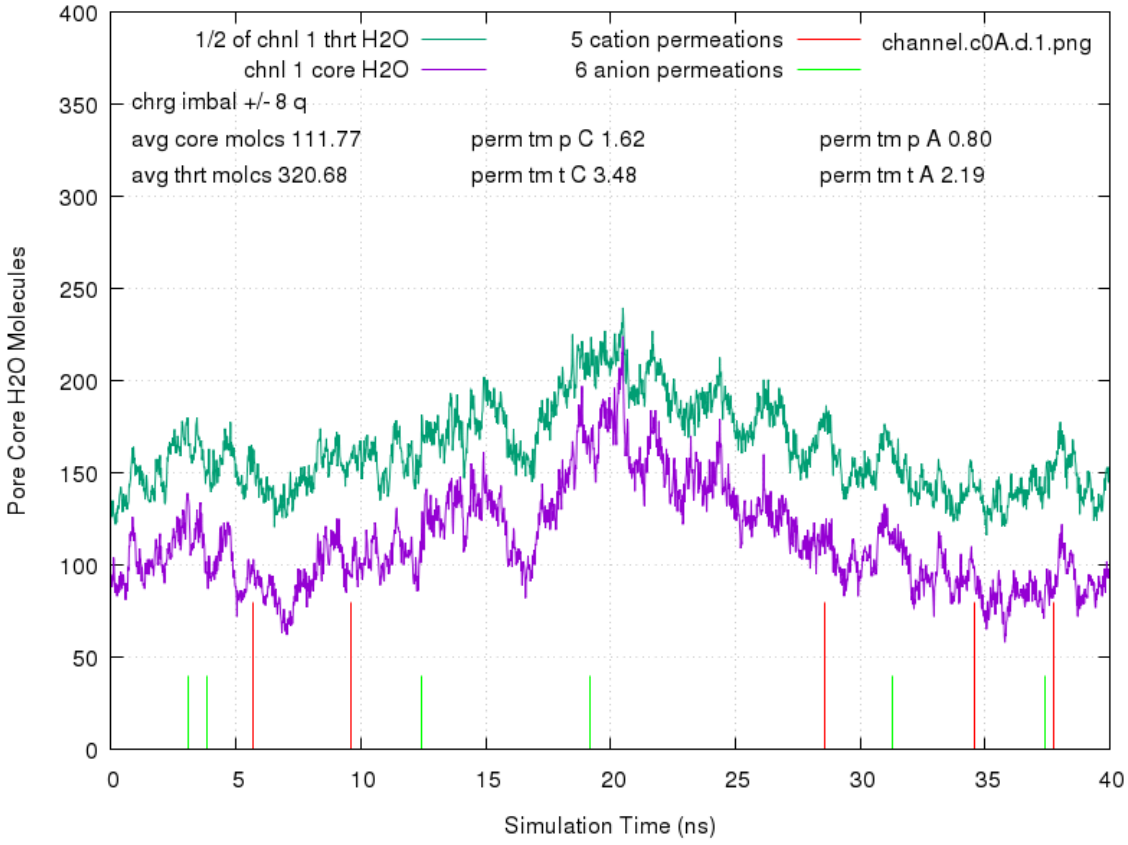

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
4 cation permeations
8 anion permeations
channel.c0A.b.0.png
chrg imbal +/- 8 q
avg core molcs 100.89
avg thrt molcs 293.46
perm tm p C 1.36
perm tm t C 3.27
perm tm p A 0.58
perm tm t A 1.42
Pore Core H2O Molecules
Simulation Time (ns)

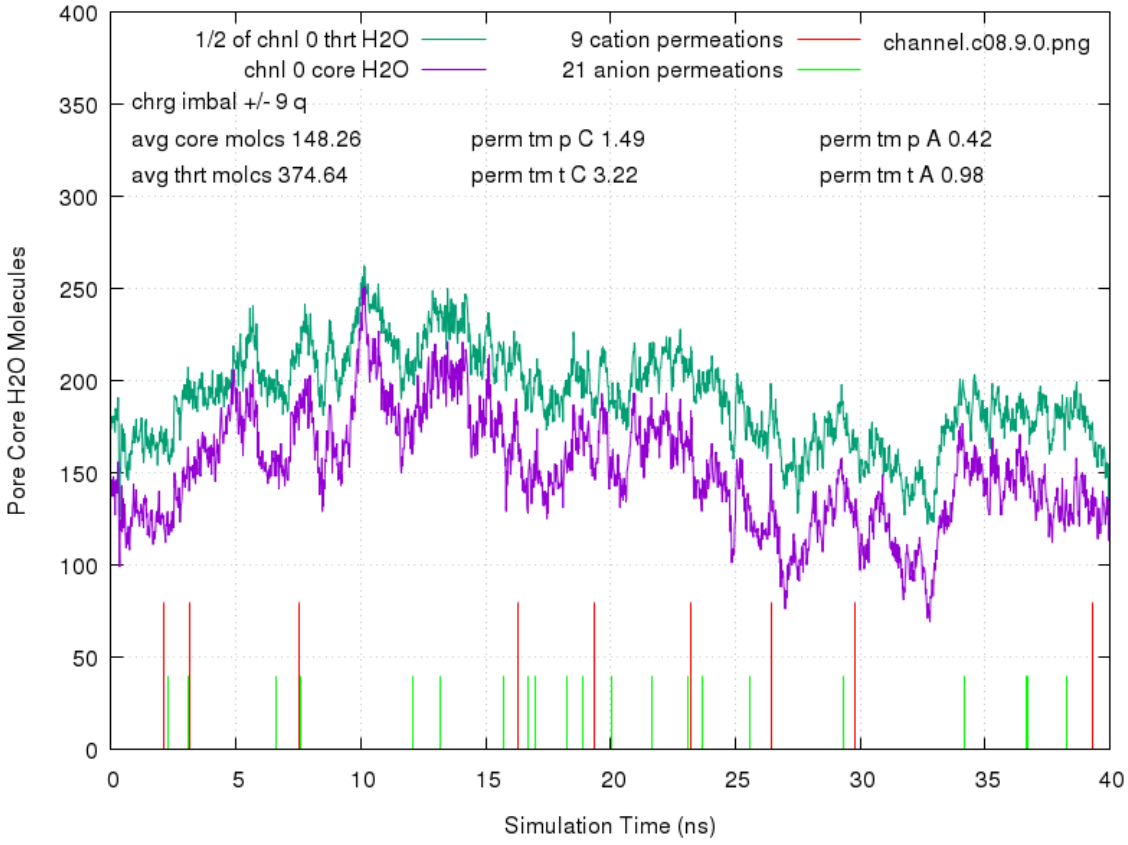

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
8 cation permeations
31 anion permeations
channel.c0A.b.1.png
chrg imbal +/- 8 q
avg core molcs 131.30
avg thrt molcs 351.47
perm tm p C 2.23
perm tm t C 3.97
perm tm p A 0.35
perm tm t A 0.91
Pore Core H2O Molecules
Simulation Time (ns)

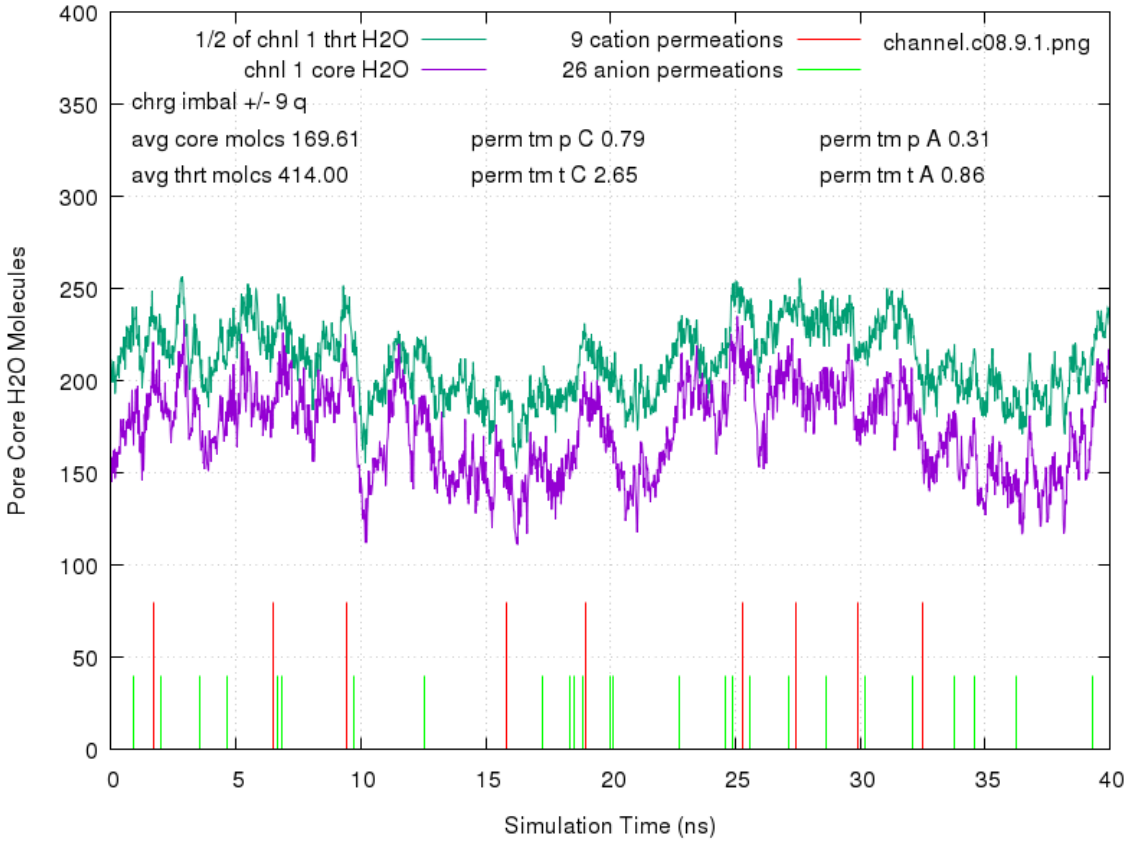

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
4 cation permeations
13 anion permeations
channel.c0A.c.0.png
chrg imbal +/- 8 q
avg core molcs 114.58
avg thrt molcs 320.18
perm tm p C 2.20
perm tm t C 5.19
perm tm p A 0.43
perm tm t A 1.42
Pore Core H2O Molecules
Simulation Time (ns)

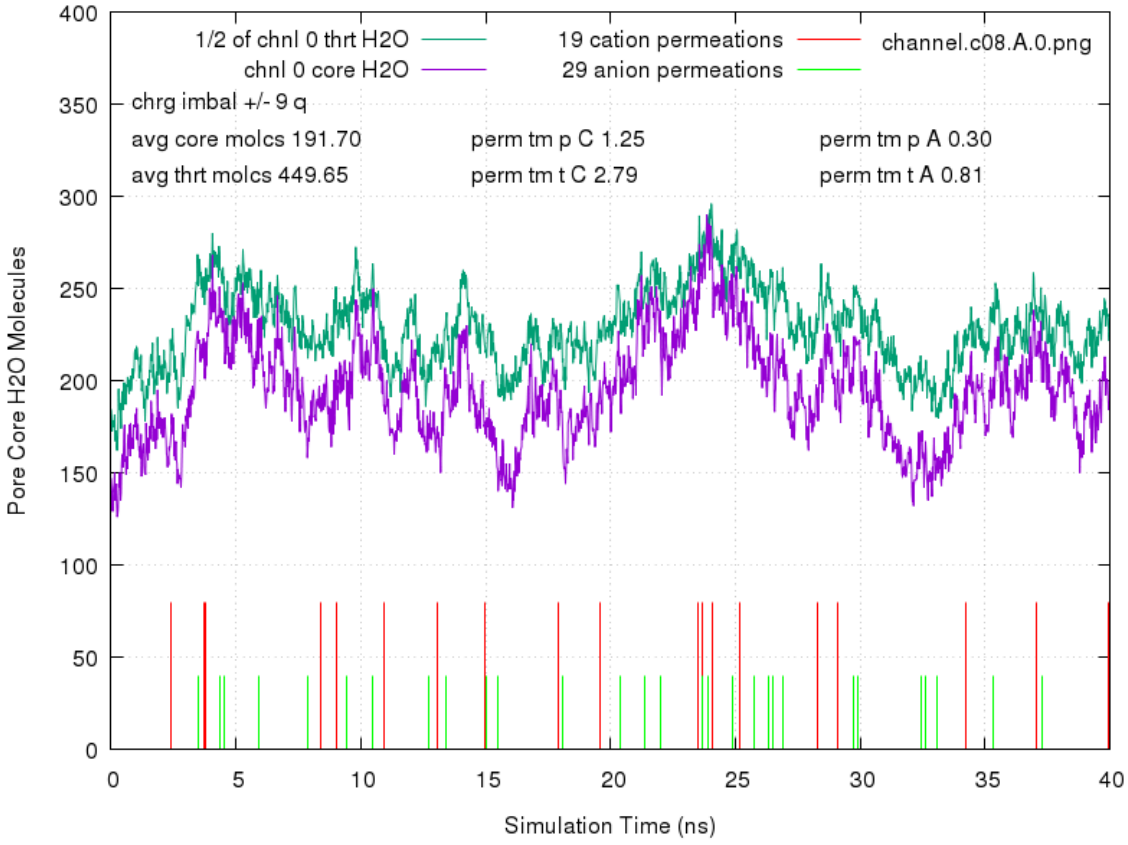

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
4 cation permeations
9 anion permeations
channel.c0A.c.1.png
chrg imbal +/- 8 q
avg core molcs 110.80
avg thrt molcs 318.82
perm tm p C 1.40
perm tm t C 3.91
perm tm p A 0.60
perm tm t A 1.57
Pore Core H2O Molecules
Simulation Time (ns)

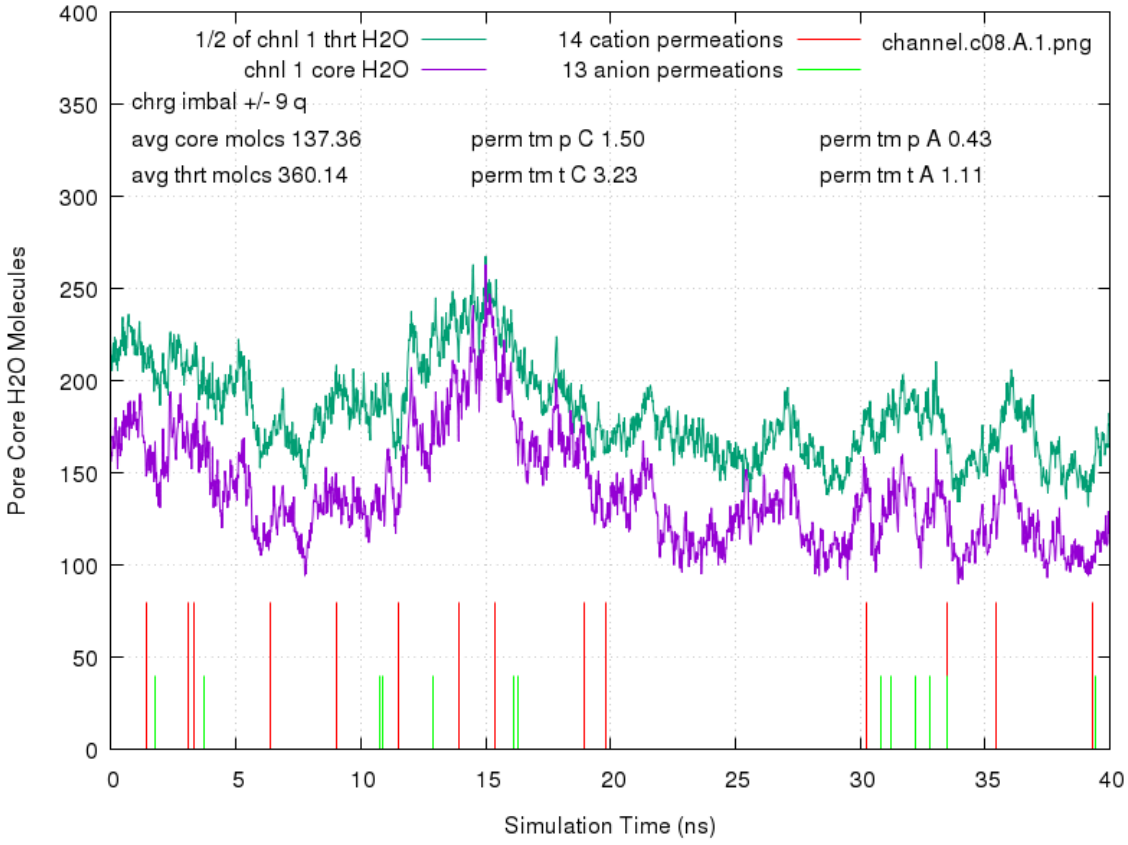

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
40 anion permeations
channel.c0A.d.0.png
chrg imbal +/- 8 q
avg core molcs 158.68
avg thrt molcs 393.37
perm tm p C 3.10
perm tm t C 5.84
perm tm p A 0.38
perm tm t A 0.89
Pore Core H2O Molecules
Simulation Time (ns)

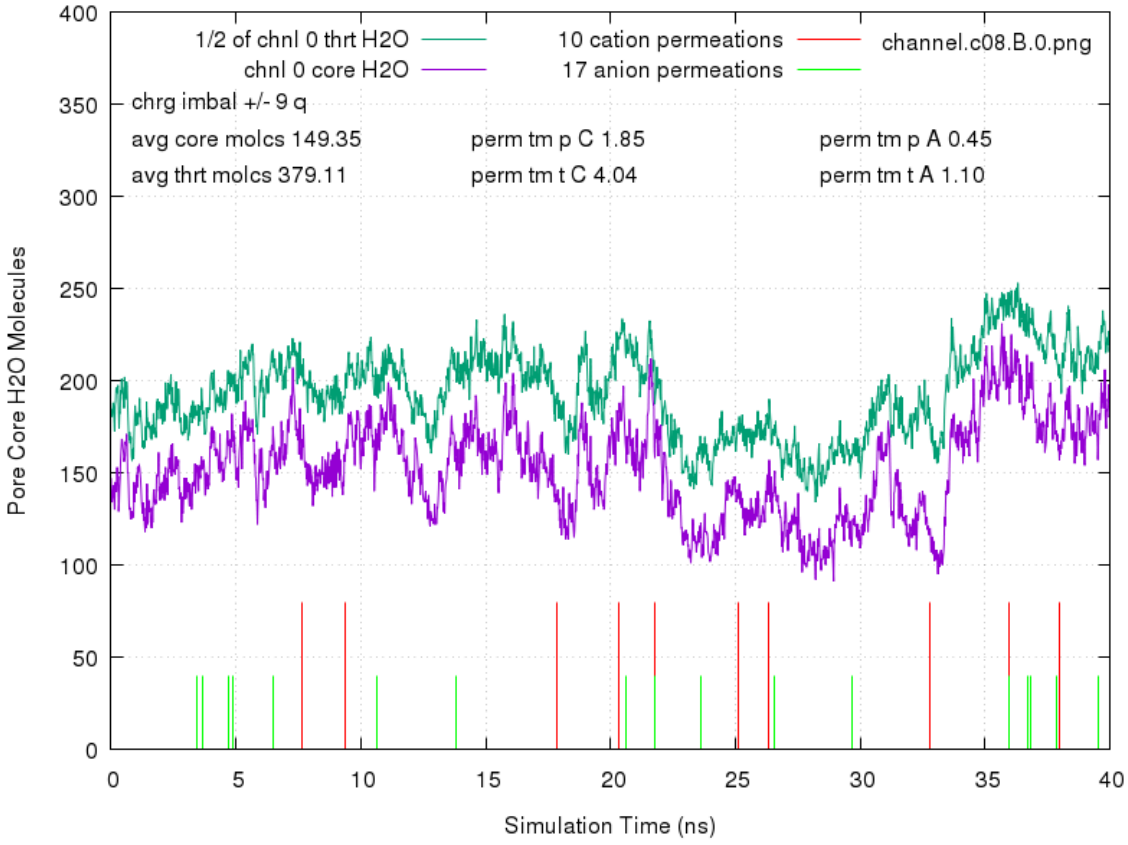

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
5 cation permeations
6 anion permeations
channel.c0A.d.1.png
chrg imbal +/- 8 q
avg core molcs 111.77
avg thrt molcs 320.68
perm tm p C 1.62
perm tm t C 3.48
perm tm p A 0.80
perm tm t A 2.19
Pore Core H2O Molecules
Simulation Time (ns)

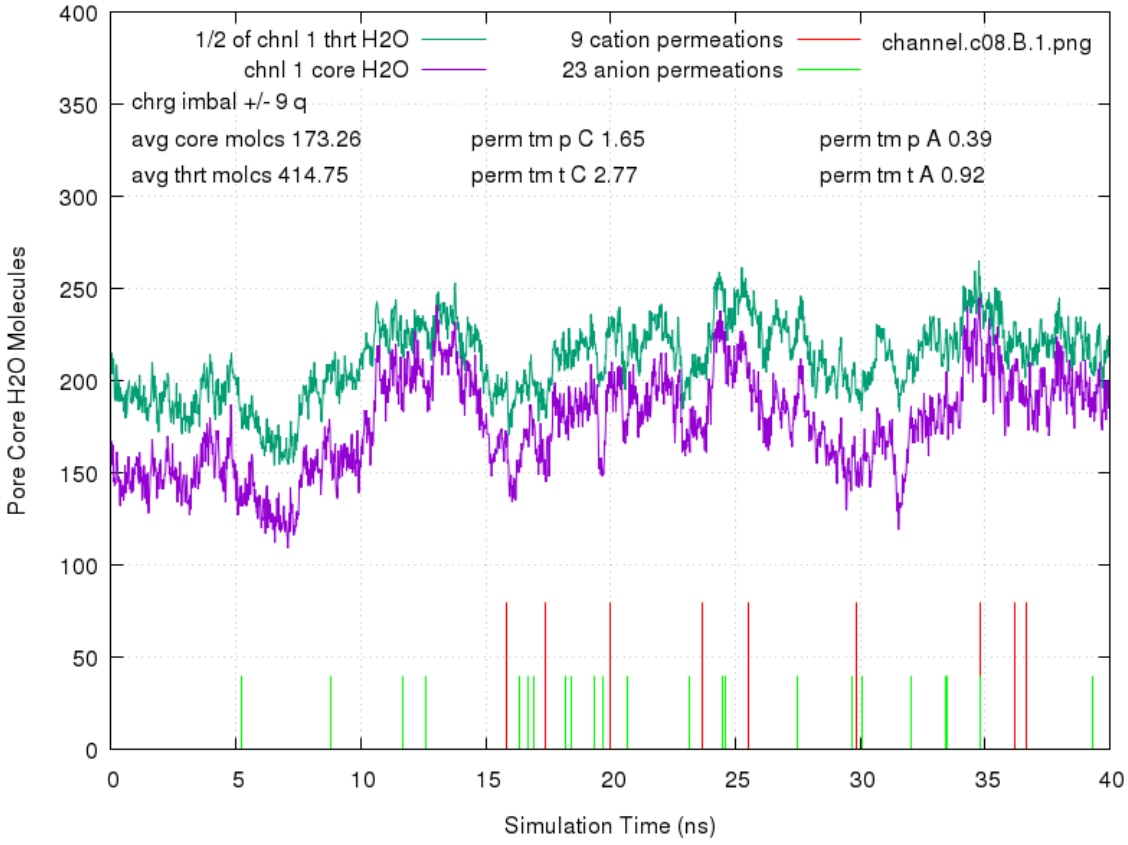

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
9 cation permeations
21 anion permeations
channel.c08.9.0.png
chrg imbal +/- 9 q
avg core molcs 148.26
avg thrt molcs 374.64
perm tm p C 1.49
perm tm t C 3.22
perm tm p A 0.42
perm tm t A 0.98
Pore Core H2O Molecules
Simulation Time (ns)

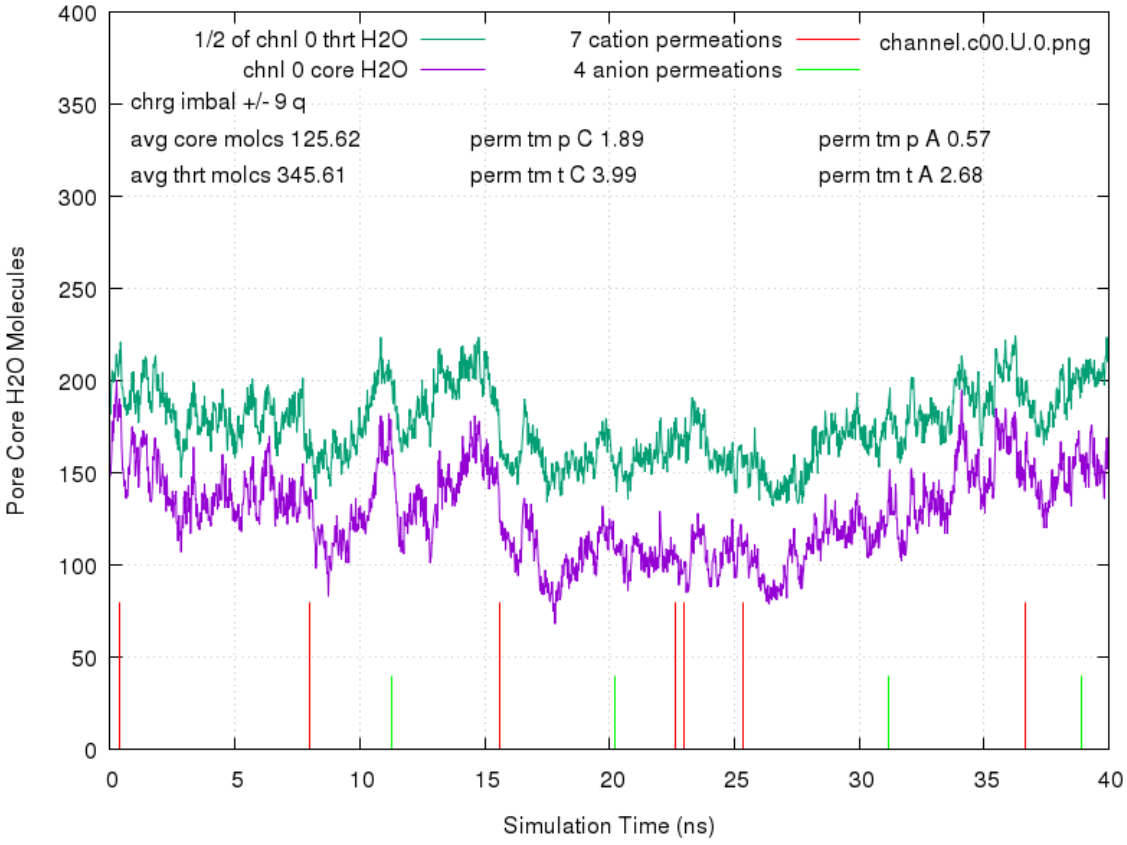

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
9 cation permeations
26 anion permeations
channel.c08.9.1.png
chrg imbal +/- 9 q
avg core molcs 169.61
avg thrt molcs 414.00
perm tm p C 0.79
perm tm t C 2.65
perm tm p A 0.31
perm tm t A 0.86
Pore Core H2O Molecules
Simulation Time (ns)

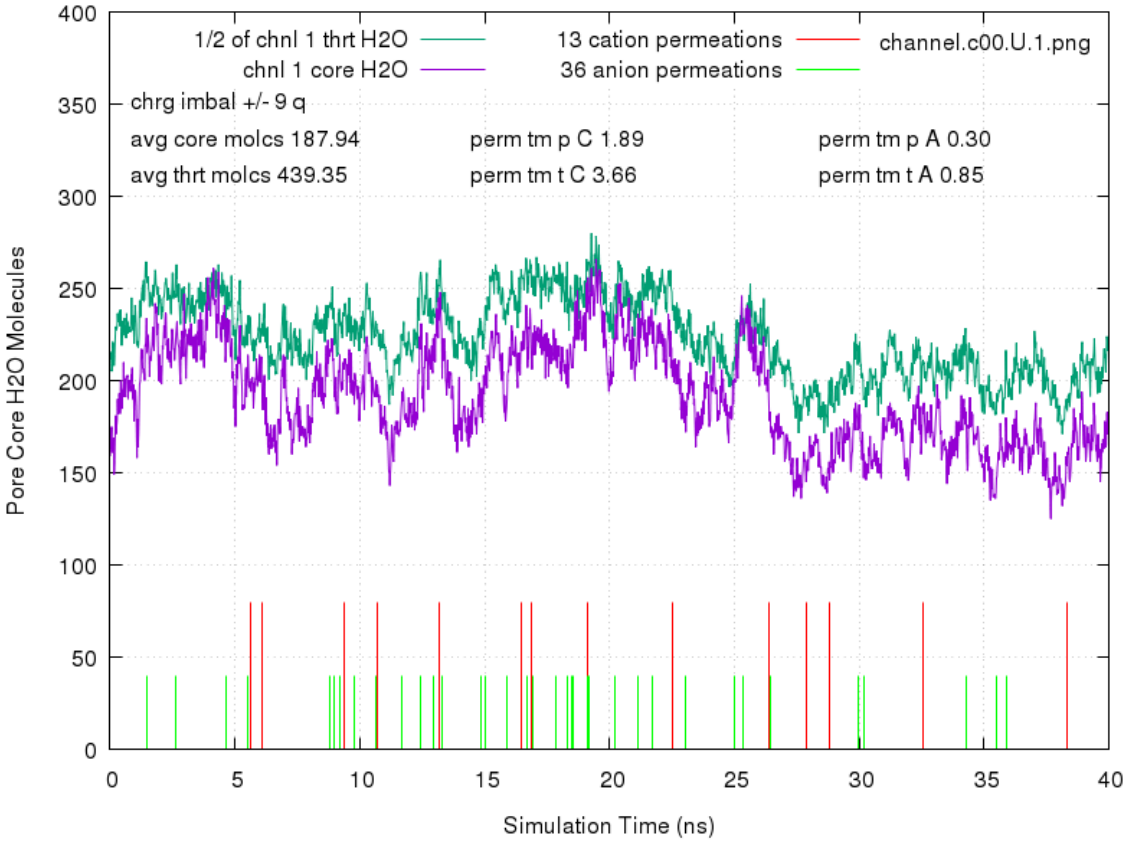

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
19 cation permeations
29 anion permeations
channel.c08.A.0.png
chrg imbal +/- 9 q
avg core molcs 191.70
avg thrt molcs 449.65
perm tm p C 1.25
perm tm t C 2.79
perm tm p A 0.30
perm tm t A 0.81
Pore Core H2O Molecules
Simulation Time (ns)

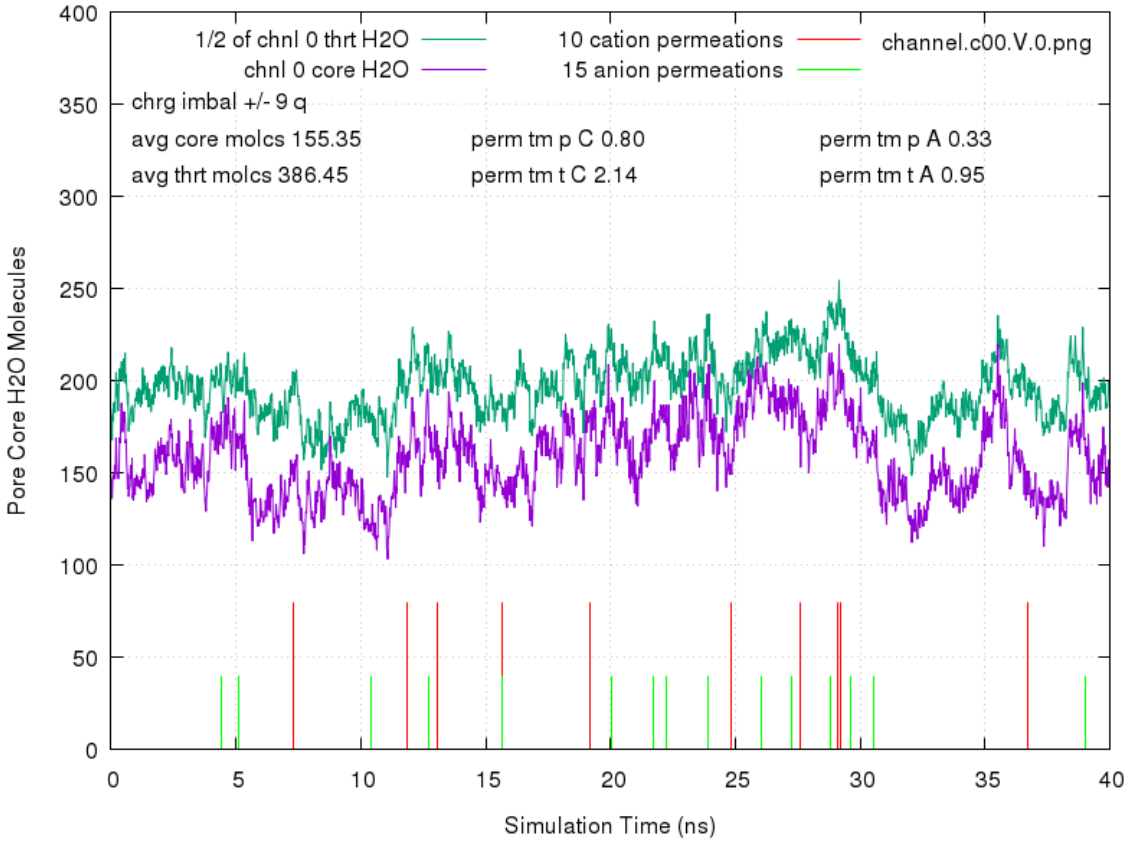

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
14 cation permeations
13 anion permeations
channel.c08.A.1.png
chrg imbal +/- 9 q
avg core molcs 137.36
avg thrt molcs 360.14
perm tm p C 1.50
perm tm t C 3.23
perm tm p A 0.43
perm tm t A 1.11
Pore Core H2O Molecules
Simulation Time (ns)

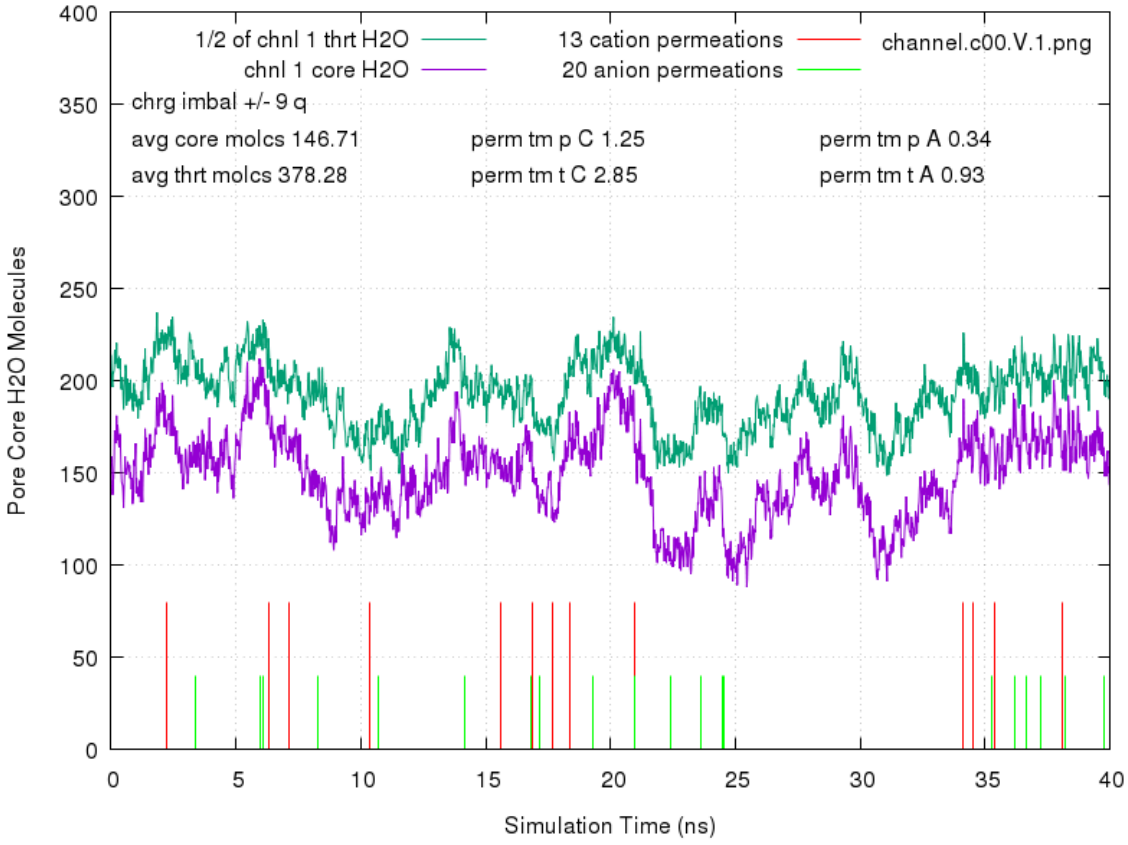

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
10 cation permeations
17 anion permeations
channel.c08.B.0.png
chrg imbal +/- 9 q
avg core molcs 149.35
avg thrt molcs 379.11
perm tm p C 1.85
perm tm t C 4.04
perm tm p A 0.45
perm tm t A 1.10
Pore Core H2O Molecules
Simulation Time (ns)

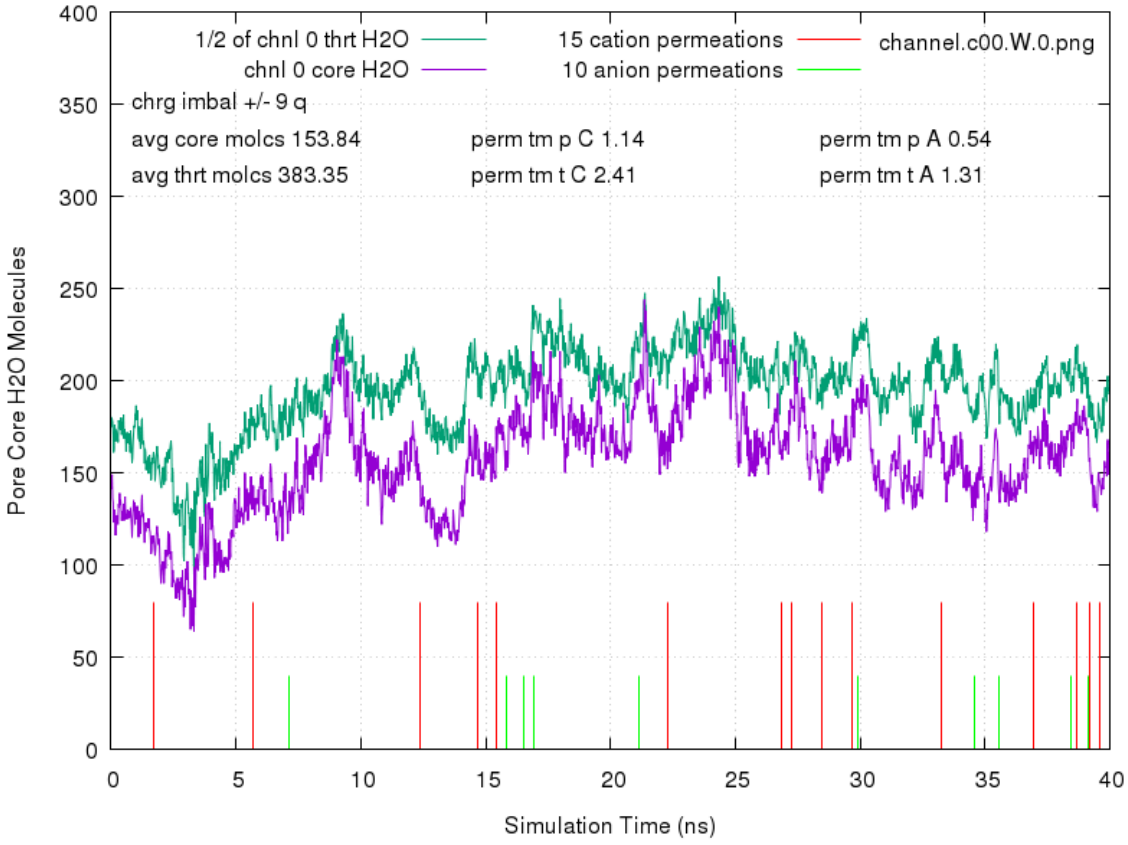

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
9 cation permeations
23 anion permeations
channel.c08.B.1.png
chrg imbal +/- 9 q
avg core molcs 173.26
avg thrt molcs 414.75
perm tm p C 1.65
perm tm t C 2.77
perm tm p A 0.39
perm tm t A 0.92
Pore Core H2O Molecules
Simulation Time (ns)

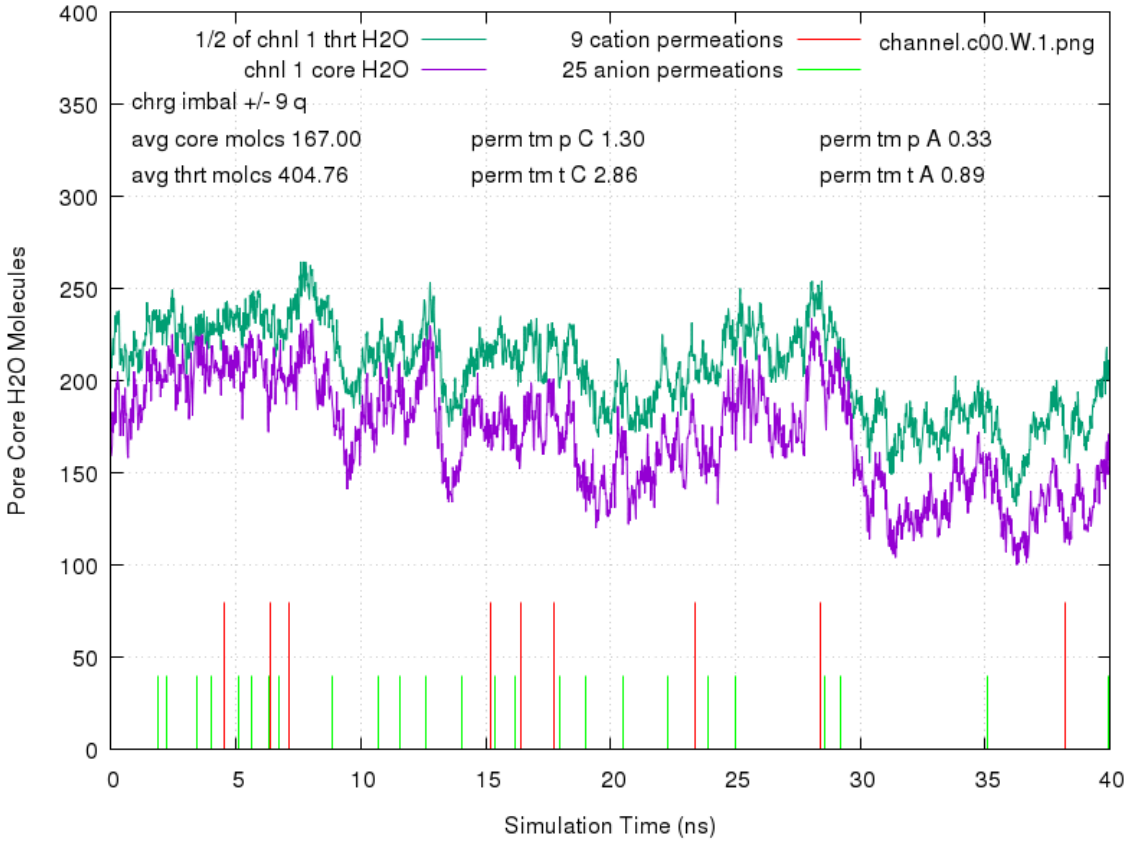

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
7 cation permeations
4 anion permeations
channel.c00.U.0.png
chrg imbal +/- 9 q
avg core molcs 125.62
avg thrt molcs 345.61
perm tm p C 1.89
perm tm t C 3.99
perm tm p A 0.57
perm tm t A 2.68
Pore Core H2O Molecules
Simulation Time (ns)

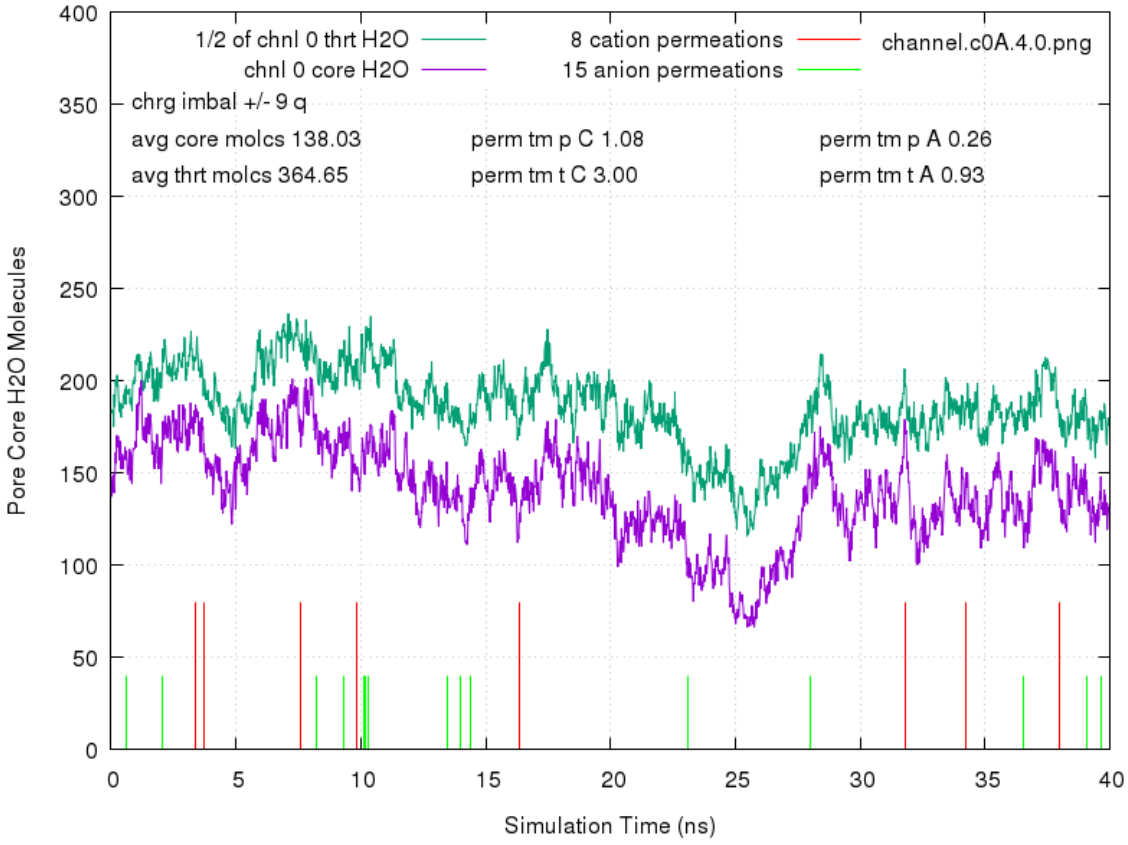

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
13 cation permeations
36 anion permeations
channel.c00.U.1.png
chrg imbal +/- 9 q
avg core molcs 187.94
avg thrt molcs 439.35
perm tm p C 1.89
perm tm t C 3.66
perm tm p A 0.30
perm tm t A 0.85
Pore Core H2O Molecules
Simulation Time (ns)

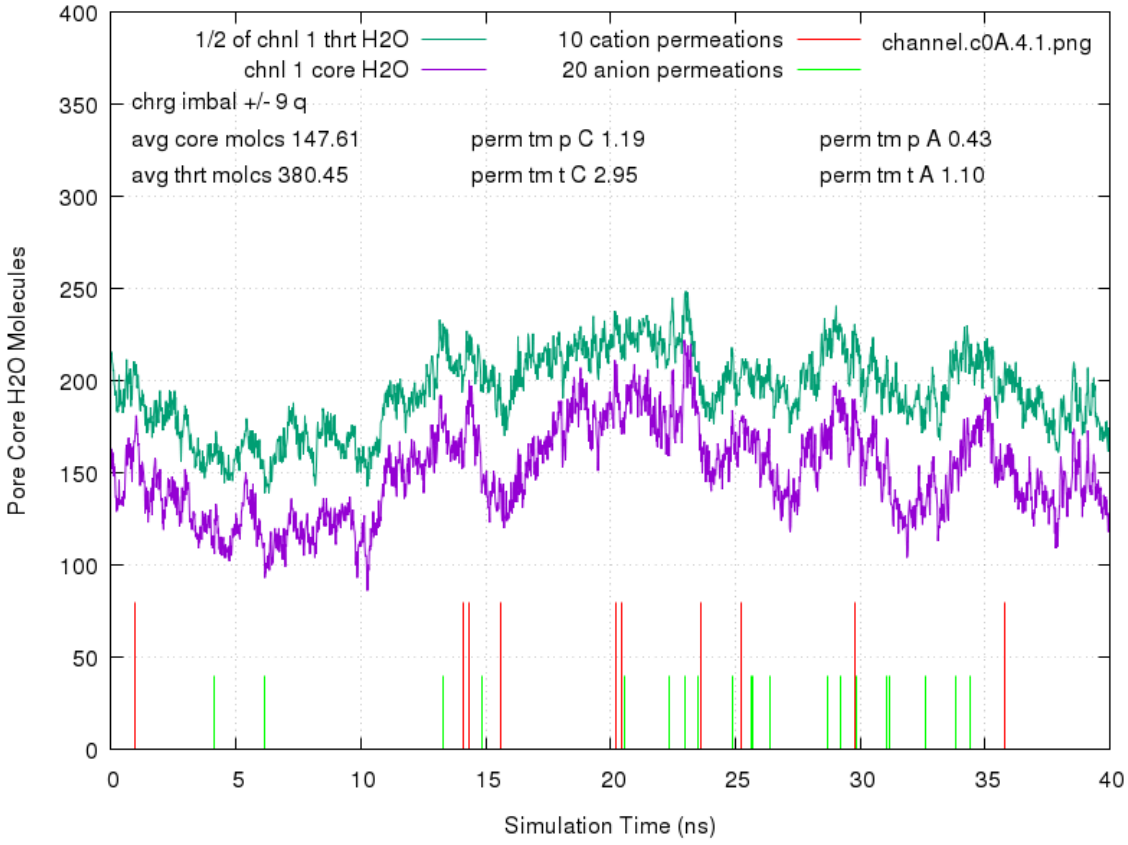

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
10 cation permeations
15 anion permeations
channel.c00.V.0.png
chrg imbal +/- 9 q
avg core molcs 155.35
avg thrt molcs 386.45
perm tm p C 0.80
perm tm t C 2.14
perm tm p A 0.33
perm tm t A 0.95
Pore Core H2O Molecules
Simulation Time (ns)

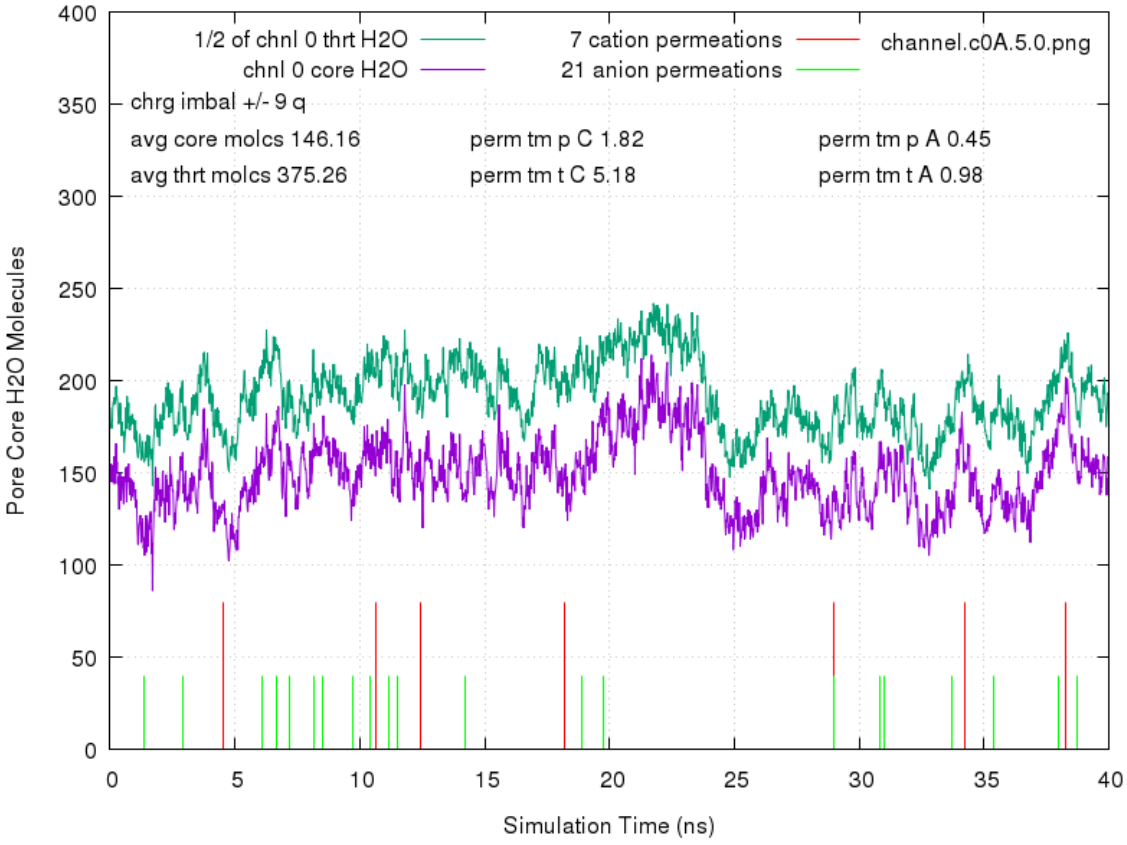

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
13 cation permeations
20 anion permeations
channel.c00.V.1.png
chrg imbal +/- 9 q
avg core molcs 146.71
avg thrt molcs 378.28
perm tm p C 1.25
perm tm t C 2.85
perm tm p A 0.34
perm tm t A 0.93
Pore Core H2O Molecules
Simulation Time (ns)

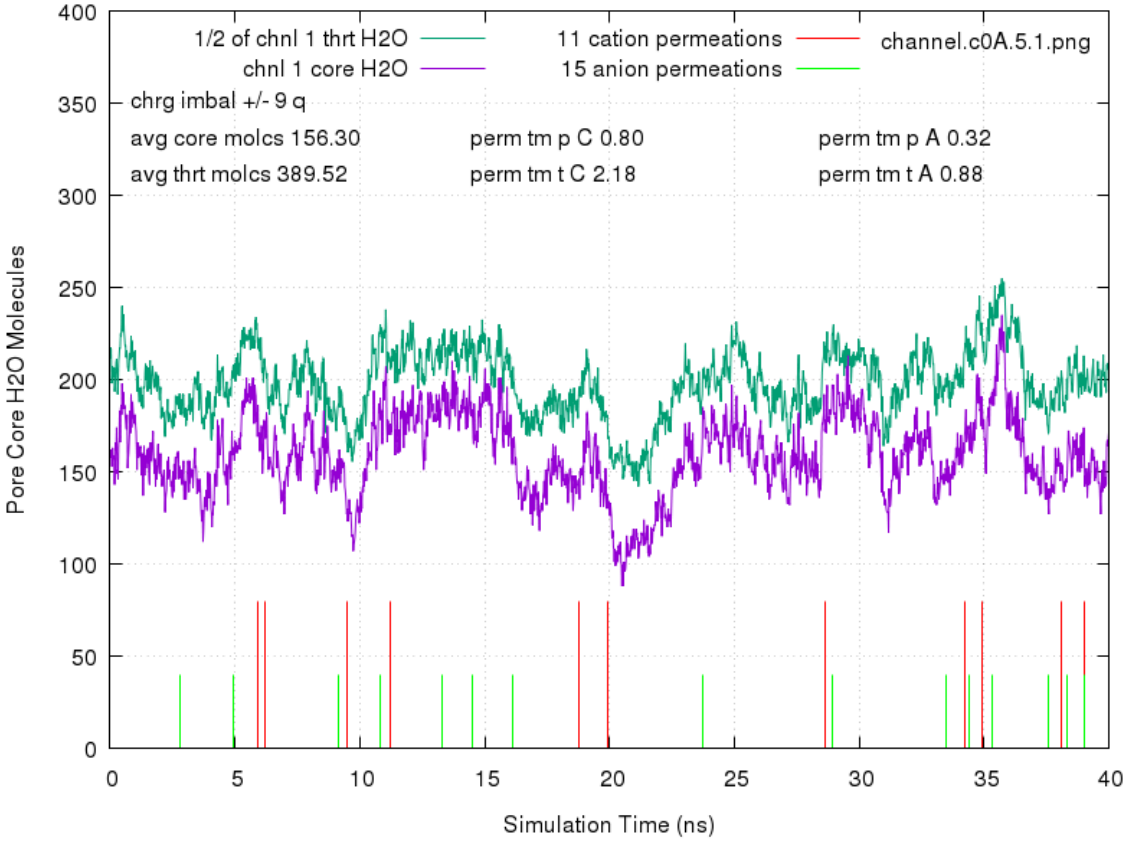

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
15 cation permeations
10 anion permeations
channel.c00.W.0.png
chrg imbal +/- 9 q
avg core molcs 153.84
avg thrt molcs 383.35
perm tm p C 1.14
perm tm t C 2.41
perm tm p A 0.54
perm tm t A 1.31
Pore Core H2O Molecules
Simulation Time (ns)

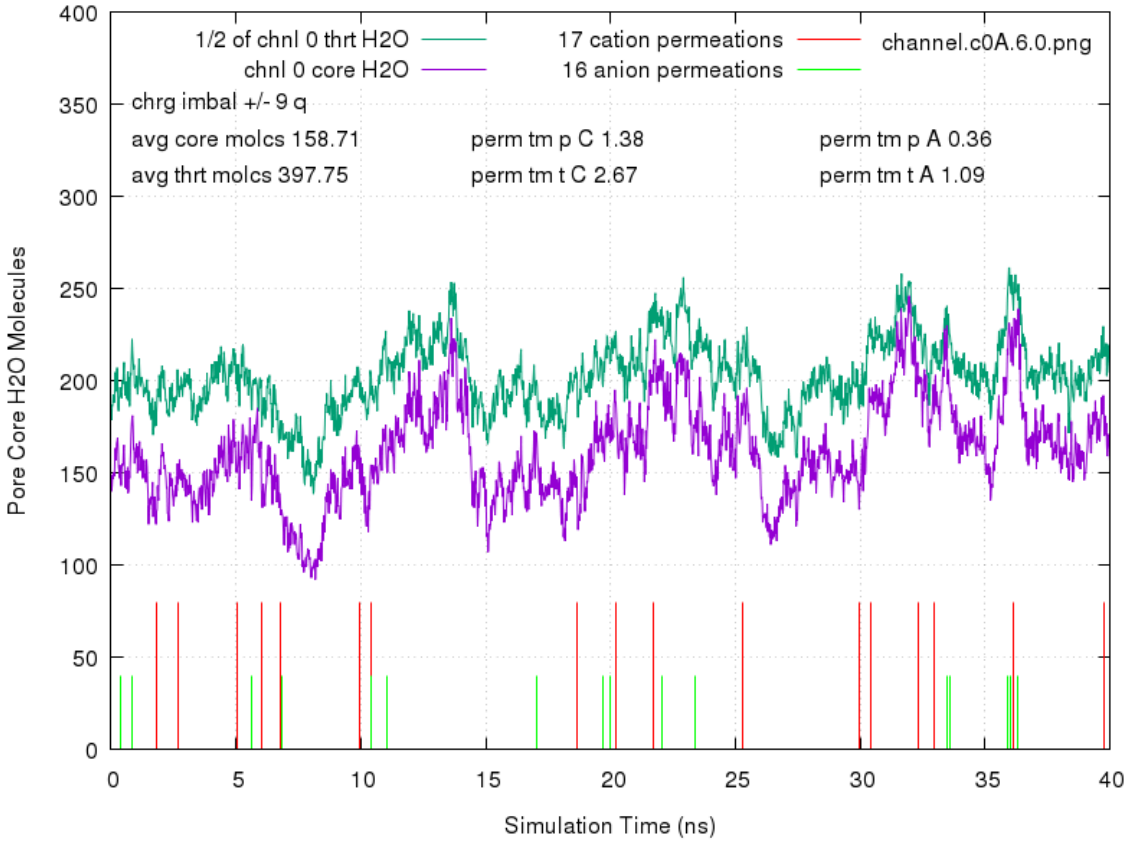

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
9 cation permeations
25 anion permeations
channel.c00.W.1.png
chrg imbal +/- 9 q
avg core molcs 167.00
avg thrt molcs 404.76
perm tm p C 1.30
perm tm t C 2.86
perm tm p A 0.33
perm tm t A 0.89
Pore Core H2O Molecules
Simulation Time (ns)

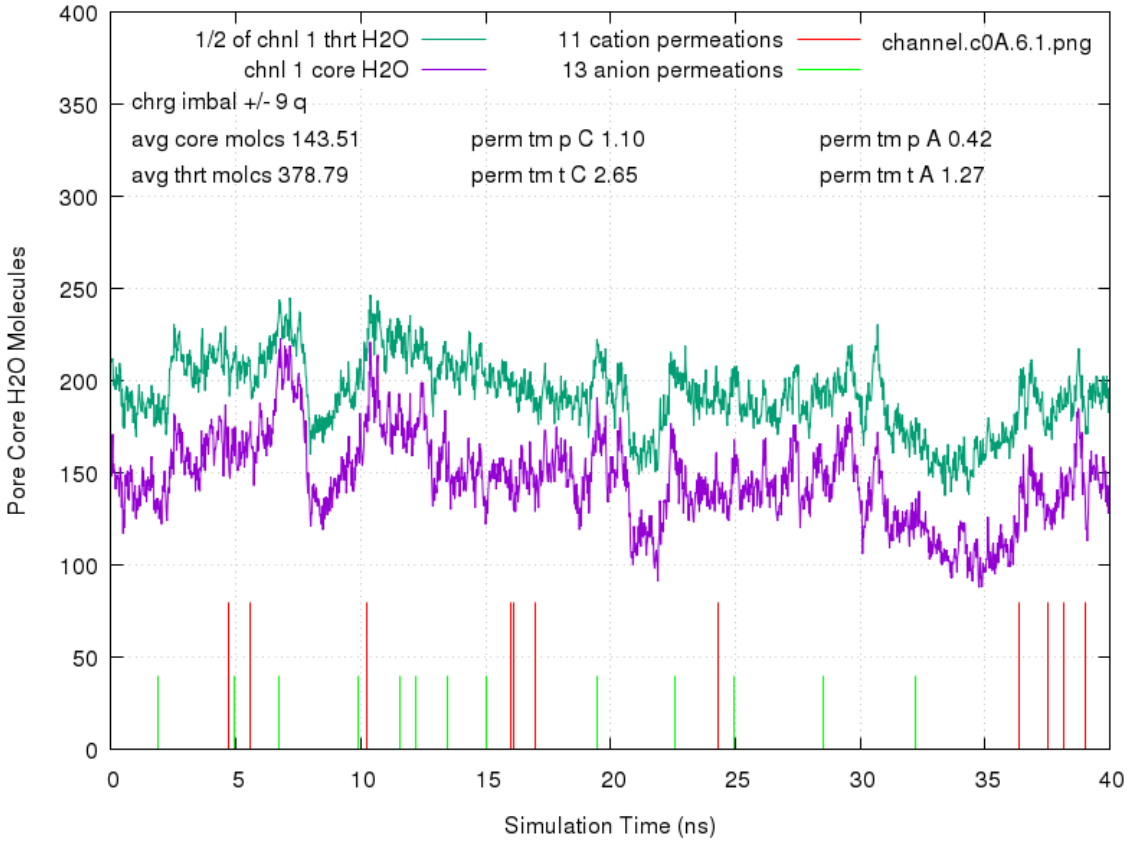

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
15 anion permeations
channel.c0A.4.0.png
chrg imbal +/- 9 q
avg core molcs 138.03
avg thrt molcs 364.65
perm tm p C 1.08
perm tm t C 3.00
perm tm p A 0.26
perm tm t A 0.93
Pore Core H2O Molecules
Simulation Time (ns)

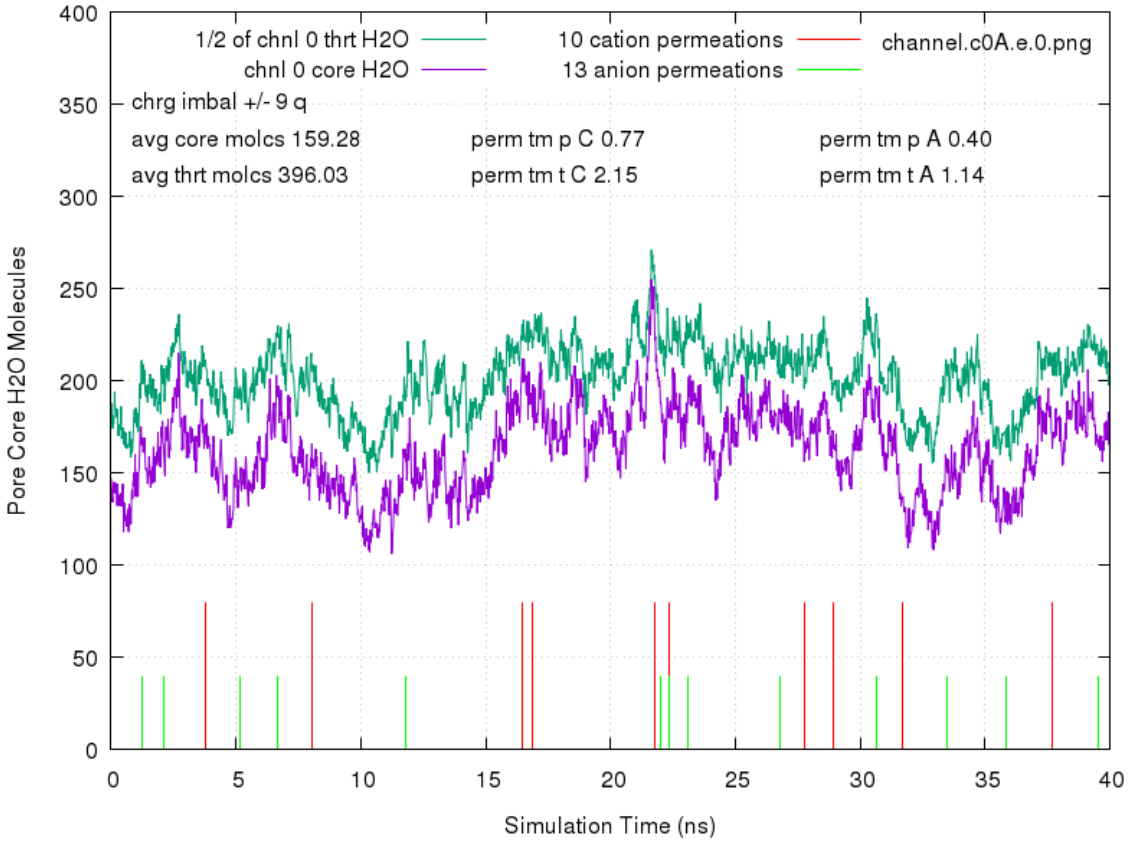

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
10 cation permeations
20 anion permeations
channel.c0A.4.1.png
chrg imbal +/- 9 q
avg core molcs 147.61
avg thrt molcs 380.45
perm tm p C 1.19
perm tm t C 2.95
perm tm p A 0.43
perm tm t A 1.10
Pore Core H2O Molecules
Simulation Time (ns)

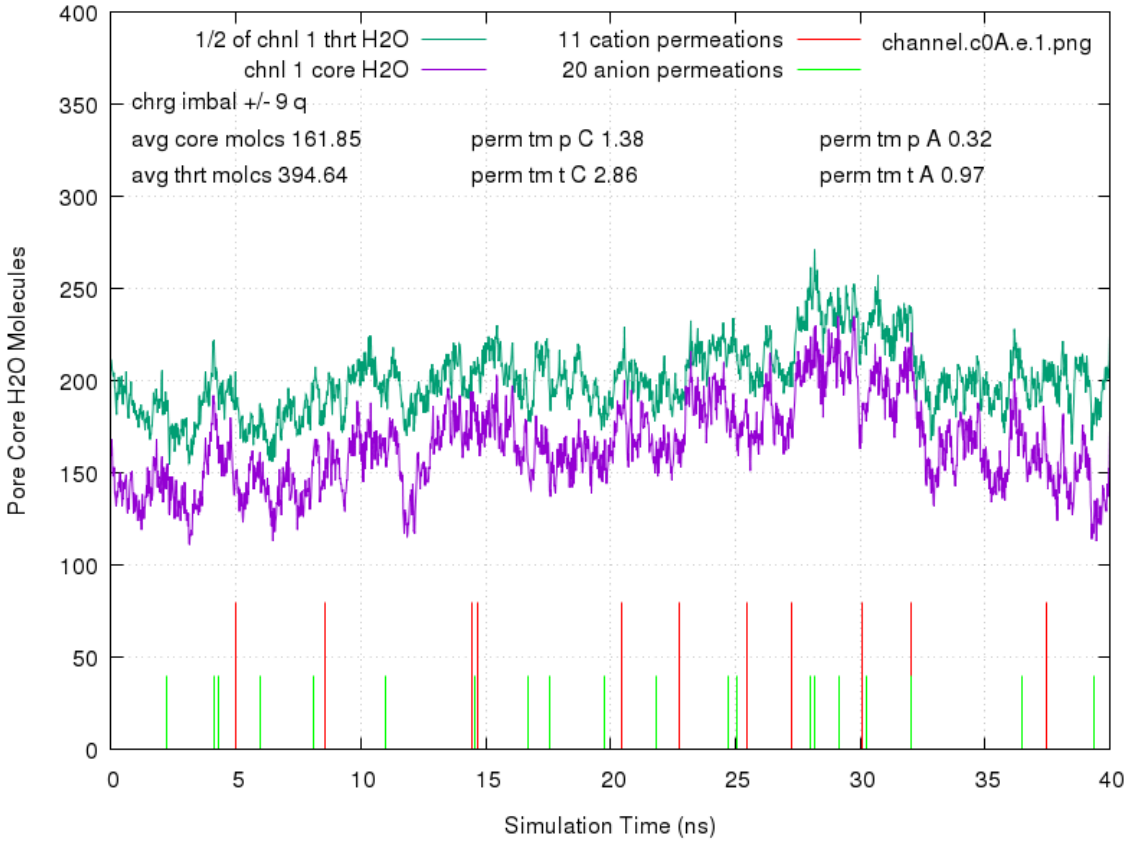

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
7 cation permeations
21 anion permeations
channel.c0A.5.0.png
chrg imbal +/- 9 q
avg core molcs 146.16
avg thrt molcs 375.26
perm tm p C 1.82
perm tm t C 5.18
perm tm p A 0.45
perm tm t A 0.98
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

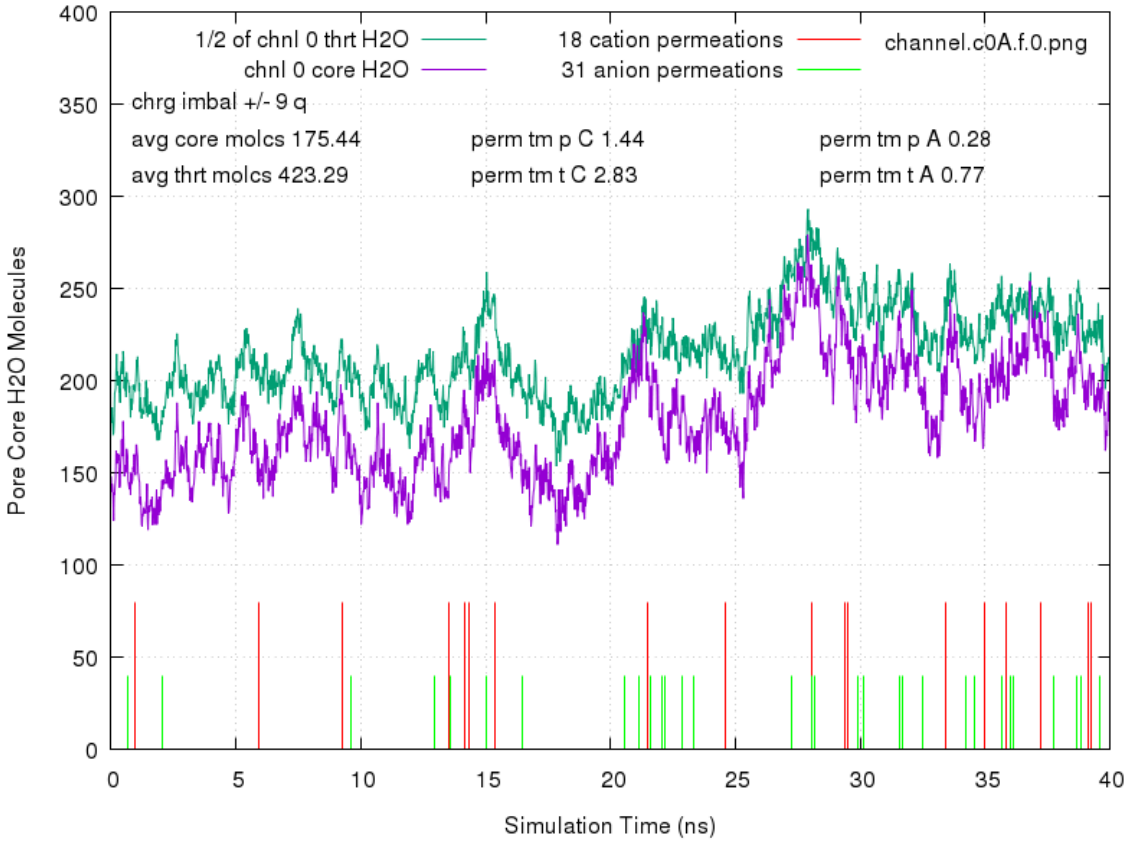

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
11 cation permeations
15 anion permeations
channel.c0A.5.1.png
chrg imbal +/- 9 q
avg core molcs 156.30
avg thrt molcs 389.52
perm tm p C 0.80
perm tm t C 2.18
perm tm p A 0.32
perm tm t A 0.88
Pore Core H2O Molecules
Simulation Time (ns)
0
50
100
150
200
250
300
350
400
0
5
10
15
20
25
30
35
40

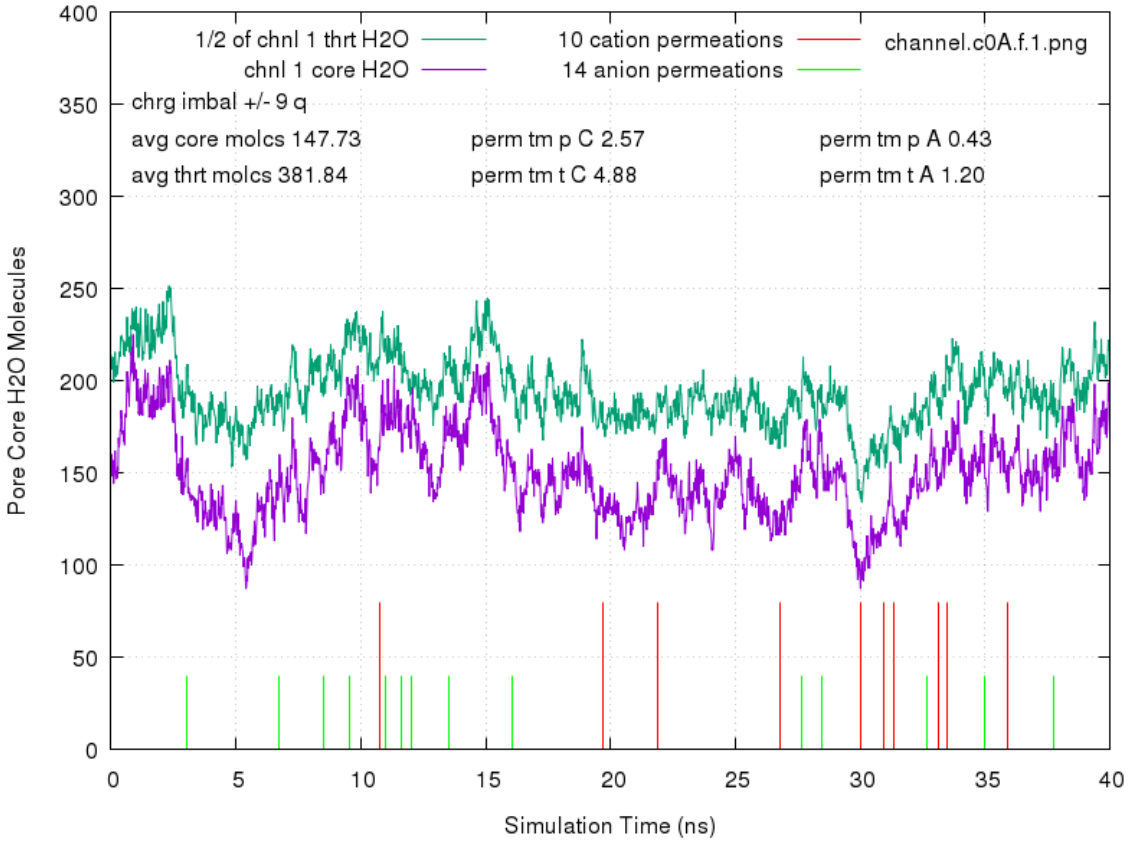

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
17 cation permeations
16 anion permeations
channel.c0A.6.0.png
chrg imbal +/- 9 q
avg core molcs 158.71
avg thrt molcs 397.75
perm tm p C 1.38
perm tm t C 2.67
perm tm p A 0.36
perm tm t A 1.09
Pore Core H2O Molecules
Simulation Time (ns)

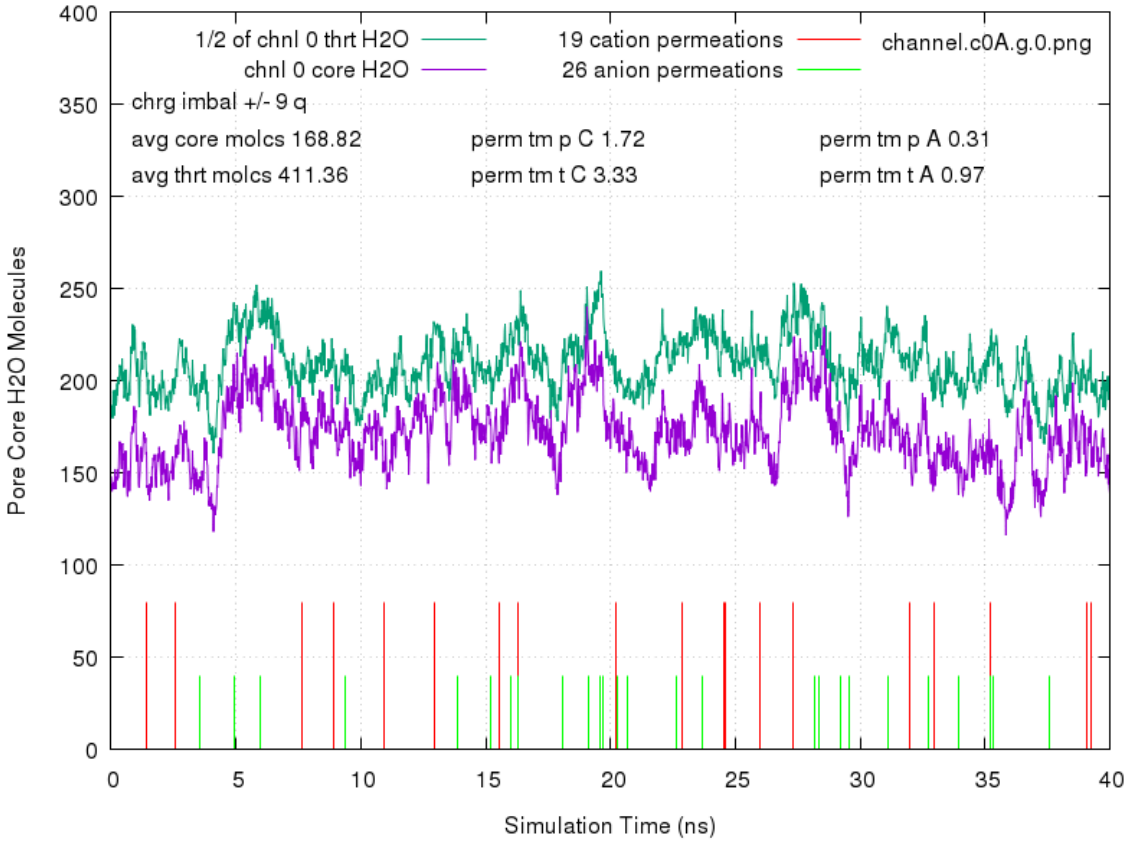

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
11 cation permeations
13 anion permeations
channel.c0A.6.1.png
chrg imbal +/- 9 q
avg core molcs 143.51
avg thrt molcs 378.79
perm tm p C 1.10
perm tm t C 2.65
perm tm p A 0.42
perm tm t A 1.27
Pore Core H2O Molecules
Simulation Time (ns)

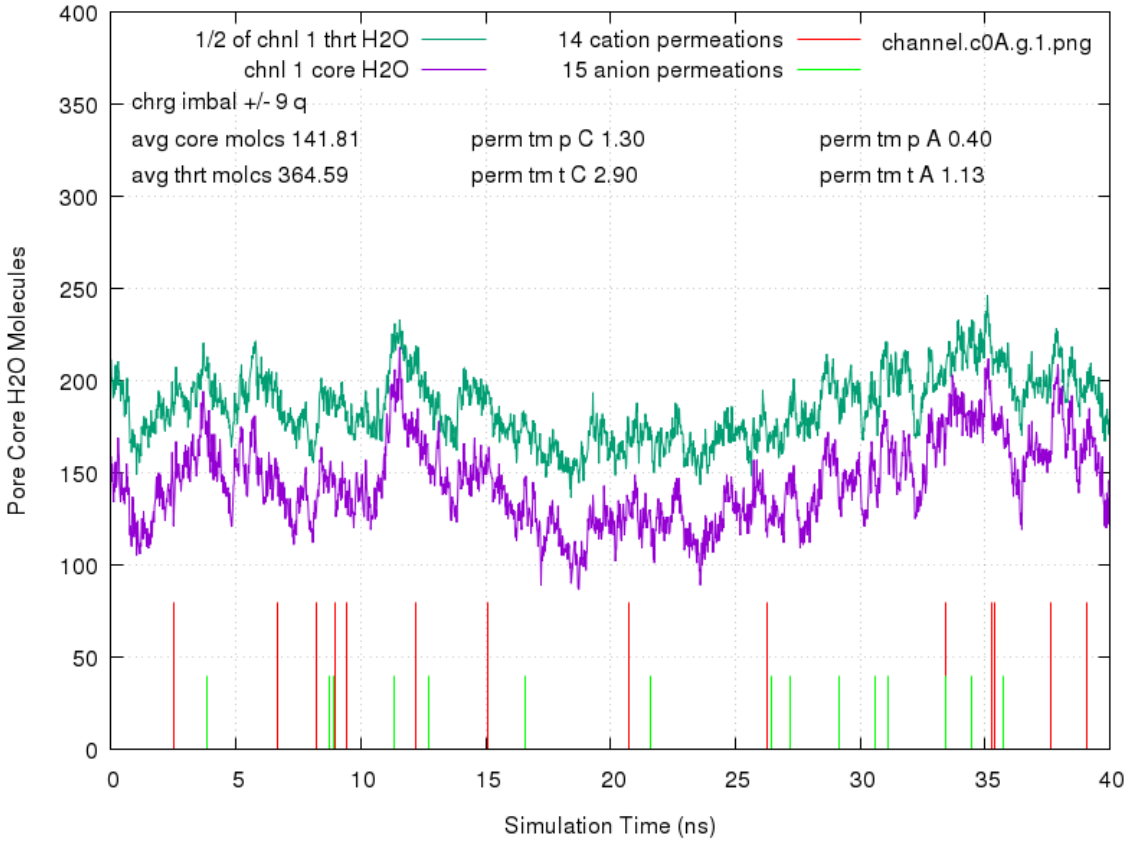

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
10 cation permeations
13 anion permeations
channel.c0A.e.0.png
chrg imbal +/- 9 q
avg core molcs 159.28
avg thrt molcs 396.03
perm tm p C 0.77
perm tm t C 2.15
perm tm p A 0.40
perm tm t A 1.14
Pore Core H2O Molecules
Simulation Time (ns)

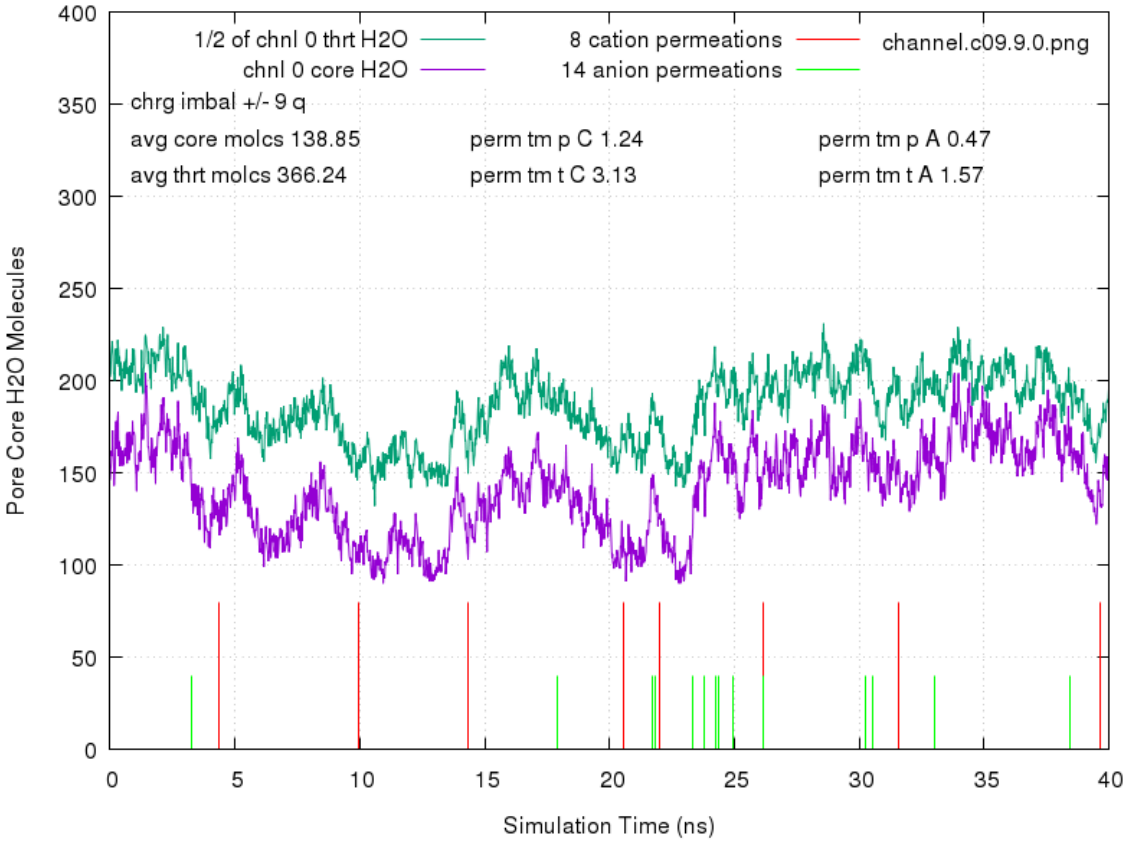

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
11 cation permeations
20 anion permeations
channel.c0A.e.1.png
chrg imbal +/- 9 q
avg core molcs 161.85
avg thrt molcs 394.64
perm tm p C 1.38
perm tm t C 2.86
perm tm p A 0.32
perm tm t A 0.97
Pore Core H2O Molecules
Simulation Time (ns)

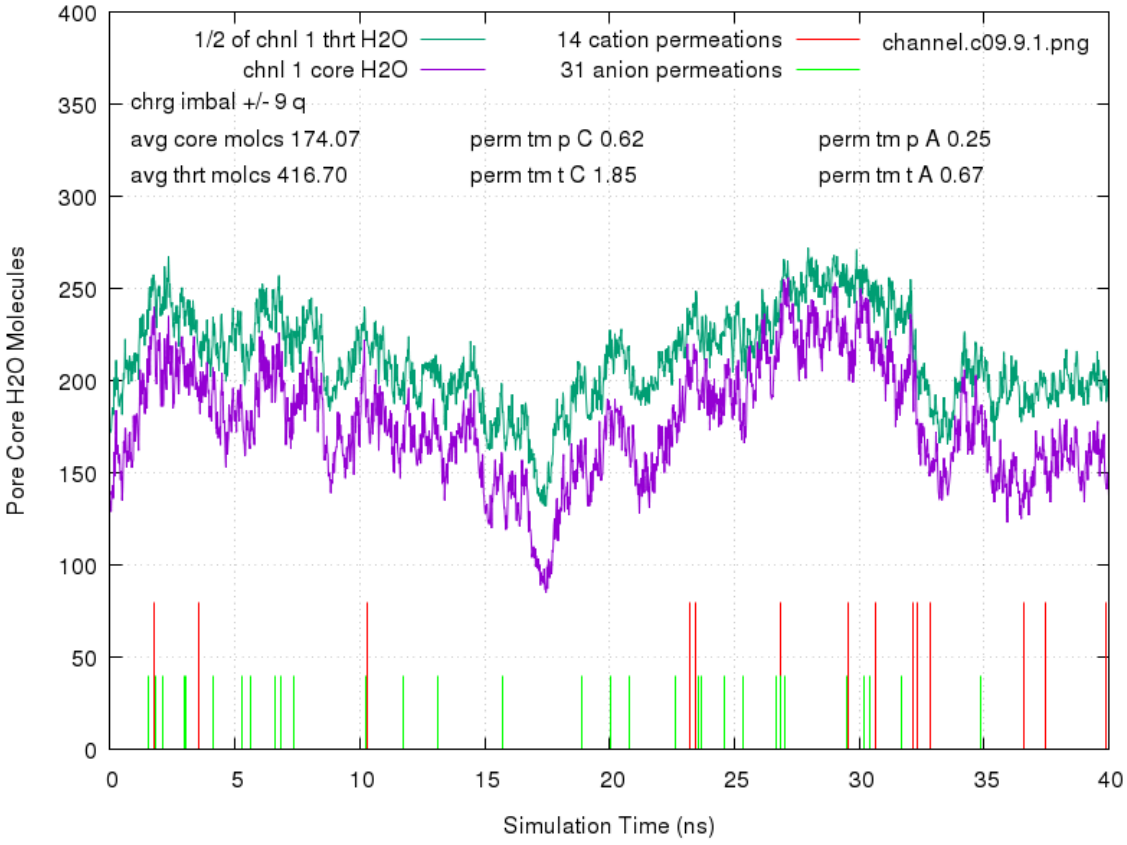

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
18 cation permeations
31 anion permeations
channel.c0A.f.0.png
chrg imbal +/- 9 q
avg core molcs 175.44
avg thrt molcs 423.29
perm tm p C 1.44
perm tm t C 2.83
perm tm p A 0.28
perm tm t A 0.77
Pore Core H2O Molecules
Simulation Time (ns)

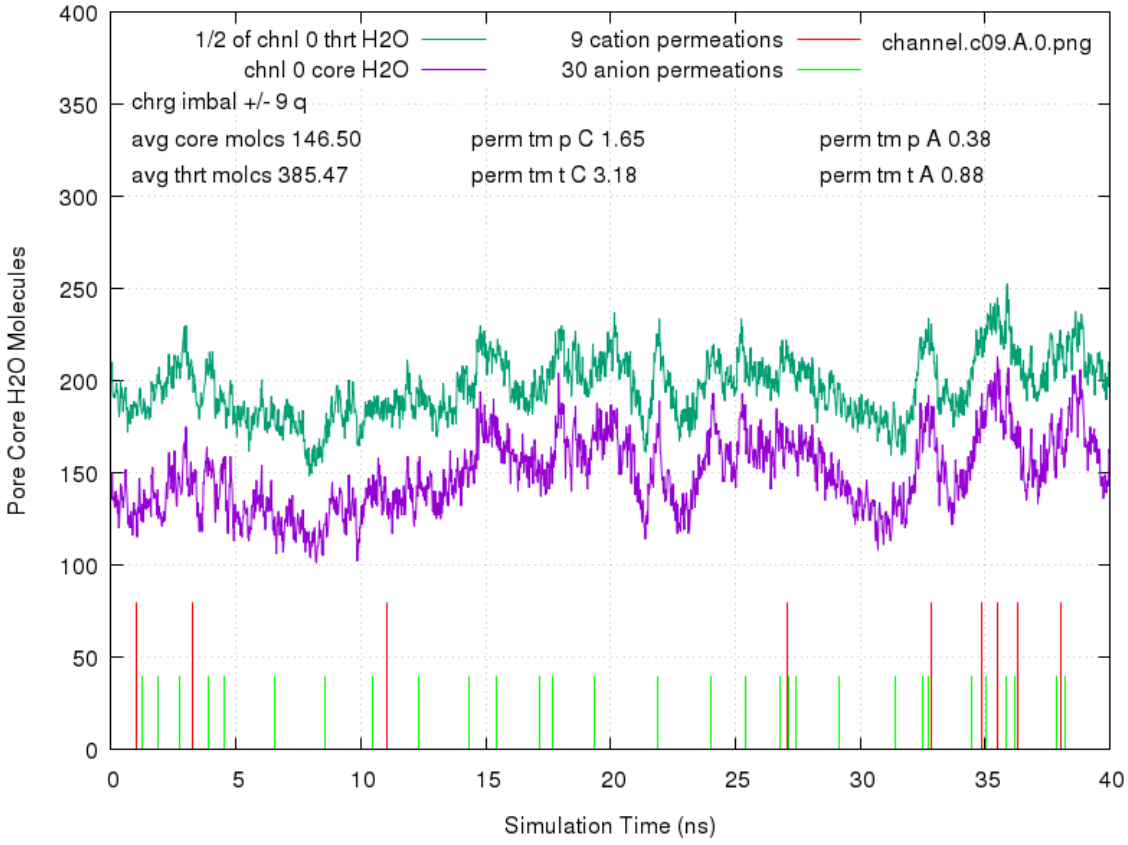

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
10 cation permeations
14 anion permeations
channel.c0A.f.1.png
chrg imbal +/- 9 q
avg core molcs 147.73
avg thrt molcs 381.84
perm tm p C 2.57
perm tm t C 4.88
perm tm p A 0.43
perm tm t A 1.20
Pore Core H2O Molecules
Simulation Time (ns)

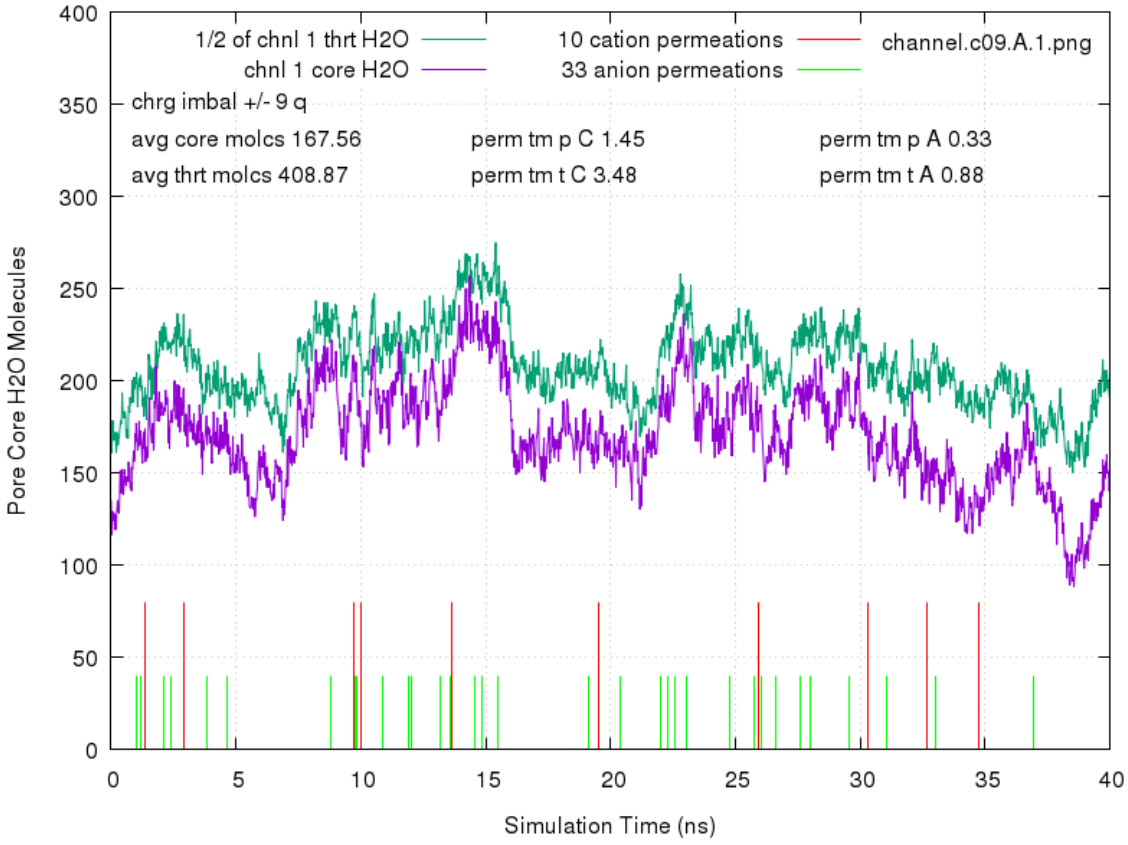

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
19 cation permeations
26 anion permeations
channel.c0A.g.0.png
chrg imbal +/- 9 q
avg core molcs 168.82
avg thrt molcs 411.36
perm tm p C 1.72
perm tm t C 3.33
perm tm p A 0.31
perm tm t A 0.97
Pore Core H2O Molecules
Simulation Time (ns)

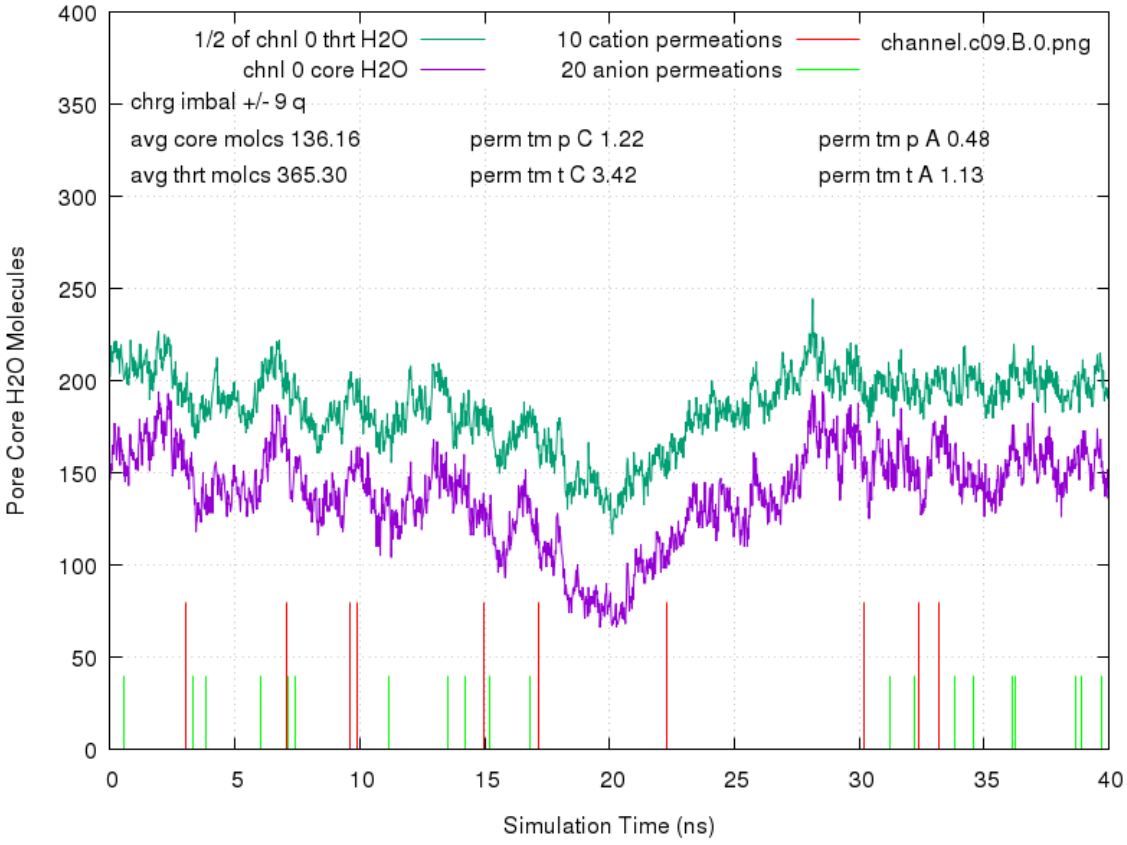

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
14 cation permeations
15 anion permeations
channel.c0A.g.1.png
chrg imbal +/- 9 q
avg core molcs 141.81
avg thrt molcs 364.59
perm tm p C 1.30
perm tm t C 2.90
perm tm p A 0.40
perm tm t A 1.13
Pore Core H2O Molecules
Simulation Time (ns)

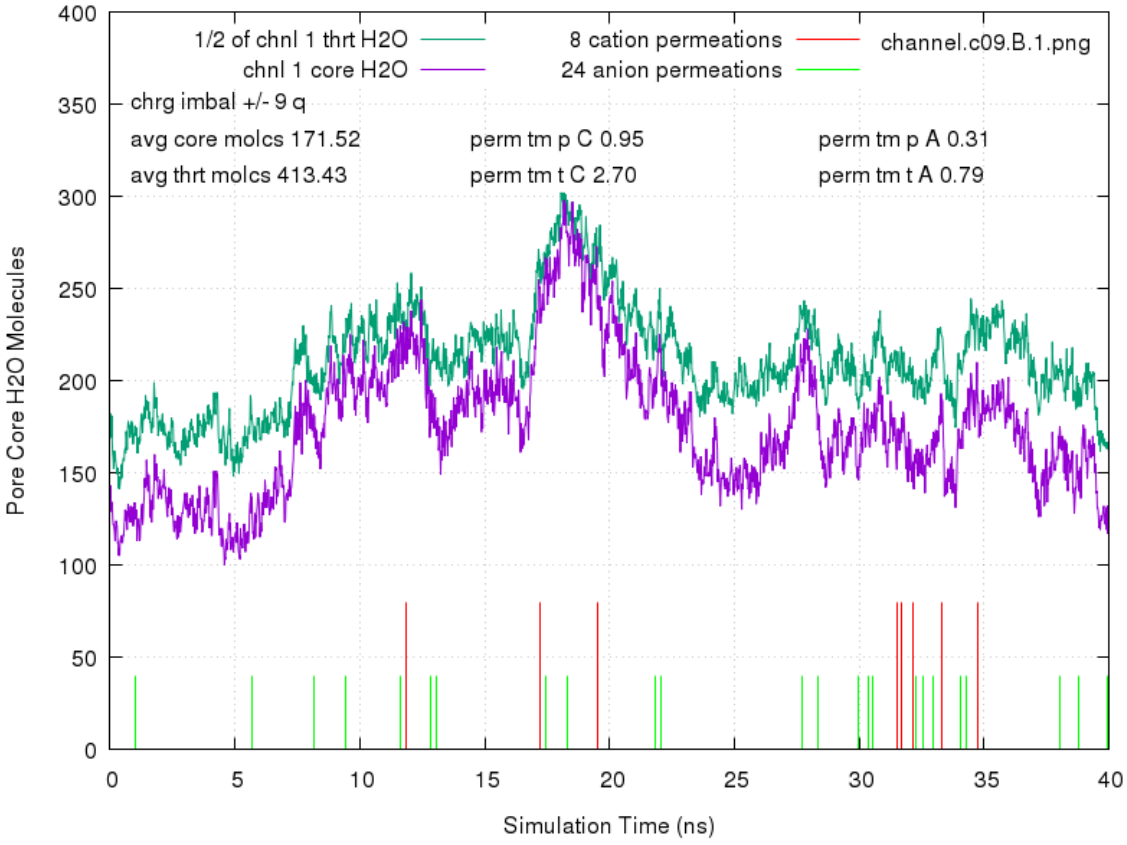

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
14 anion permeations
channel.c09.9.0.png
chrg imbal +/- 9 q
avg core molcs 138.85
avg thrt molcs 366.24
perm tm p C 1.24
perm tm t C 3.13
perm tm p A 0.47
perm tm t A 1.57
Pore Core H2O Molecules
Simulation Time (ns)

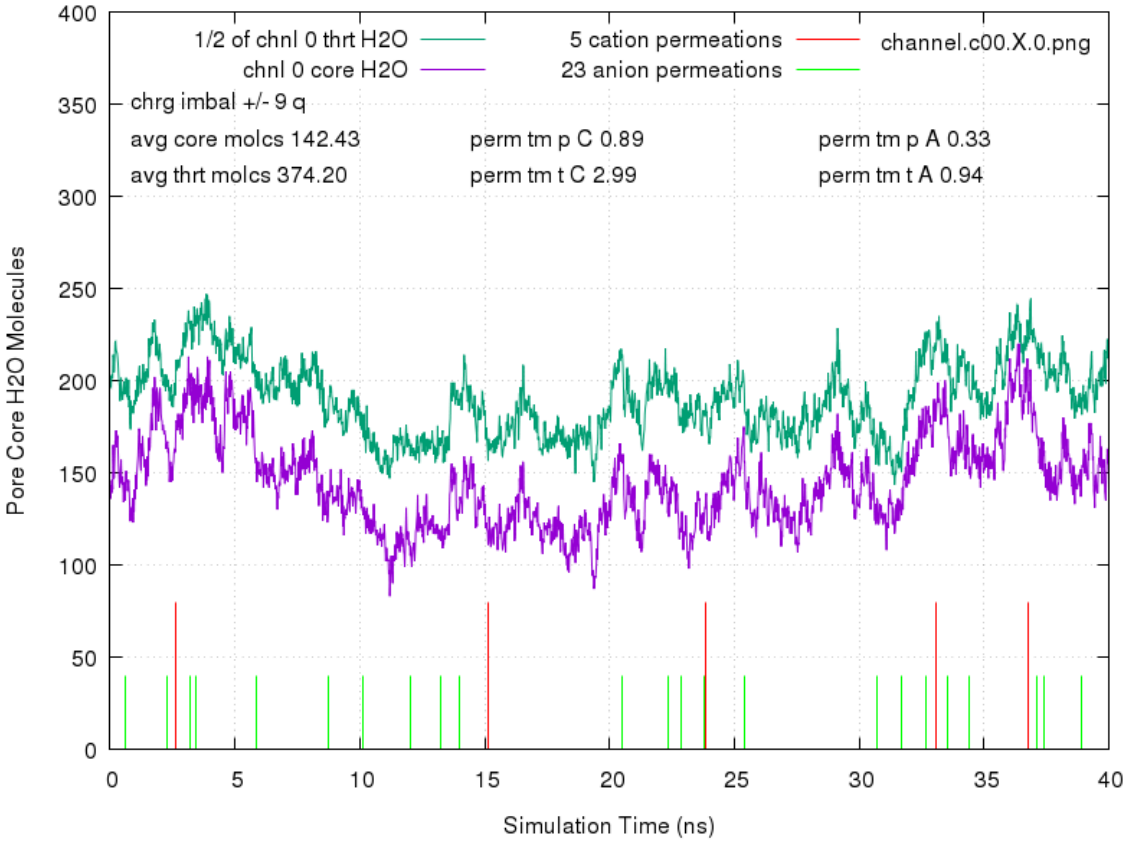

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
14 cation permeations
31 anion permeations
channel.c09.9.1.png
chrg imbal +/- 9 q
avg core molcs 174.07
avg thrt molcs 416.70
perm tm p C 0.62
perm tm t C 1.85
perm tm p A 0.25
perm tm t A 0.67
Pore Core H2O Molecules
Simulation Time (ns)

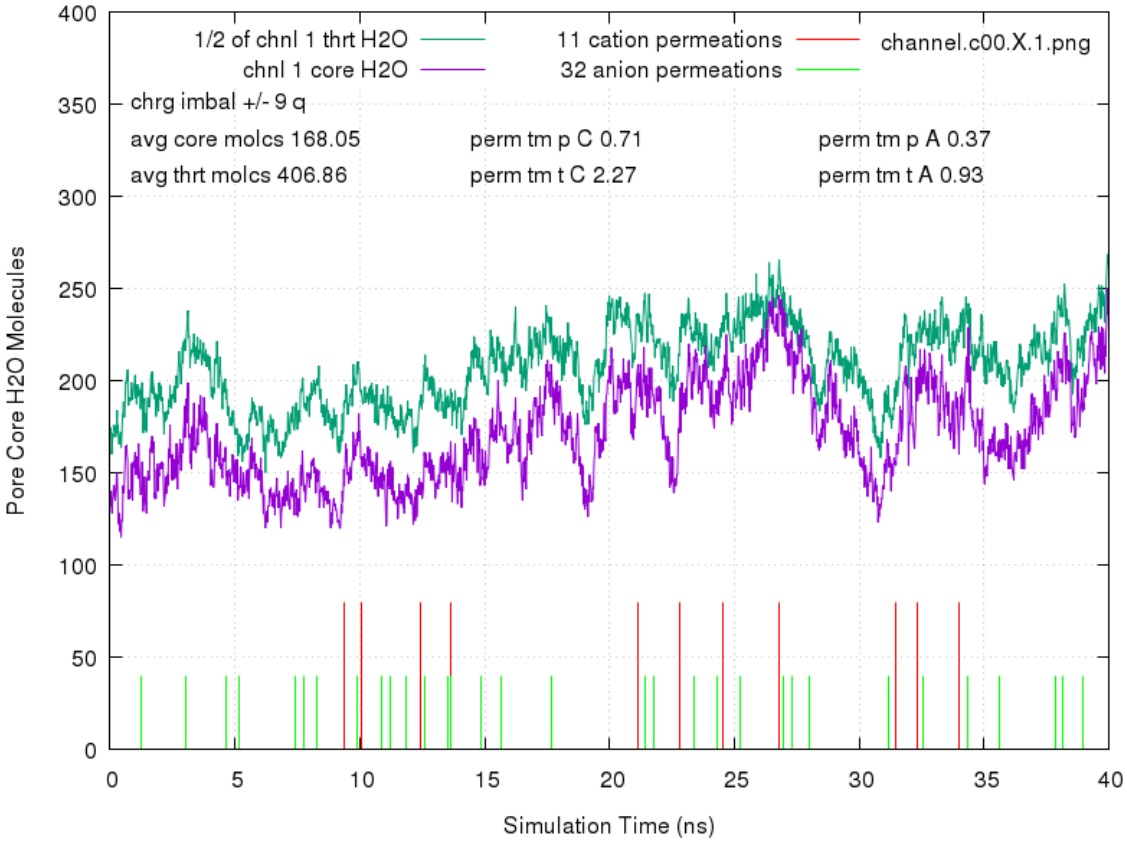

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
9 cation permeations
30 anion permeations
channel.c09.A.0.png
chrg imbal +/- 9 q
avg core molcs 146.50
avg thrt molcs 385.47
perm tm p C 1.65
perm tm t C 3.18
perm tm p A 0.38
perm tm t A 0.88
Pore Core H2O Molecules
Simulation Time (ns)

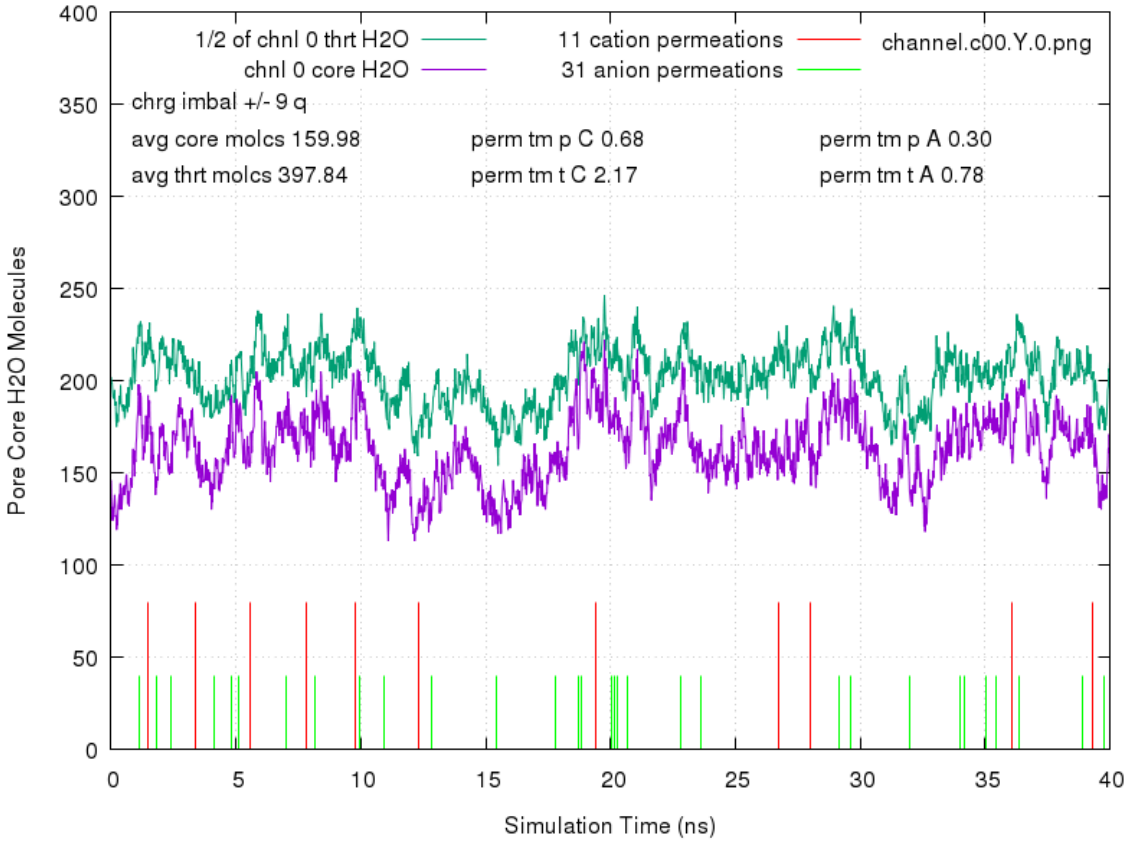

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
10 cation permeations
33 anion permeations
channel.c09.A.1.png
chrg imbal +/- 9 q
avg core molcs 167.56
avg thrt molcs 408.87
perm tm p C 1.45
perm tm t C 3.48
perm tm p A 0.33
perm tm t A 0.88
Pore Core H2O Molecules
Simulation Time (ns)

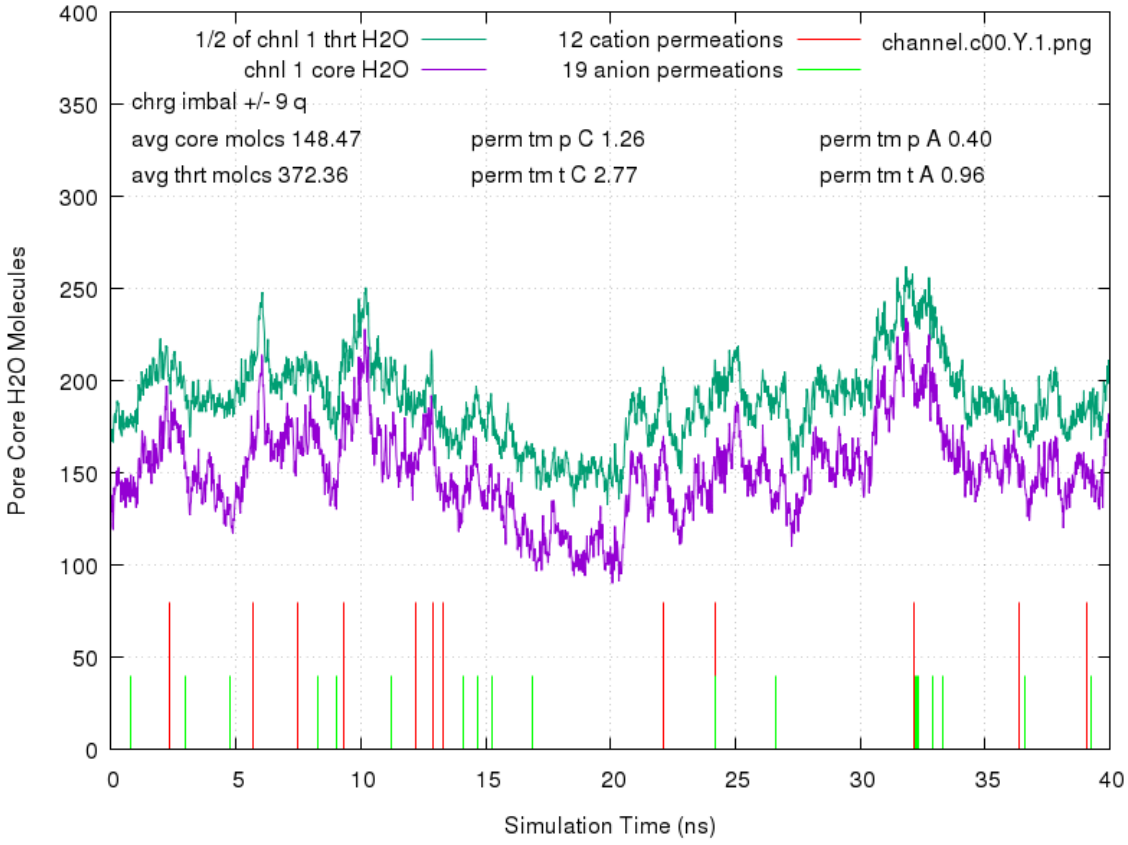

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
10 cation permeations
20 anion permeations
channel.c09.B.0.png
chrg imbal +/- 9 q
avg core molcs 136.16
avg thrt molcs 365.30
perm tm p C 1.22
perm tm t C 3.42
perm tm p A 0.48
perm tm t A 1.13
Pore Core H2O Molecules
Simulation Time (ns)

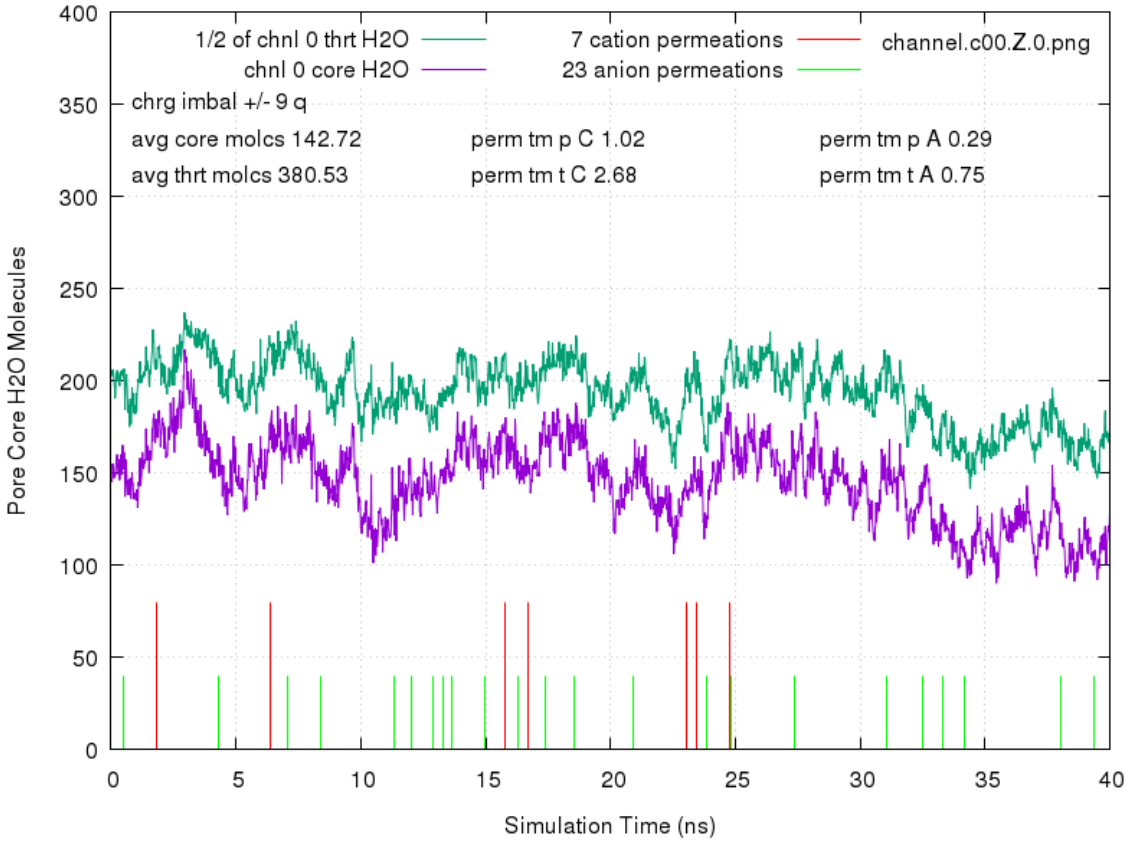

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
8 cation permeations
24 anion permeations
channel.c09.B.1.png
chrg imbal +/- 9 q
avg core molcs 171.52
avg thrt molcs 413.43
perm tm p C 0.95
perm tm t C 2.70
perm tm p A 0.31
perm tm t A 0.79
Pore Core H2O Molecules
Simulation Time (ns)

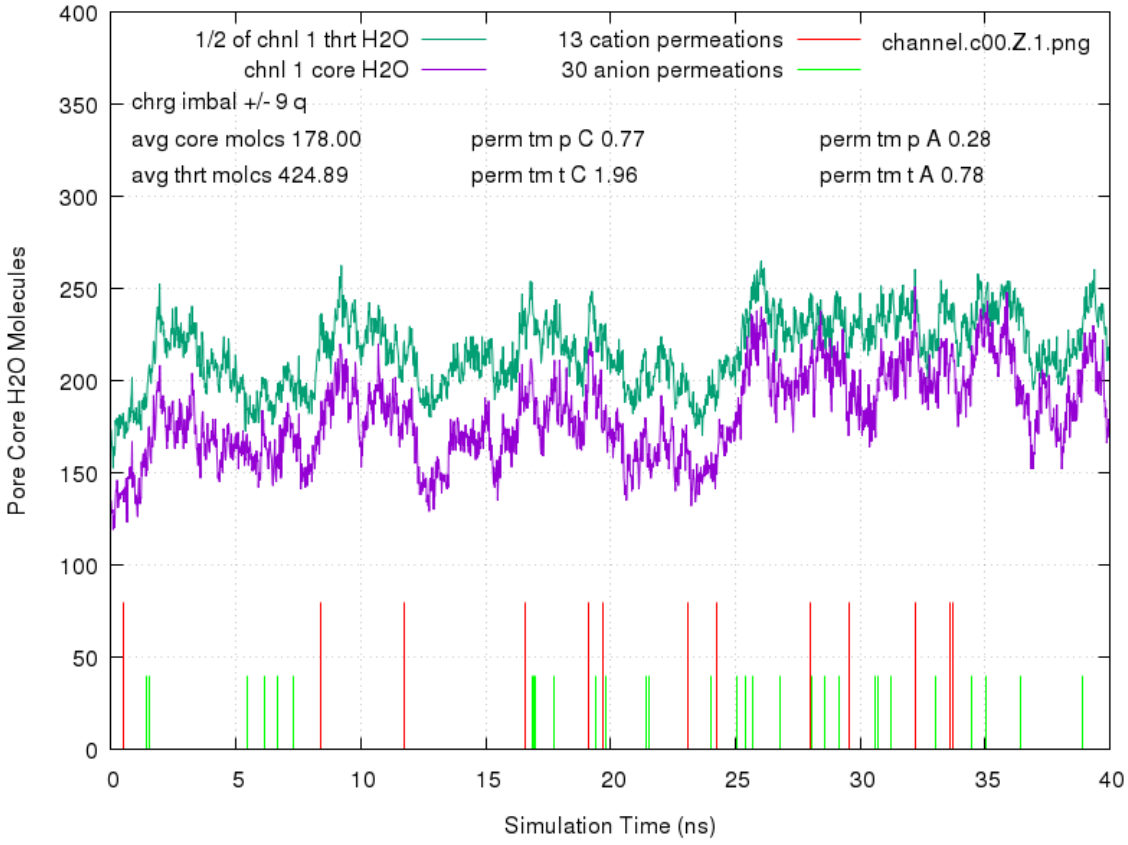

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
5 cation permeations
23 anion permeations
channel.c00.X.0.png
chrg imbal +/- 9 q
avg core molcs 142.43
avg thrt molcs 374.20
perm tm p C 0.89
perm tm t C 2.99
perm tm p A 0.33
perm tm t A 0.94
Pore Core H2O Molecules
Simulation Time (ns)

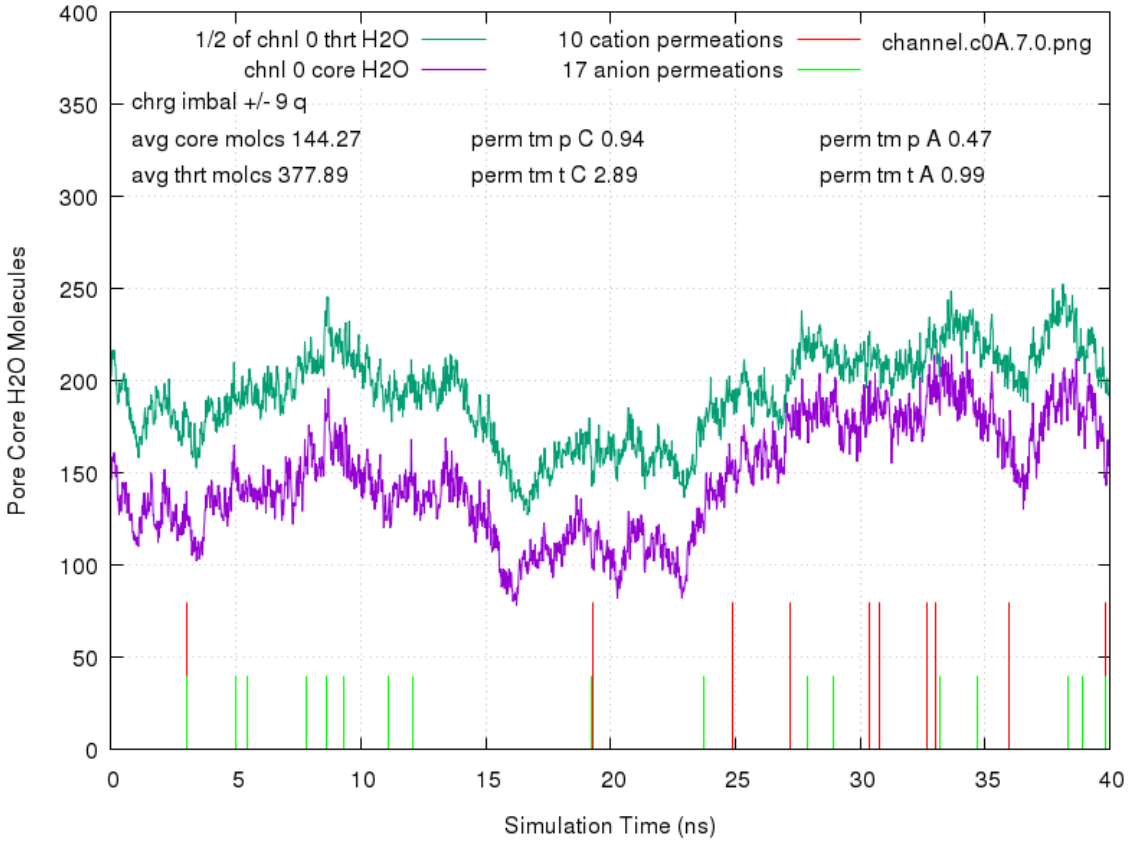

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
11 cation permeations
32 anion permeations
channel.c00.X.1.png
chrg imbal +/- 9 q
avg core molcs 168.05
avg thrt molcs 406.86
perm tm p C 0.71
perm tm t C 2.27
perm tm p A 0.37
perm tm t A 0.93
Pore Core H2O Molecules
Simulation Time (ns)

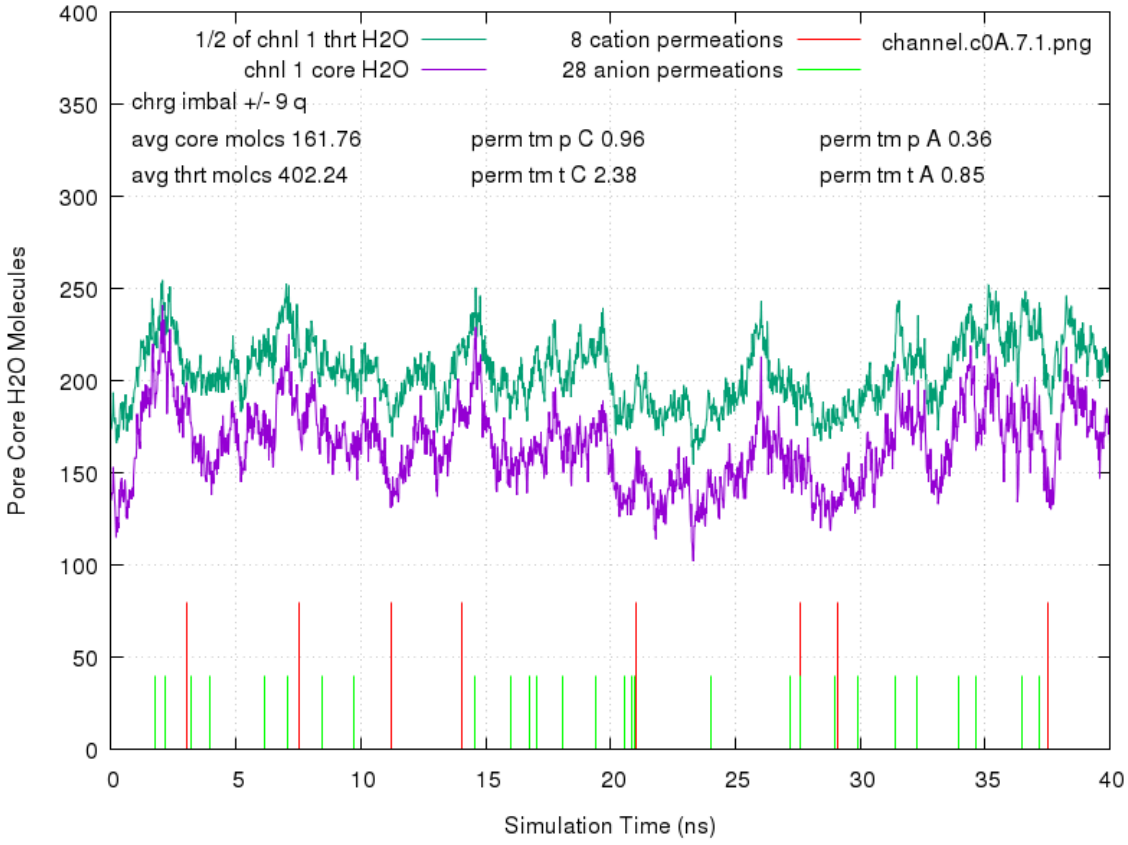

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
11 cation permeations
31 anion permeations
channel.c00.Y.0.png
chrg imbal +/- 9 q
avg core molcs 159.98
avg thrt molcs 397.84
perm tm p C 0.68
perm tm t C 2.17
perm tm p A 0.30
perm tm t A 0.78
Pore Core H2O Molecules
Simulation Time (ns)

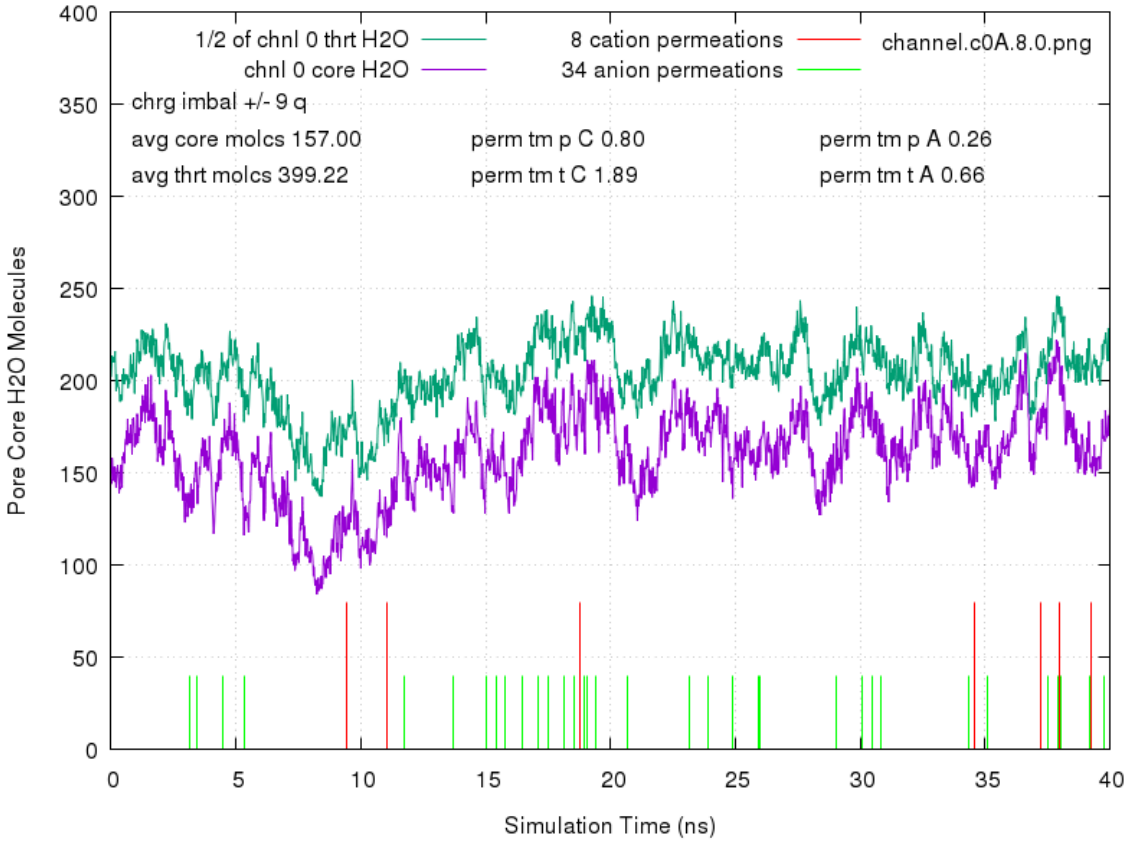

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
12 cation permeations
19 anion permeations
channel.c00.Y.1.png
chrg imbal +/- 9 q
avg core molcs 148.47
avg thrt molcs 372.36
perm tm p C 1.26
perm tm t C 2.77
perm tm p A 0.40
perm tm t A 0.96
Pore Core H2O Molecules
Simulation Time (ns)

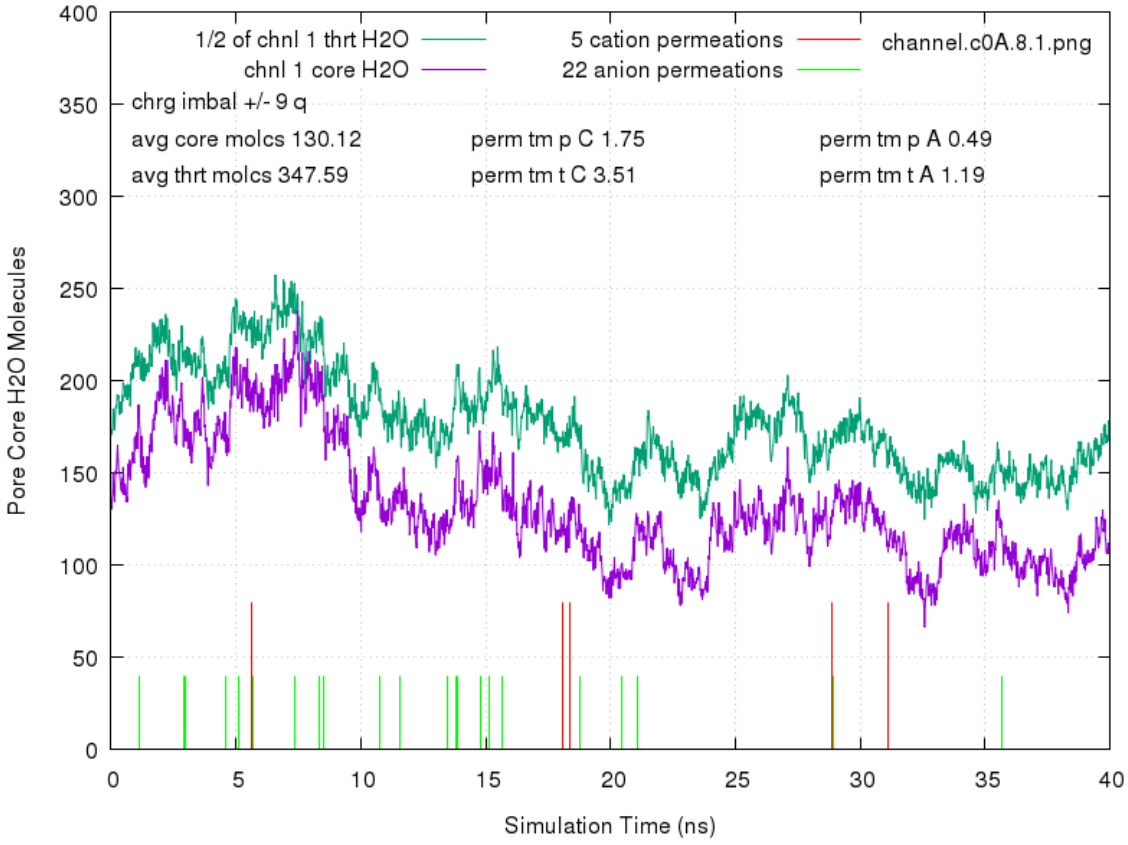

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
7 cation permeations
23 anion permeations
channel.c00.Z.0.png
chrg imbal +/- 9 q
avg core molcs 142.72
avg thrt molcs 380.53
perm tm p C 1.02
perm tm t C 2.68
perm tm p A 0.29
perm tm t A 0.75
Pore Core H2O Molecules
Simulation Time (ns)

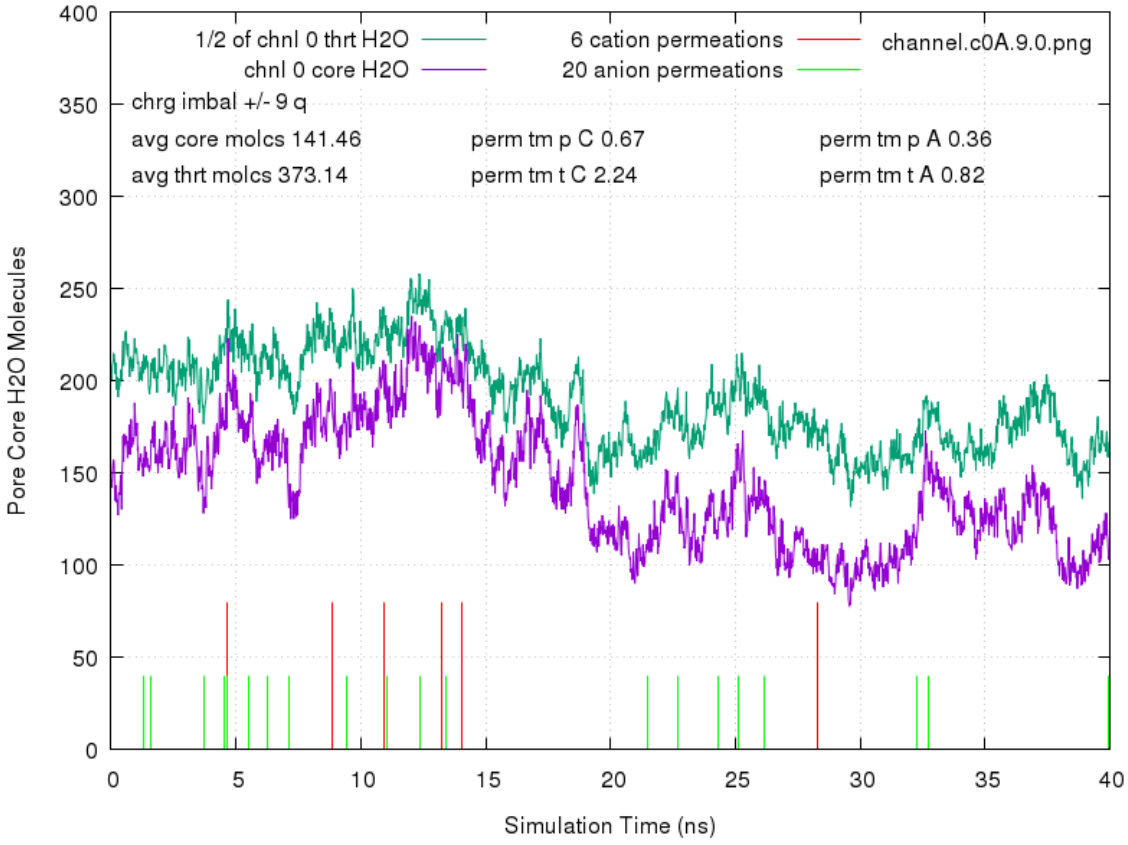

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
13 cation permeations
30 anion permeations
channel.c00.Z.1.png
chrg imbal +/- 9 q
avg core molcs 178.00
avg thrt molcs 424.89
perm tm p C 0.77
perm tm t C 1.96
perm tm p A 0.28
perm tm t A 0.78
Pore Core H2O Molecules
Simulation Time (ns)

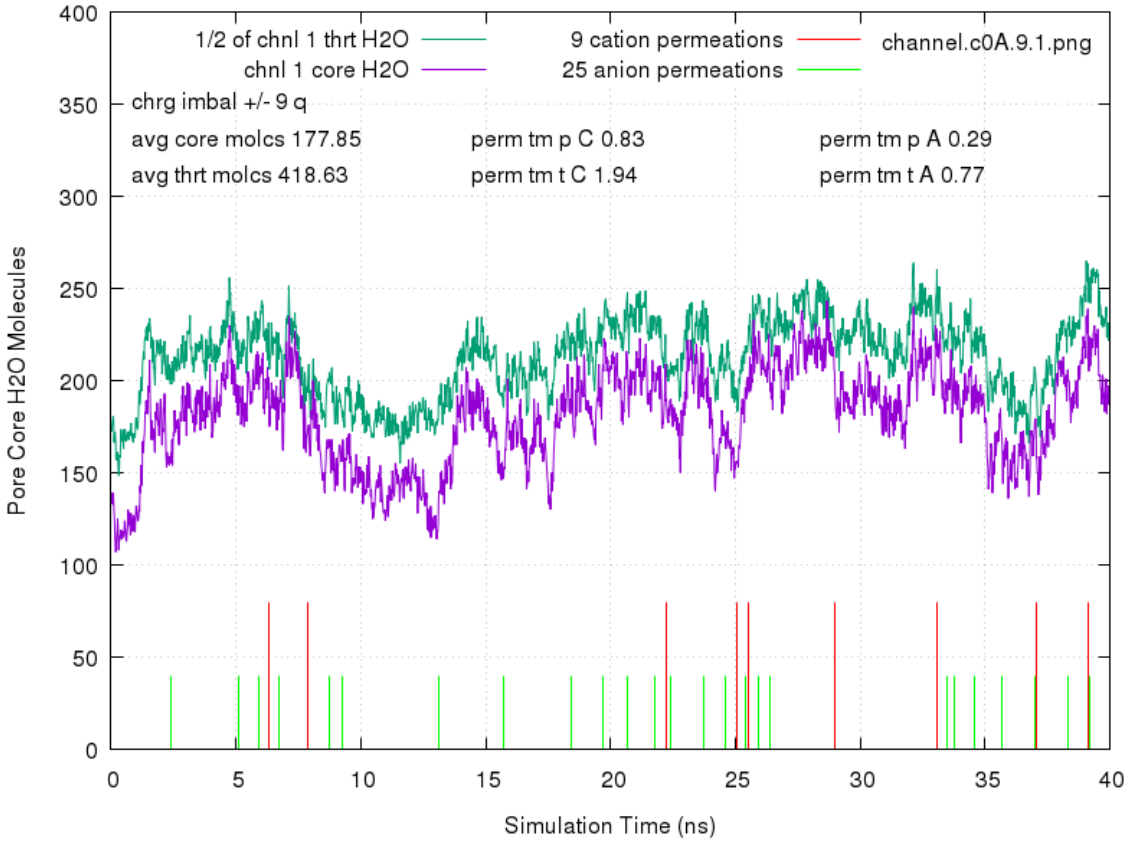

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
10 cation permeations
17 anion permeations
channel.c0A.7.0.png
chrg imbal +/- 9 q
avg core molcs 144.27
avg thrt molcs 377.89
perm tm p C 0.94
perm tm t C 2.89
perm tm p A 0.47
perm tm t A 0.99
Pore Core H2O Molecules
Simulation Time (ns)

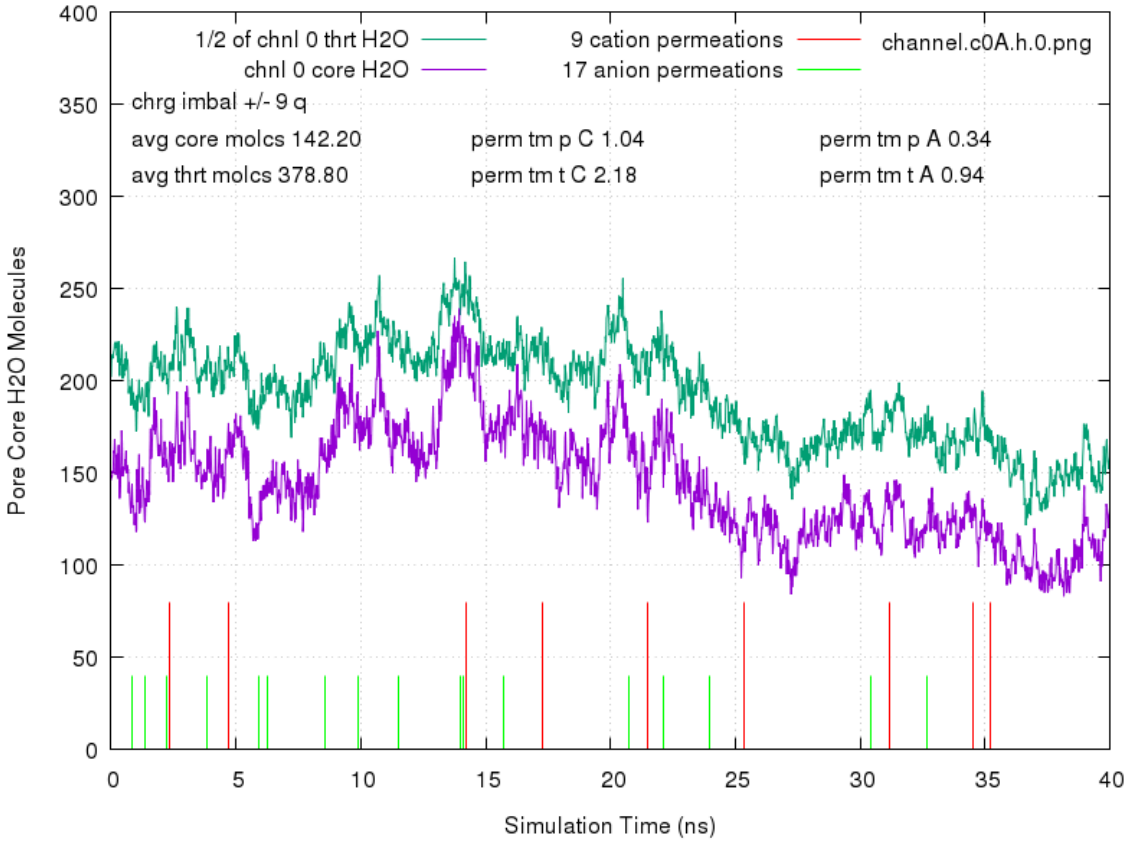

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
8 cation permeations
28 anion permeations
channel.c0A.7.1.png
chrg imbal +/- 9 q
avg core molcs 161.76
avg thrt molcs 402.24
perm tm p C 0.96
perm tm t C 2.38
perm tm p A 0.36
perm tm t A 0.85
Pore Core H2O Molecules
Simulation Time (ns)

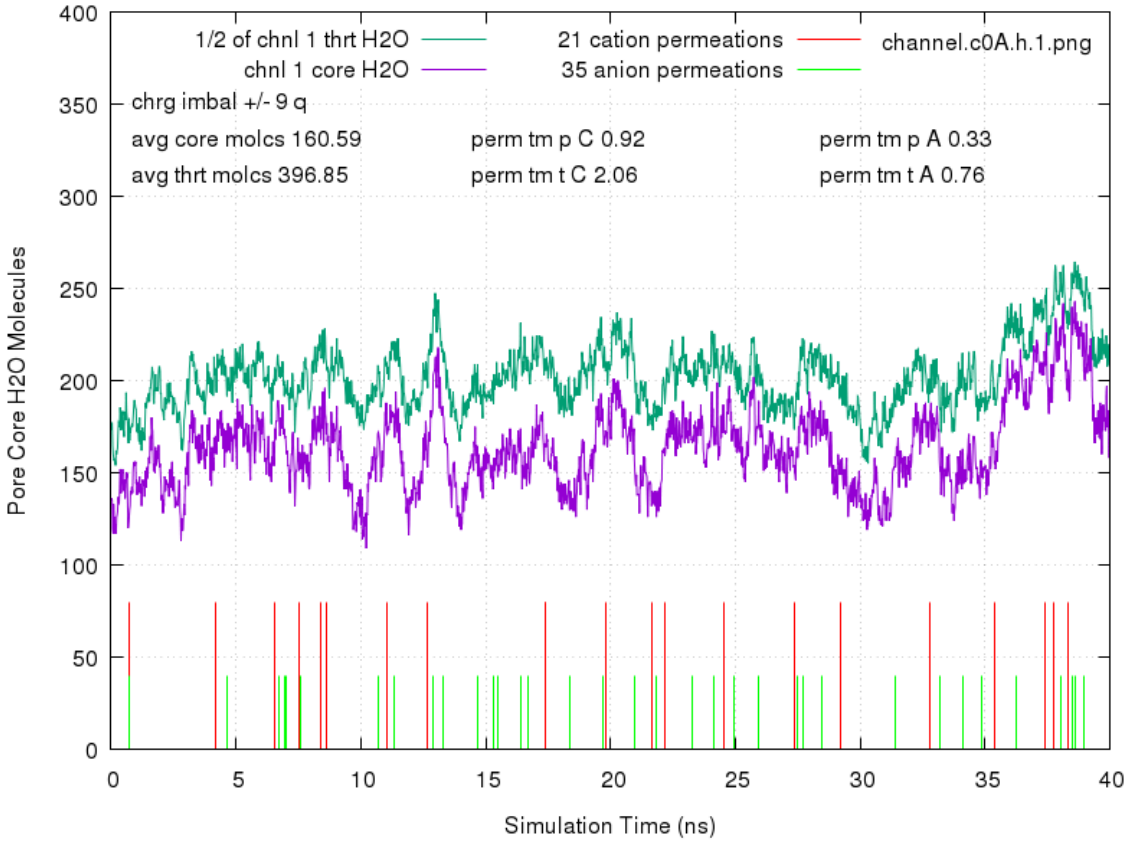

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
34 anion permeations
channel.c0A.8.0.png
chrg imbal +/- 9 q
avg core molcs 157.00
avg thrt molcs 399.22
perm tm p C 0.80
perm tm t C 1.89
perm tm p A 0.26
perm tm t A 0.66
Pore Core H2O Molecules
Simulation Time (ns)

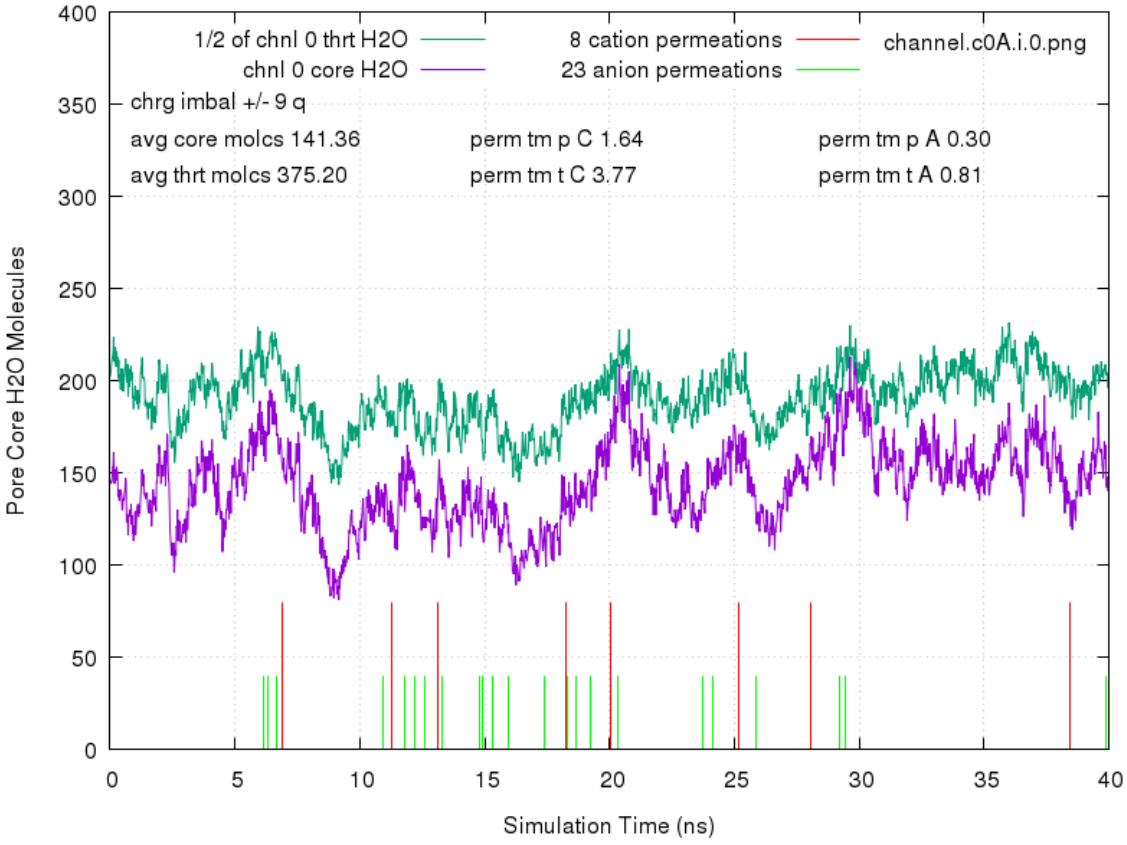

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
5 cation permeations
22 anion permeations
channel.c0A.8.1.png
chrg imbal +/- 9 q
avg core molcs 130.12
avg thrt molcs 347.59
perm tm p C 1.75
perm tm t C 3.51
perm tm p A 0.49
perm tm t A 1.19
Pore Core H2O Molecules
Simulation Time (ns)

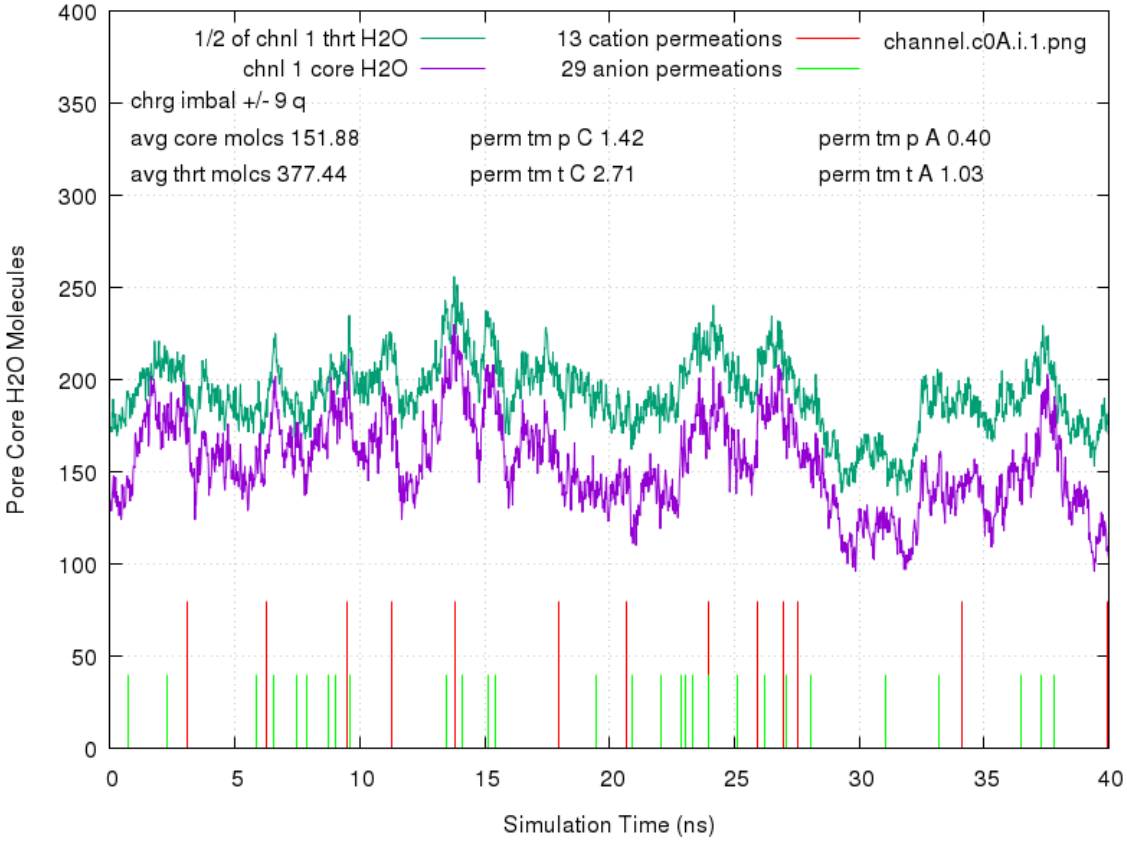

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
6 cation permeations
20 anion permeations
channel.c0A.9.0.png
chrg imbal +/- 9 q
avg core molcs 141.46
avg thrt molcs 373.14
perm tm p C 0.67
perm tm t C 2.24
perm tm p A 0.36
perm tm t A 0.82
Pore Core H2O Molecules
Simulation Time (ns)

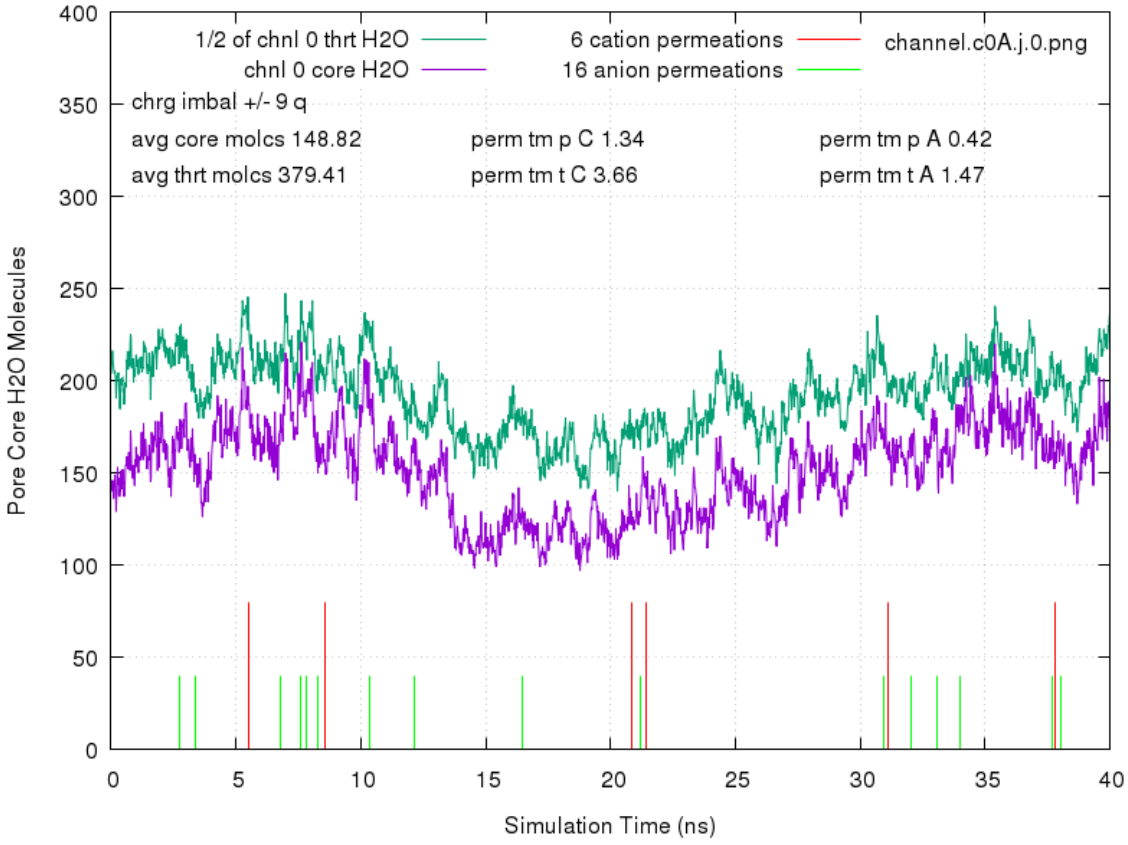

1/2 of chnl 1 thrt H2O
chnl 1 core H2O
9 cation permeations
25 anion permeations
channel.c0A.9.1.png
chrg imbal +/- 9 q
avg core molcs 177.85
avg thrt molcs 418.63
perm tm p C 0.83
perm tm t C 1.94
perm tm p A 0.29
perm tm t A 0.77
Pore Core H2O Molecules
Simulation Time (ns)

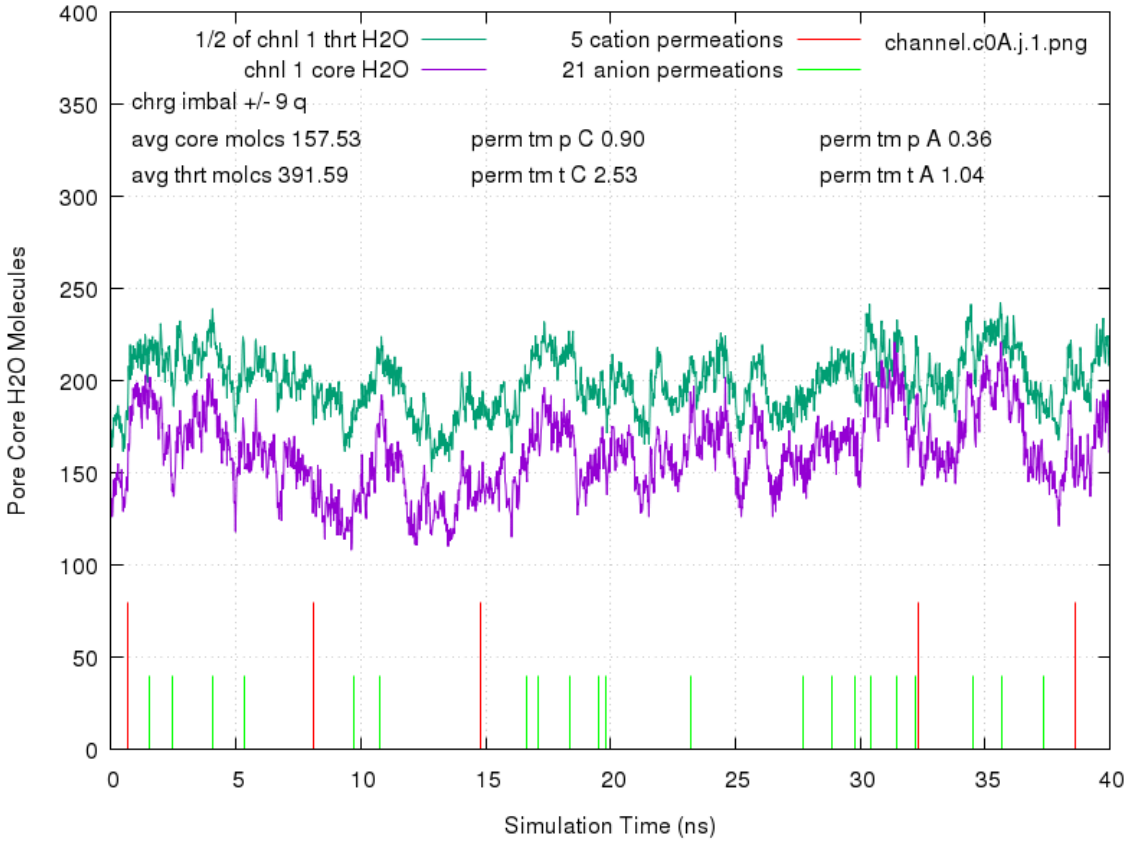

1/2 of chnl 0 thrt H2O
chnl 0 core H2O
9 cation permeations
17 anion permeations
channel.c0A.h.0.png
chrg imbal +/- 9 q
avg core molcs 142.20
avg thrt molcs 378.80
perm tm p C 1.04
perm tm t C 2.18
perm tm p A 0.34
perm tm t A 0.94
Pore Core H2O Molecules
Simulation Time (ns)


1/2 of chnl 1 thrt H2O
chnl 1 core H2O
21 cation permeations
35 anion permeations
channel.c0A.h.1.png
chrg imbal +/- 9 q
avg core molcs 160.59
avg thrt molcs 396.85
perm tm p C 0.92
perm tm t C 2.06
perm tm p A 0.33
perm tm t A 0.76
Pore Core H2O Molecules
Simulation Time (ns)


1/2 of chnl 0 thrt H2O
chnl 0 core H2O
8 cation permeations
23 anion permeations
channel.c0A.i.0.png
chrg imbal +/- 9 q
avg core molcs 141.36
avg thrt molcs 375.20
perm tm p C 1.64
perm tm t C 3.77
perm tm p A 0.30
perm tm t A 0.81
Pore Core H2O Molecules
Simulation Time (ns)


1/2 of chnl 1 thrt H2O
chnl 1 core H2O
13 cation permeations
29 anion permeations
channel.c0A.i.1.png
chrg imbal +/- 9 q
avg core molcs 151.88
avg thrt molcs 377.44
perm tm p C 1.42
perm tm t C 2.71
perm tm p A 0.40
perm tm t A 1.03
Pore Core H2O Molecules
Simulation Time (ns)


1/2 of chnl 0 thrt H2O
chnl 0 core H2O
6 cation permeations
16 anion permeations
channel.c0A.j.0.png
chrg imbal +/- 9 q
avg core molcs 148.82
avg thrt molcs 379.41
perm tm p C 1.34
perm tm t C 3.66
perm tm p A 0.42
perm tm t A 1.47
Pore Core H2O Molecules
Simulation Time (ns)


1/2 of chnl 1 thrt H2O
chnl 1 core H2O
5 cation permeations
21 anion permeations
channel.c0A.j.1.png
chrg imbal +/- 9 q
avg core molcs 157.53
avg thrt molcs 391.59
perm tm p C 0.90
perm tm t C 2.53
perm tm p A 0.36
perm tm t A 1.04
Pore Core H2O Molecules
Simulation Time (ns)

Appendix B

Section 2: Program Listings and Visualization Images

PROGRAM LISTINGS

```
#!/bin/bash -l
#================================================================================
# shift.     ### determines how much frames must be translated to make pore  ###
#            ### coordinate values unimodal and does so                      ###
# parameters
# $1         Processing Option
#   options: 0 - consider membrane 0 i.e. the top membrane
#            1 - consider membrane 1 i.e. the bottom membrane
#            run - consider both membranes and if necessary, produce a unique
#                  trajectory for each membrane
#_______________________________________________________________________________

thrshld=11

FR=`sed -n '/      0.000/{=;q}' twatc.xvg`

tail -n +${FR} twatc.xvg  | awk '{printf "%0.1d\n", $2}' | sort -k1,1n | uniq -c |
grep -v nan > datax
lines=`cat datax | wc -l`
j=1
matchx=0
firstx=0
transx=0
until [ ${j} -gt ${lines} ]; do
  patch=$(awk "NR==$j {print \$2}" datax)
  if [ ${matchx} -eq ${patch} ]; then
    molcs=$(awk "NR==$j {print \$1}" datax)
    if [ ${molcs} -gt ${thrshld} ]; then
```

```
      if [ ${firstx} -eq ${matchx} ]; then
        let "j=j+1"
        let "matchx=matchx+1"
        let "firstx=matchx"
      else
        gapx=$(( ${patch}-${firstx} ))
        transx=$(echo "scale=2;${patch} - ${gapx}/2" | bc)
        break 1
      fi
    else
      let "j=j+1"
    fi
  else
    molcs=$(awk "NR==$j {print \$1}" datax)
    if [ ${molcs} -gt ${thrshld} ]; then
      if [ ${firstx} -eq 0 -a ${patch} -lt 3 -a ${j} -le 2 ]; then
        break 1
      else
        gapx=$(( ${patch}-${firstx} ))
        transx=$(echo "scale=2;${patch} - ${gapx}/2" | bc)
        break 1
      fi
    else
      if [ ${firstx} -gt 0 ]; then
        transx=$(echo "scale=2;${matchx} + .25" | bc)
        break 1
      else
        let "matchx=matchx+1"
      fi
    fi
  fi
done
if [ ${j} -gt ${lines} ]; then
  transx=$(echo "scale=2;${patch} + 1.5" | bc)
fi

tail -n +${FR} twatc.xvg  | awk '{printf "%0.1d\n", $3}' | sort -k1,1n | uniq -c |
grep -v nan > datay
lines=`cat datay | wc -l`
j=1
matchy=0
firsty=0
transy=0
until [ $j -gt $lines ]; do
  patch=$(awk "NR==$j {print \$2}" datay)
  if [ ${matchy} -eq ${patch} ]; then
    molcs=$(awk "NR==$j {print \$1}" datay)
    if [ ${molcs} -gt ${thrshld} ]; then
      if [ ${firsty} -eq ${matchy} ]; then
        let "j=j+1"
        let "matchy=matchy+1"
        let "firsty=matchy"
      else
        gapy=$(( ${patch}-${firsty} ))
        transy=$(echo "scale=2;${patch} - ${gapy}/2" | bc)
        break 1
      fi
    else
      let "j=j+1"
    fi
  else
    molcs=$(awk "NR==$j {print \$1}" datay)
    if [ ${molcs} -gt ${thrshld} ]; then
      if [ ${firsty} -eq 0 -a ${patch} -lt 3 -a ${j} -le 2 ]; then
        break 1
      else
        gapy=$(( ${patch}-${firsty} ))
```

```
        transy=$(echo "scale=2;${patch} - ${gapy}/2" | bc)
        break 1
      fi
    else
      if [ ${firsty} -gt 0 ]; then
        transy=$(echo "scale=2;${matchy} + .25" | bc)
        break 1
      else
        let "matchy=matchy+1"
      fi
    fi
  fi
done
if [ ${j} -gt ${lines} ]; then
  transy=$(echo "scale=2;${patch} + 1.5" | bc)
fi

echo top shift "${transx}" "${transy}"
if ! [ "${transx}" == "0" -a "${transy}" == "0" ]; then
  if [ "${1}" == "0" -o "${1}" == "run" ]; then
    gmx trjconv -f trajout.xtc -s topol.tpr -o trajout0.xtc -pbc mol -trans -
${transx} -${transy} 0 < <(echo 0)
  fi
fi

FR=`sed -n '/      0.000/{=;q}' bwatc.xvg`

tail -n +${FR} bwatc.xvg  | awk '{printf "%0.1d\n", $2}' | sort -k1,1n | uniq -c |
grep -v nan > dabax
lines=`cat dabax | wc -l`
j=1
matchx=0
firstx=0
transx=0
until [ $j -gt $lines ]; do
  patch=$(awk "NR==$j {print \$2}" dabax)
  if [ ${matchx} -eq ${patch} ]; then
    molcs=$(awk "NR==$j {print \$1}" dabax)
    if [ ${molcs} -gt ${thrshld} ]; then
      if [ ${firstx} -eq ${matchx} ]; then
        let "j=j+1"
        let "matchx=matchx+1"
        let "firstx=matchx"
      else
        gapx=$(( ${patch}-${firstx} ))
        transx=$(echo "scale=2;${patch} - ${gapx}/2" | bc)
        break 1
      fi
    else
      let "j=j+1"
    fi
  else
    molcs=$(awk "NR==$j {print \$1}" dabax)
    if [ ${molcs} -gt ${thrshld} ]; then
      if [ ${firstx} -eq 0 -a ${patch} -lt 2 ]; then
        break 1
      else
        gapx=$(( ${patch}-${firstx} ))
        transx=$(echo "scale=2;${patch} - ${gapx}/2" | bc)
        break 1
      fi
    else
      let "matchx=matchx+1"
    fi
  fi
done
if [ ${j} -gt ${lines} ]; then
```

```
  transx=$(echo "scale=2;${patch} + 1.5" | bc)
fi

tail -n +${FR} bwatc.xvg  | awk '{printf "%0.1d\n", $3}' | sort -k1,1n | uniq -c |
grep -v nan > dabay
lines=`cat dabay | wc -l`
j=1
matchy=0
firsty=0
transy=0
until [ $j -gt $lines ]; do
  patch=$(awk "NR==$j {print \$2}" dabay)
  if [ ${matchy} -eq ${patch} ]; then
    molcs=$(awk "NR==$j {print \$1}" dabay)
    if [ ${molcs} -gt ${thrshld} ]; then
      if [ ${firsty} -eq ${matchy} ]; then
        let "j=j+1"
        let "matchy=matchy+1"
        let "firsty=matchy"
      else
        gapy=$(( ${patch}-${firsty} ))
        transy=$(echo "scale=2;${patch} - ${gapy}/2" | bc)
        break 1
      fi
    else
      let "j=j+1"
    fi
  else
    molcs=$(awk "NR==$j {print \$1}" dabay)
    if [ ${molcs} -gt ${thrshld} ]; then
      if [ ${firsty} -eq 0 -a ${patch} -lt 2 ]; then
        break 1
      else
        gapy=$(( ${patch}-${firsty} ))
        transy=$(echo "scale=2;${patch} - ${gapy}/2" | bc)
        break 1
      fi
    else
      let "matchy=matchy+1"
    fi
  fi
done
if [ ${j} -gt ${lines} ]; then
  transy=$(echo "scale=2;${patch} + 1.5" | bc)
fi

echo bottom shift "${transx}" "${transy}"
if ! [ "${transx}" == "0" -a "${transy}" == "0" ]; then
  if [ "${1}" == "1" -o "${1}" == "run" ]; then
    gmx trjconv -f trajout.xtc -s topol.tpr -o trajout1.xtc -pbc mol -trans -
${transx} -${transy} 0 < <(echo 0)
  fi
fi


#!/usr/bin/bash -l
#================================================================================
# select.    ### script for measuring and counting water and ions            ###
#            ###                                                             ###
# parameters
# $1         routine name
#   options: twat,twatt,twatc,twattc,tca,tan,tcat,tant
#            bwat,bwatt,bwatc,bwattc,bca,ban,bcat,bant
#_______________________________________________________________________________

if [ -s trajout0.xtc -o -s trajout0c.xtc ]; then
  if [ -s trajout0c.xtc ]; then
```

```
    tfyle="trajout0c.xtc"
  else
    tfyle="trajout0.xtc"
  fi
else
  tfyle="trajout.xtc"
fi
if [ -s trajout1.xtc -o -s trajout1c.xtc ]; then
  if [ -s trajout1c.xtc ]; then
    bfyle="trajout1c.xtc"
  else
    bfyle="trajout1.xtc"
  fi
else
  bfyle="trajout.xtc"
fi

case ${1} in
  twat)
    if [ -s twat.xvg ]; then source ../jobs/roll. twat.xvg; fi
    gmx select -f ${tfyle} -s topol.tpr -os twat.xvg -select \
                                                      ' \
    lhgrps = resname "DOPC" and resid < 10000 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid < 1000 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname TIP3 and ( (z+0.75) > zl and (z-0.75) < zl ) and radius <
1.5;'
    ;;
  twatt)
    if [ -s twatt.xvg ]; then source ../jobs/roll. twatt.xvg; fi
    gmx select -f ${tfyle} -s topol.tpr -os twatt.xvg -select \
                                                       ' \
    lhgrps = resname "DOPC" and resid < 10000 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid < 10000 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname TIP3 and ( (z+1.5) > zl and (z-1.5) < zl ) and radius <
1.5;'
    ;;
  twatc)
    if [ -s twatc.xvg ]; then source ../jobs/roll. twatc.xvg; fi
    gmx trajectory -f ${tfyle} -s topol.tpr -ox twatc.xvg -pbc yes -select \
                                                                    ' \
    lhgrps = resname "DOPC" and resid < 10000 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of resname TIP3 and ( (z+.45)>zm and (z-.45)<zm ); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
```

```
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid < 10000 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
    core   = resname TIP3 and ( (z+.75)>zl and (z-.75)<zl ) and radius < 1.5; \
                            \
    cog of core;'
    ;;
  twctc)
    if [ -s twctc.xvg ]; then source ../jobs/roll. twctc.xvg; fi
    gmx trajectory -f ${tfyle} -s topol.tpr -ox twctc.xvg -pbc yes -select \
                                                                        ' \
    lhgrps = resname "DOPC" and resid < 10000 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of resname TIP3 and ( (z+.45)>zm and (z-.45)<zm ); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid < 10000 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
    core   = resname TIP3 and ( (z+.75)>zl and (z-.75)<zl ) and radius < 1.5; \
                            \
    cog of core;'
    ;;
  tca)
    if [ -s tca.xvg ]; then source ../jobs/roll. tca.xvg; fi
    gmx select -f ${tfyle} -s topol.tpr -os tca.xvg -select \
                                                          ' \
    lhgrps = resname "DOPC" and resid < 10000 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid < 10000 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname SOD and ( (z+0.75) > zl and (z-0.75) < zl ) and radius <
1.5;'
    ;;
  tan)
    if [ -s tan.xvg ]; then source ../jobs/roll. tan.xvg; fi
    gmx select -f ${tfyle} -s topol.tpr -os tan.xvg -select \
                                                          ' \
    lhgrps = resname "DOPC" and resid < 10000 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid < 10000 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname CLA and ( (z+0.75) > zl and (z-0.75) < zl ) and radius <
1.5;'
    ;;
  tcat)
    if [ -s tcat.xvg ]; then source ../jobs/roll. tcat.xvg; fi
```

```
    gmx select -f ${tfyle} -s topol.tpr -os tcat.xvg -select \
                                                              ' \
    lhgrps = resname "DOPC" and resid < 10000 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid < 10000 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname SOD and ( (z+1.5) > zl and (z-1.5) < zl ) and radius <
1.5;'
    ;;
  tant)
    if [ -s tant.xvg ]; then source ../jobs/roll. tant.xvg; fi
    gmx select -f ${tfyle} -s topol.tpr -os tant.xvg -select \
                                                              ' \
    lhgrps = resname "DOPC" and resid < 10000 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid < 10000 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname CLA and ( (z+1.5) > zl and (z-1.5) < zl ) and radius <
1.5;'
    ;;
  bwat)
    if [ -s bwat.xvg ]; then source ../jobs/roll. bwat.xvg; fi
    gmx select -f ${bfyle} -s topol.tpr -os bwat.xvg -select \
                                                            ' \
    lhgrps = resname "DOPC" and resid > 9999 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid > 9999 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname TIP3 and ( (z+0.75) > zl and (z-0.75) < zl ) and radius <
1.5;'
   ;;
  bwatt)
    if [ -s bwatt.xvg ]; then source ../jobs/roll. bwatt.xvg; fi
    gmx select -f ${bfyle} -s topol.tpr -os bwatt.xvg -select \
                                                              ' \
    lhgrps = resname "DOPC" and resid > 9999 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
```

```
    lipids = resname "DOPC" and resid > 9999 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname TIP3 and ( (z+1.5) > zl and (z-1.5) < zl ) and radius <
1.5;'
   ;;
  bwatc)
    if [ -s bwatc.xvg ]; then source ../jobs/roll. bwatc.xvg; fi
    gmx trajectory -f ${bfyle} -s topol.tpr -ox bwatc.xvg -select \
                                                                  ' \
    lhgrps = resname "DOPC" and resid > 9999 and (name "P.*" or name "N.*" or name
"O." or name "O[12].."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of resname TIP3 and ( (z+.45)>zm and (z-.45)<zm ); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid > 9999 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
    core   = resname TIP3 and ( (z+.75)>zl and (z-.75)<zl ) and radius < 1.5; \
                            \
     cog of core;'
    ;;
  bwctc)
    if [ -s bwctc.xvg ]; then source ../jobs/roll. bwctc.xvg; fi
    gmx trajectory -f ${bfyle} -s topol.tpr -ox bwctc.xvg -select \
                                                                  ' \
    lhgrps = resname "DOPC" and resid > 9999 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of resname TIP3 and ( (z+.45)>zm and (z-.45)<zm ); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid > 9999 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
    core   = resname TIP3 and ( (z+.75)>zl and (z-.75)<zl ) and radius < 1.5; \
                            \
    cog of core;'
    ;;
  bca)
    if [ -s bca.xvg ]; then source ../jobs/roll. bca.xvg; fi
    gmx select -f ${bfyle} -s topol.tpr -os bca.xvg -select \
                                                          ' \
    lhgrps = resname "DOPC" and resid > 9999 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid > 9999 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname SOD and ( (z+0.75) > zl and (z-0.75) < zl ) and radius <
1.5;'
   ;;
  ban)
    if [ -s ban.xvg ]; then source ../jobs/roll. ban.xvg; fi
    gmx select -f ${bfyle} -s topol.tpr -os ban.xvg -select \
```

```
                                                                ' \
    lhgrps = resname "DOPC" and resid > 9999 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid > 9999 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname CLA and ( (z+0.75) > zl and (z-0.75) < zl ) and radius <
1.5;'
   ;;
  bcat)
    if [ -s bcat.xvg ]; then source ../jobs/roll. bcat.xvg; fi
    gmx select -f ${bfyle} -s topol.tpr -os bcat.xvg -select \
                                                            ' \
    lhgrps = resname "DOPC" and resid > 9999 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid > 9999 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname SOD and ( (z+1.5) > zl and (z-1.5) < zl ) and radius <
1.5;'
   ;;
  bant)
    if [ -s bant.xvg ]; then source ../jobs/roll. bant.xvg; fi
    gmx select -f ${bfyle} -s topol.tpr -os bant.xvg -select \
                                                            ' \
    lhgrps = resname "DOPC" and resid > 9999 and (name "P.*" or name "N.*" or name
"O." or name "O[12]."); \
    zm     = z of cog of lhgrps; \
    wcog   = cog of (resname TIP3 and (z+.45) > zm and (z-.45) < zm); \
    xw     = x of wcog; \
    yw     = y of wcog; \
    zw     = z of wcog; \
    radius = ((x-xw)^2+(y-yw)^2)^0.5; \
    lipids = resname "DOPC" and resid > 9999 and radius < 1.5; \
    lcog   = cog of lipids; \
    zl     = z of lcog;     \
                            \
    dyn_mol_cog of resname CLA and ( (z+1.5) > zl and (z-1.5) < zl ) and radius <
1.5;'
   ;;
  *)
    echo select : ${1} is not a valid case
    ;;
esac
```

```
#!/bin/bash -l
#================================================================================
# cross.     ### extracts the counts of ions that have crossed thru the      ###
#            ### membrane                                                     ###
# parameters
# none       N/A
#   options: N/A
#_______________________________________________________________________________

if [ -s ch0+AB.xvg ]; then rm ch0+AB.xvg; fi
if [ -s ch0-AB.xvg ]; then rm ch0-AB.xvg; fi
if [ -s ch1+AB.xvg ]; then rm ch1+AB.xvg; fi
if [ -s ch1-AB.xvg ]; then rm ch1-AB.xvg; fi

tail -n +63 swapions.xvg | grep -A 1 finished | grep -v finished | cut -c1-12,179-
182 | grep "\-\-" -v | uniq -s 13 | grep -v "   0" | sort -k 2 | uniq -f 1 >
ch0+AB.xvg
tail -n +63 swapions.xvg | grep -A 1 finished | grep -v finished | cut -c1-12,189-
192 | grep "\-\-" -v | uniq -s 13 | grep -v "   0" | sort -k 2 | uniq -f 1 > ch0-
AB.xvg
tail -n +63 swapions.xvg | grep -A 1 finished | grep -v finished | cut -c1-12,159-
162 | grep "\-\-" -v | uniq -s 13 | grep -v "   0" | sort -k 2 | uniq -f 1 >
ch1+AB.xvg
tail -n +63 swapions.xvg | grep -A 1 finished | grep -v finished | cut -c1-12,169-
172 | grep "\-\-" -v | uniq -s 13 | grep -v "   0" | sort -k 2 | uniq -f 1 > ch1-
AB.xvg

tail -n 1 swapions.xvg | awk -F " " '{print $18 " " $19 " " $16 " " $17}' >
crossers.xvg


#!/usr/gnuplot
#================================================================================
# cross      ### plots simulation history of channel water during a          ###
#            ### simulation this script plots the data for both channels 0 & ###
#            ### 1                                                            ###
# parameters
# none       N/A
#   options: N/A
#_______________________________________________________________________________

set term pngcairo size 800,600 font "helvetica,12"
set tmargin 1
set lmargin 0.97
set key top left maxrows 2
set grid xtics mxtics ytics mytics y2tics my2tics
set xlabel "Simulation Time (ns)"
set ylabel "Pore Core H2O Molecules"
set xrange [0:40]
set yrange [0:400]
set y2range [0:800]

ca=system("awk '{print $1}' crossers.xvg")
an=system("awk '{print sqrt(($2)^2)}' crossers.xvg")
cal=sprintf(ca)." cation permeations"
anl=sprintf(an)." anion permeations"
ch0w=system("awk '{printf(\"%4.2f\",$3)}' twat.std")
ch0d=system("awk '{printf(\"%4.2f\",$3)}' twatt.std")
ch0C=system("awk '{printf(\"%4.2f\",$1)}' tca.tot")
ch0Cd=system("awk '{printf(\"%4.2f\",$1)}' tcat.tot")
ch0A=system("awk '{printf(\"%4.2f\",$1)}' tan.tot")
ch0Ad=system("awk '{printf(\"%4.2f\",$1)}' tant.tot")
if (ca == 0) {ca=1;ch0C=0;ch0Cd=0}
if (an == 0) {an=1;ch0A=0;ch0Ad=0}
pt0C="perm tm p C ".sprintf("%.2f",ch0C*.02/ca)
pt0Cd="perm tm t C ".sprintf("%.2f",ch0Cd*.02/ca)
pt0A="perm tm p A ".sprintf("%.2f",ch0A*.02/an)
```

```
pt0Ad="perm tm t A ".sprintf("%.2f",ch0Ad*.02/an)
AVGWT="avg core molcs ".sprintf(ch0w)
AVGWTd="avg thrt molcs ".sprintf(ch0d)
CIMBL="chrg imbal +/- ".system(sprintf("tail -n 1 swapions.xvg | awk '{print
sqrt(($2-$5)^2)}'"))." q"

set label AVGWT at graph 0.02,0.83
set label AVGWTd at graph 0.02,0.78
set label pt0C at graph 0.36,0.83
set label pt0Cd at graph 0.36,0.78
set label pt0A at graph 0.71,0.83
set label pt0Ad at graph 0.71,0.78
set label CIMBL at graph 0.02,0.88

NAME1="channel.".system(sprintf("pwd | rev | cut -f2 -d'/' - |
rev"))."."system(sprintf("pwd | rev | cut -f1 -d'/' - | rev"))."0.png"
set title NAME1 offset 29,-2.5
set output NAME1

pl "twatt.xvg" u ($1/1000):2 title "1/2 of chnl 0 thrt H2O" w l lw 1 lc 2 axes x1y2,
"twat.xvg" u ($1/1000):2 title "chnl 0 core H2O" with lines lw 1 lc rgb "#9400D3",
"ch0+AB.xvg" u ($1/1000):(80) title cal w impulses lc rgb "#FF0000", "ch0-AB.xvg" u
($1/1000):(40) title anl w impulses lc rgb "#00FF00"

reset session
set tmargin 1
set key top left maxrows 2
set grid xtics mxtics ytics mytics y2tics my2tics
set xlabel "Simulation Time (ns)"
set ylabel "Pore Core H2O Molecules"
set yrange [0:400]
set y2range [0:800]

ca=system("awk '{print $3}' crossers.xvg")
an=system("awk '{print sqrt(($4)^2)}' crossers.xvg")
cal=sprintf(ca)." cation permeations"
anl=sprintf(an)." anion permeations"
ch1w=system("awk '{printf(\"%4.2f\",$3)}' bwat.std")
ch1d=system("awk '{printf(\"%4.2f\",$3)}' bwatt.std")
ch1C=system("awk '{printf(\"%4.2f\",$1)}' bca.tot")
ch1Cd=system("awk '{printf(\"%4.2f\",$1)}' bcat.tot")
ch1A=system("awk '{printf(\"%4.2f\",$1)}' ban.tot")
ch1Ad=system("awk '{printf(\"%4.2f\",$1)}' bant.tot")
if (ca == 0) {ca=1;ch1C=0;ch1Cd=0}
if (an == 0) {an=1;ch1A=0;ch1Ad=0}
pt1C="perm tm p C ".sprintf("%.2f",ch1C*.02/ca)
pt1Cd="perm tm t C ".sprintf("%.2f",ch1Cd*.02/ca)
pt1A="perm tm p A ".sprintf("%.2f",ch1A*.02/an)
pt1Ad="perm tm t A ".sprintf("%.2f",ch1Ad*.02/an)
AVGWT="avg core molcs ".sprintf(ch1w)
AVGWTd="avg thrt molcs ".sprintf(ch1d)
CIMBL="chrg imbal +/- ".system(sprintf("tail -n 1 swapions.xvg | awk '{print
sqrt(($2-$5)^2)}'"))." q"

set label AVGWT at graph 0.02,0.83
set label AVGWTd at graph 0.02,0.78
set label pt1C ai graph 0.36,0.83
set label pt1Cd at graph 0.36,0.78
set label pt1A at graph 0.71,0.83
set label pt1Ad at graph 0.71,0.78
set label CIMBL at graph 0.02,0.88

NAME1="channel.".system(sprintf("pwd | rev | cut -f2 -d'/' - |
rev"))."."system(sprintf("pwd | rev | cut -f1 -d'/' - | rev"))."1.png"
set title NAME1 offset 29,-2.5
set output NAME1
```

```
pl "bwatt.xvg" u ($1/1000):2 title "1/2 of chnl 1 thrt H2O" w l lw 1 lc 2 axes x1y2,
"bwat.xvg" u ($1/1000):2 title "chnl 1 core H2O" with lines lw 1 lc rgb "#9400D3",
"ch1+AB.xvg" u ($1/1000):(80) title cal w impulses lc rgb "#FF0000", "ch1-AB.xvg" u
($1/1000):(40) title anl w impulses lc rgb "#00FF00"

unset output


#!/bin/bash -l
#================================================================================
# count.     ### count pore water molecules and ions                          ###
#            ###                                                              ###
# parameters
# $1         processing option
#   options: <null> - count everything
#            T  - count everything in channel 0 (top)
#            t  - count waters in channel 0
#            ti - count ions in channel 0
#            B  - count everything in channel 1 (bot)
#            t  - count waters in channel 1
#            ti - count ions in channel 1
#________________________________________________________________________________

declare -a fyles
fyles=(twat twatt tca tan tcat tant bwat bwatt bca ban bcat bant)

i=0
last=11
if [ "${1}" == "T" ]; then
  last=5
elif [ "${1}" == "t" ]; then
  last=1
elif [ "${1}" == "ti" ]; then
  i=2
  last=5
elif [ "${1}" == "B" ]; then
  i=6
elif [ "${1}" == "b" ]; then
  i=6
  last=7
elif [ "${1}" == "bi" ]; then
  i=8
fi

while [ ${i} -le $(134) ]; do
  if [ -a ${fyles[$i]}.xvg ]; then rm ${fyles[$i]}.xvg; fi
  source ../jobs/select. ${fyles[$i]}
  let "i=i+1"
done

for i in {0..1} {6..7}; do
  source ../jobs/stddev.awk ${fyles[$i]}.xvg > ${fyles[$i]}.std
done


#!/bin/bash -l
#================================================================================
# center.    ### script for centering pores in trajectories                   ###
#            ###                                                              ###
# parameters
# $1         channel to center
#   options: 0 - channel 0 in the top membrane
#            1 - channel 1 in the bottom membrane
#________________________________________________________________________________

case ${1} in
 0)
```

```
   if [ -s trajout0.xtc ]; then
     tfyle="trajout0.xtc"
   else
     tfyle="trajout.xtc"
   fi

   if ! ( [ -a boxcntr.xvg ] ); then
     source ../jobs/boxdims.
   fi

   declare -a boxcntr
   boxcntr=($(cat boxcntr.xvg))

   if [ "${tfyle}" == "trajout0.xtc" -a ${tfyle} -nt "twatc.xvg" ]; then
     source ../jobs/roll. twatc.xvg
     source ../jobs/select. twatc
   fi

   if ! ( [ -s twatc.xvg ] ); then source ../jobs/select. twatc; fi

   s ../jobs/trans. ${boxcntr[0]} ${boxcntr[1]} ${boxcntr[2]} twatc.xvg

   gmx trjconv -f ${tfyle} -o /local/scratch/temp.gro -pbc mol -sep -exec 'source
../jobs/translate.' < <(echo 0)

   if [ -s "trajout0c.xtc" ]; then source ../jobs/roll. trajout0c.xtc; fi

   gmx trjcat -f /local/scratch/tump{0..2000}.xtc -o trajout0c.xtc

   rm /local/scratch/tump*.xtc

   tfyle="trajout0c.xtc"
   source ../jobs/select. twatc
   s ../jobs/trans. ${boxcntr[0]} ${boxcntr[1]} ${boxcntr[2]} twatc.xvg
   gmx trjconv -f ${tfyle} -o /local/scratch/temp.gro -pbc mol -sep -exec 'source
../jobs/translate.' < <(echo 0)
   source ../jobs/roll. trajout0c.xtc
   gmx trjcat -f /local/scratch/tump{0..2000}.xtc -o trajout0c.xtc
   rm /local/scratch/tump*.xtc
   ;;

 1)
   if [ -s trajout1.xtc ]; then
     bfyle="trajout1.xtc"
   else
     bfyle="trajout.xtc"
   fi

   if ! ( [ -a boxcntr.xvg ] ); then
     source ../jobs/boxdims.
   fi

   declare -a boxcntr
   boxcntr=($(cat boxcntr.xvg))

   if [ "${bfyle}" == "trajout1.xtc" -a ${bfyle} -nt "bwatc.xvg" ]; then
     source ../jobs/roll. bwatc.xvg
     source ../jobs/select. bwatc
   fi

   if ! ( [ -s bwatc.xvg ] ); then source ../jobs/select. bwatc; fi

   source ../jobs/trans. ${boxcntr[0]} ${boxcntr[1]} ${boxcntr[2]} bwatc.xvg

   gmx trjconv -f ${bfyle} -o /local/scratch/temp.gro -pbc mol -sep -exec 'source
../jobs/translate.' < <(echo 0)
```

```
   if [ -s "trajout1c.xtc" ]; then source ../jobs/roll. trajout1c.xtc; fi

   gmx trjcat -f /local/scratch/tump{0..2000}.xtc -o trajout1c.xtc

   rm /local/scratch/tump*.xtc

   bfyle="trajout1c.xtc"
   source ../jobs/select. bwatc
   source ../jobs/trans. ${boxcntr[0]} ${boxcntr[1]} ${boxcntr[2]} bwatc.xvg
   gmx trjconv -f ${bfyle} -o /local/scratch/temp.gro -pbc mol -sep -exec 'source
../jobs/translate.' < <(echo 0)
   source ../jobs/roll. trajout1c.xtc
   gmx trjcat -f /local/scratch/tump{0..2000}.xtc -o trajout1c.xtc
   rm /local/scratch/tump*.xtc
   ;;

 *)
   echo center : ${1} is not a valid case
   ;;
esac


#!/bin/bash -l
#================================================================================
# boxdims. ### Calculates the average simulation box dimensions and average  ###
#          ### coordinates of the simulation box center                       ###
#
# $1       Simulation ensemble
# options: <null> - check .mdp file to determine
#          npt - calculate values from energy.xvg file
#          nvt - calculate values from dimensions in conf.gro file
#_______________________________________________________________________________

ARG1=${1:-'mt'}

if [ "${ARG1}" == "mt" ]; then
  if [ `grep -m 1 "pcoupl" mdout.mdp | awk '{print $3}'` == "no" ]; then
    ARG1="nvt"
  else
    ARG1="npt"
  fi
fi

case ${ARG1} in
  npt)

    if [ -s "energy.xvg" ]; then source ../jobs/roll. energy.xvg; fi

    gmx energy < <(echo -e "Box-X\nBox-Y\nBox-Z\n \n") | grep Box- | awk '{print
$2}' | paste - - - > boxdims.xvg
    rm energy.xvg
    gmx energy < <(echo -e "Box-X\nBox-Y\nBox-Z\n \n") | grep Box- | awk '{print
$2/2}' | paste - - - > boxcntr.xvg
    rm energy.xvg
  ;;
  nvt)

    tail -n 1 conf.gro > boxdims.xvg
    tail -n 1 conf.gro | awk '{print $1/2, $2/2, $3/2}' > boxcntr.xvg

  ;;
esac
```

```
#!/bin/bash -l
#================================================================================
# roll.      ### adds index numbers to file names to prevent overwrite. this ###
#            ### preserves the file extension so gromacs utilities will      ###
#            ### still accept them as input files                            ###
# parameters
# $1         file name to roll
#   example: with files conf.gro & conf.00.gro, source roll. conf.gro
#            moves previous files to conf.00.gro & conf.01.gro
#            therefore, lowest index "00" is always most recent file
#_______________________________________________________________________________

if [[ $# != "0" ]]; then
  if [[ $1 =~ "." ]]; then
    fe=${1##*.}
    bn=${1%.*}
  else
    fe=''
    bn=${1%.*}
  fi
  if [[ "$fe" != '' ]]; then
    fe=".$fe"
  fi
  if [[ -s $1 ]]; then
    if [ ! -s $bn.00$fe ]; then
      mv $1 $bn.00$fe
    else
      for i in {98..0}
        do
        if [ -s $bn.$(printf "%02d" $i)$fe ]; then
          let "j = i + 1"
          mv $bn.$(printf "%02d" $i)$fe $bn.$(printf "%02d" $j)$fe
        fi
      done
      mv $1 $bn.00$fe
    fi
  fi
fi


#!/bin/bash -l
#================================================================================
# trans.     ### calculates the rectilinear translation values for every     ###
#            ### frame trajectory to center the biostructure of interest     ###
# parameters
# $1         x coordinate of average box center, cannot be NaN
# $2         y coordinate of average box center, cannot be NaN
# $3         z coordinate of average box center, cannot be NaN
# $4         file name of biostructure center coordinates for each frame
#_______________________________________________________________________________

FR=`sed -n '/      0.000/{=;q}' ${4}`

tail -n +${FR} ${4} | awk "{px[NR]=\$2;py[NR]=\$3;pz[NR]=\$4} END {for
(i=1;i<=2001;i++) {print $1-px[i],$2-py[i],$3-pz[i]} }" > trans.xvg
```

```
#!/bin/bash -l
#================================================================================
# translate. ### translate one .gro file and append it to output .xtc file   ###
#            ###                                                             ###
#
# $1         index of .gro file to translate 0-?
#   #!NOTE - this script is executed by the -exec option of trjconv. See the
#            GROMACS documentation and methods file of this repository for
#            explanation.
#_______________________________________________________________________________

line=$(( $1+1 ))
trans=$(sed "$line!d" trans.xvg)

/scratch/scratch/jpbria01/software/gromacs-2019.2.install/bin/gmx_mpi trjconv -f
/local/scratch/temp${1}.gro -n index.ndx -o /local/scratch/tump${1}.xtc -pbc mol -
trans $trans 0<<< 0

rm /local/scratch/temp${1}.gro


#!/bin/bash -l
#================================================================================
# 3dchrg4A.awk ### calculate avg charge in bins of volume for a trajectory   ###
#              ###                                                            ###
# parameters
# $1         number of horizontal (x-y) slices to create along z axis (2-?)
#   options: the output of this algorithm can be used to calculate
#            non-uniform electric fields or to produce computational X-rays.
#            since the computational cost of calculating electric fields
#            scales as N6, and that algorithm must process all the slices
#            created here, balancing the volume resolution and computational
#            time of both steps is required subject to the requirements and
#            limitations of each system on which it is used. higher
#            resolutions can be produced here if the intention is to
#            produce computational X-rays since they in incur negligible
#            scaling
# $2         name of file containing charges of each atom in system
# $3         output filename containing bins of average charge for each
#            system volume element
#_______________________________________________________________________________

ARG1=${1:-'50'}
ARG2=${2:-'chrgs2.xvg'}
ARG3=${3:-'chrg4.xvg'}

if [ -s "${ARG3}" ]; then s ../jobs/roll. ${ARG3}; fi

declare -a boxdims
boxdims=($(cat ../boxdims.xvg))

awk     -v PREC="quad" \
        -v zd=${boxdims[2]} -v xd=${boxdims[0]} -v yd=${boxdims[1]} \
        -v pags=${ARG1} -v cfyl=${ARG2} \
    '
BEGIN   { frms=2001
          dlns=0; drcs=0
          dza=zd/pags
          cols=int(xd/dza)+1; rows=int(yd/dza)+1
          dxa=xd/cols; dya=yd/rows
          xs=int(2/dxa); ys=int(2/dya); zs=int(2/dza)
          iz=cols*rows; iy=cols
          ibxs=cols*rows*pags
          for ( i=1; i<=ibxs; i++ ) {
            chrg[i]=0;
        } }
```

```
BEGINFILES
        /^[^\n#@]/ \
        { if ( FILENAME != "energy.xvg" ) {
            split( $0, chrgs )
          }
          if ( FILENAME == "energy.xvg" ) {
            if ( drcs%10 == 0 ) {
              dlns++
              dims[dlns][""]
              split( $0, dims[dlns])
            }
            drcs++
        } }
ENDFILES

END     \
        { for ( irecs=1; irecs<=frms; irecs++ ) {
            getline < "../coord.xvg" > 0
            FIRST_POSN=match( $0, /[\n@#]/ )
              while ( FIRST_POSN == 1 ) {
                getline < "../coord.xvg" > 0
                FIRST_POSN=match( $0, /[\n@#]/ )
              }
            split( $0, posns )
            ipos=0
            dx=dims[irecs][2]/cols
            dy=dims[irecs][3]/rows
            dz=dims[irecs][4]/pags
            for ( i=2; i<=NF; i+=3 ) {
              idx0=(i-2)+2
              x=int((posns[idx0]  +2)/dx)-xs
              y=int((posns[idx0+1]+2)/dy)-ys
              z=int((posns[idx0+2]+2)/dz)-zs
              if ( x>cols-1) { x-=cols }
              if ( y>rows-1) { y-=rows }
              if ( z>pags-1) { z-=pags }
              if ( x<0     ) { x+=cols }
              if ( y<0     ) { y+=rows }
              if ( z<0     ) { z+=pags }
              ipos++
              chrg[z*iz+x+y*iy+1]+=chrgs[ipos]
            } }
            adjc=0
            if ( substr(cfyl,1,4) != "nden" ) {
              totc=0; ipos=0; bxls=0
              for ( i=0; i<pags; i++ ) {
                for ( j=0; j<rows; j++ ) {
                  for ( k=1; k<=cols; k++ ) {
                    ipos++
                    if ( chrg[ipos] != 0 ) {
                      totc+=chrg[ipos]
                      bxls++
              } } } }
              adjc=totc/bxls/frms
            }
            printf("\n\n")
            ipos=0; zero=0
            for ( i=0; i<pags; i++ ) {
              for ( j=0; j<rows; j++ ) {
                for ( k=1; k<=cols; k++ ) {
                  ipos++
                  if ( chrg[ipos] != 0 ) {
                    printf(" %9.6g",chrg[ipos]/frms-adjc) }
                  else {
                    printf(" %9.6g",zero)
                } }
                printf("\n")
```

```
            }
            printf("\n\n")
        } }
    '   ../../../${ARG2} energy.xvg \
        > ${ARG3}
```

#!NOTE: The following *AWK* script (*3dchrg4C.awk*) is one of ten (C,H,L,N,P,R,S,T,W,Y) that calculate charges for individual moieties, (i.e. molecular sub-species). Unlike *3dchrg4A.awk*, all ten require seven arguments (accented below in grn font) to determine the indices of the atoms of a sub-species. A Future improvement will determine this information directly from the structure of the simulation topology file.

These charge calculation routines [3dcharg4(A,C,H,L,N,P,R,S,T,W,Y)] also calculate the number density files for CompXRs. The only changes required to do so is pass them a file name in parameter #2 that contains and integer value of 1 for each atom in the system and specify a different output file name in parameter #3. I suggest replacing the characters “chrg” with the characters “dens”.

```
#!/bin/bash -l
#================================================================================
# 3dchrg4C.awk ### calculate avg charge in bins of volume for a moiety C is  ###
#              ### the chlorine ions                                          ###
# parameters
# $1         number of horizontal slices
# $2         file name, contains the charges for each atom in the .gro file
# $3         name of the output file
# $4         membrane channel to analyze
#   options  0 - top membrane, 1 - bottom membrane
# $5         number of extra water molecules in simulation
#   #! NOTE  there should have always been 11480 H2O molecules in each
#            simulation. Somewhere, somehow, another variant with 11484 H2O
#            molecules crept in. We must know which we're dealing with.
# $6         number of ions in alpha bath
# $7         number of ions in beta bath
#________________________________________________________________________________

ARG1=${1:-'50'}
ARG2=${2:-'chrgs2.xvg'}
ARG3=${3:-'chrg4C.xvg'}
ARG4=${4:-'0'}
ARG5=${5:-'0'}
ARG6=${6:-'11'}
ARG7=${7:-'11'}

if [ -s "${ARG3}" ]; then s ../jobs/roll. ${ARG3}; fi
```

```
declare -a boxdims
boxdims=($(cat ../boxdims.xvg))

awk     -v PREC="quad" \
        -v zd=${boxdims[2]} -v xd=${boxdims[0]} -v yd=${boxdims[1]} -v ch=${ARG4} \
        -v pags=${ARG1} -v cfyl=${ARG2} -v wa=${ARG5} -v io0=${ARG6} -v io1=${ARG7}
\
    '
BEGIN   { frms=2001
          dlns=0; drcs=0
          dza=zd/pags
          cols=int(xd/dza)+1; rows=int(yd/dza)+1
          dxa=xd/cols; dya=yd/rows
          xs=int(2/dxa); ys=int(2/dya); zs=int(2/dza)
          iz=cols*rows; iy=cols
          ibxs=cols*rows*pags
          for ( i=1; i<=ibxs; i++ ) {
            chrg[i]=0;
        } }

BEGINFILES
        /^[^\n#@]/ \
        { if ( FILENAME != "energy.xvg" ) {
            split( $0, chrgs )
          }
          if ( FILENAME == "energy.xvg" ) {
            if ( drcs%10 == 0 ) {
              dlns++
              dims[dlns][""]
              split( $0, dims[dlns])
            }
            drcs++
        } }
ENDFILES

END     \
        { fr0=386+22080+34440+wa*3+io0
          fr1=fr0*2+io1
          for ( irecs=1; irecs<=frms; irecs++ ) {
            getline < "../coord.xvg" > 0
            FIRST_POSN=match( $0, /[\n@#]/ )
              while ( FIRST_POSN == 1 ) {
                getline < "../coord.xvg" > 0
                FIRST_POSN=match( $0, /[\n@#]/ )
              }
            split( $0, posns )
            dx=dims[irecs][2]/cols
            dy=dims[irecs][3]/rows
            dz=dims[irecs][4]/pags
            k=fr0
            for ( i=1; i<=io0; i++ ) {
              idx0=((k+i)-1)*3+2
              x=int((posns[idx0]  +2)/dx)-xs
              y=int((posns[idx0+1]+2)/dy)-ys
              z=int((posns[idx0+2]+2)/dz)-zs
              if ( x>cols-1) { x-=cols }
              if ( y>rows-1) { y-=rows }
              if ( z>pags-1) { z-=pags }
              if ( x<0     ) { x+=cols }
              if ( y<0     ) { y+=rows }
              if ( z<0     ) { z+=pags }
              chrg[z*iz+x+y*iy+1]+=chrgs[k+i]
            }
            k=fr1
            for ( i=1; i<=io1; i++ ) {
              idx0=((k+i)-1)*3+2
              x=int((posns[idx0]  +2)/dx)-xs
```

```
            y=int((posns[idx0+1]+2)/dy)-ys
            z=int((posns[idx0+2]+2)/dz)-zs
            if ( x>cols-1) { x-=cols }
            if ( y>rows-1) { y-=rows }
            if ( z>pags-1) { z-=pags }
            if ( x<0     ) { x+=cols }
            if ( y<0     ) { y+=rows }
            if ( z<0     ) { z+=pags }
            chrg[z*iz+x+y*iy+1]+=chrgs[k+i]
          } }
          adjc=0
          if ( substr(cfyl,1,4) != "nden" ) {
           totc=0; ipos=0; bxls=0
            for ( i=0; i<pags; i++ ) {
              for ( j=0; j<rows; j++ ) {
                for ( k=1; k<=cols; k++ ) {
                 ipos++
                 if ( chrg[ipos] != 0 ) {
                   totc+=chrg[ipos]
                   bxls++
           } } } }
           adjc=totc/bxls/frms
          }
          printf("\n\n")
          ipos=0; zero=0
          for ( i=0; i<pags; i++ ) {
            for ( j=0; j<rows; j++ ) {
              for ( k=1; k<=cols; k++ ) {
                ipos++
                if ( chrg[ipos] != 0 ) {
                  printf(" %9.6g",chrg[ipos]/frms-adjc) }
                else {
                  printf(" %9.6g",zero)
                } }
              printf("\n")
            }
            printf("\n\n")
        } }
    '   ../../../${ARG2} energy.xvg \
        > ${ARG3}
```

#!NOTE: The following script lchrgall. can be used to conveniently execute all ten moiety charge calculation routines or any individual one with a simple command.

```
#!/bin/bash -l
#================================================================================
# lchrgall.  ### calculates charge files with parameters applicable to this  ###
#            ### particular simulation, parameters 2 & 4-7.                   ###
# parameters
# $1         number of vertical bins
#   options: typically, 50 or 100
#_______________________________________________________________________________

chr1=${2:-'50'} # how many pages?
chr2='chrgs3.xvg' # which sim version?
chr3=${2:0:1} # number for file character
chr4='1' # which pore?
chr5='4' # 11480 or 1148(4) waters?
chr6='10' # alpha bath ions
chr7='11' # beta bath ions
```

```
case ${1} in
  4A)
    s ../jobs/chrgall. ${chr1} 3dchrg4A.awk ${chr2}
  ;;
  4H)
    s ../jobs/chrgall. ${chr1} 3dchrg4H.awk ${chr2} chrg${chr3}H.xvg ${chr4} ${chr5}
${chr6} ${chr7}
  ;;
  4N)
    s ../jobs/chrgall. ${chr1} 3dchrg4N.awk ${chr2} chrg${chr3}N.xvg ${chr4} ${chr5}
${chr6} ${chr7}
  ;;
  4P)
    s ../jobs/chrgall. ${chr1} 3dchrg4P.awk ${chr2} chrg${chr3}P.xvg ${chr4} ${chr5}
${chr6} ${chr7}
  ;;
  4W)
    s ../jobs/chrgall. ${chr1} 3dchrg4W.awk ${chr2} chrg${chr3}W.xvg ${chr4} ${chr5}
${chr6} ${chr7}
  ;;
  4R)
    s ../jobs/chrgall. ${chr1} 3dchrg4R.awk ${chr2} chrg${chr3}R.xvg ${chr4} ${chr5}
${chr6} ${chr7}
  ;;
  4T)
    s ../jobs/chrgall. ${chr1} 3dchrg4T.awk ${chr2} chrg${chr3}T.xvg ${chr4} ${chr5}
${chr6} ${chr7}
  ;;
  4Y)
    s ../jobs/chrgall. ${chr1} 3dchrg4Y.awk ${chr2} chrg${chr3}Y.xvg ${chr4} ${chr5}
${chr6} ${chr7}
  ;;
  4S)
    s ../jobs/chrgall. ${chr1} 3dchrg4S.awk ${chr2} chrg${chr3}S.xvg ${chr4} ${chr5}
${chr6} ${chr7}
  ;;
  4C)
    s ../jobs/chrgall. ${chr1} 3dchrg4C.awk ${chr2} chrg${chr3}C.xvg ${chr4} ${chr5}
${chr6} ${chr7}
  ;;
  4L)
    s ../jobs/chrgall. ${chr1} 3dchrg4L.awk ${chr2} chrg${chr3}L.xvg ${chr4} ${chr5}
${chr6} ${chr7}
  ;;
  *)
    s ../jobs/chrgall. ${chr1} 3dchrg4A.awk ${chr2}
    s ../jobs/chrgall. ${chr1} 3dchrg4H.awk ${chr2} chrg${chr3}H.xvg ${chr4} ${chr5}
${chr6} ${chr7}
    s ../jobs/chrgall. ${chr1} 3dchrg4N.awk ${chr2} chrg${chr3}N.xvg ${chr4} ${chr5}
${chr6} ${chr7}
    s ../jobs/chrgall. ${chr1} 3dchrg4P.awk ${chr2} chrg${chr3}P.xvg ${chr4} ${chr5}
${chr6} ${chr7}
    s ../jobs/chrgall. ${chr1} 3dchrg4W.awk ${chr2} chrg${chr3}W.xvg ${chr4} ${chr5}
${chr6} ${chr7}
    s ../jobs/chrgall. ${chr1} 3dchrg4R.awk ${chr2} chrg${chr3}R.xvg ${chr4} ${chr5}
${chr6} ${chr7}
    s ../jobs/chrgall. ${chr1} 3dchrg4T.awk ${chr2} chrg${chr3}T.xvg ${chr4} ${chr5}
${chr6} ${chr7}
    s ../jobs/chrgall. ${chr1} 3dchrg4Y.awk ${chr2} chrg${chr3}Y.xvg ${chr4} ${chr5}
${chr6} ${chr7}
    s ../jobs/chrgall. ${chr1} 3dchrg4S.awk ${chr2} chrg${chr3}S.xvg ${chr4} ${chr5}
${chr6} ${chr7}
    s ../jobs/chrgall. ${chr1} 3dchrg4C.awk ${chr2} chrg${chr3}C.xvg ${chr4} ${chr5}
${chr6} ${chr7}
    s ../jobs/chrgall. ${chr1} 3dchrg4L.awk ${chr2} chrg${chr3}L.xvg ${chr4} ${chr5}
${chr6} ${chr7}
esac
```

```
#!/bin/bash -l
#================================================================================
# chrgall.   ### This script takes the parameters that come from lchrgall. to ###
#            ### calculate the charge files for the different simulation      ###
#            ### configurations that exist. A more capable method would       ###
#            ### require a script that can determine the parameters from the  ###
#            ### topol.top file, but which pore channel to examine (0 or 1)   ###
#            ### will remain a manual choice, possibly always since the       ###
#            ### computational cost is so high for even 1 pore. This more     ###
#            ### capable method has not been developed yet
# parameters
# $1         number of vertical bins
#   options: typically, 50 or 100
# $2         awk routine to execute
#   options: 3dchrg4[AHNPWSCRTYL].awk
# $3         atom charge file for this version of simulation topology
#   options: whatever you have named your charge file
# $4         output file name
# $5         pore channel number to process
#   options: 0 or 1
# $6         number of alpha bath ions of each species
#   explain: if alpha bath has 11 +tive and 11 -tive ions, (always neutral)
#            enter 11
# $7         number of beta bath ions of each species
#_______________________________________________________________________________

Chr1=${1:-'50'}
Chr2=${2:-'3dchrg4A.awk'}
Chr3=${3:-'chrgs2.xvg'}
Chr4=${4:-'test.xvg'}
Chr5=${5:-'0'}
Chr6=${6:-'0'}
Chr7=${7:-'11'}
Chr8=${8:-'11'}

s ../jobs/${Chr2} ${Chr1} ${Chr3} ${Chr4} ${Chr5} ${Chr6} ${Chr7} ${Chr8}


#!/bin/bash -l
#================================================================================
# Bfintegr8.awk ### performs brute force calculation of electric field at    ###
#               ### location of each bin in simulation from charge in each   ###
#               ### bin                                                      ###
# parameters
# $1    number of bins in z direction
# $2    charge file to integrate
# $3    output file name - outputs 2 sections, 1st each voxel contains the
#                          field on just that bin; 2nd is integrated fields,
#                          i.e. potential from pb to this voxel
#   #!NOTE - the computational cost of this script scales as N6 of
#            parameter $1. it gets very expensive very quick
#_______________________________________________________________________________

ARG1=${1:-'50'}

declare -a boxdims
boxdims=($(cat ../boxdims.xvg))

awk     -v PREC="quad" \
        -v xd=${boxdims[0]} -v yd=${boxdims[1]} -v zd=${boxdims[2]} \
        -v pags=${ARG1} \
    '
BEGIN   { lns=0; dza=zd/pags;
          cols=int(xd/dza)+1; rows=int(yd/dza)+1
          dxa=xd/cols; dya=yd/rows
          divf=cols*rows*pags-1; ibxs=divf+1
          hx=(cols-1)/2; hy=(rows-1)/2; hz=(pags-1)/2
```

```
        }

BEGINFILES
        /^ / \
        { lns++
          vals[lns][""]
          split( $0, vals[lns])
        }
ENDFILES

END     \
        { FACT=18.0951281698764944
          for ( l=1; l<=pags; l++ ) {
            for ( m=1; m<=rows; m++ ) {
              indx=(l-1)*rows+m
              ft[indx][""]
              for ( n=1; n<=cols; n++ ) {
                lf=(n-1)*4+1
                for ( i=1; i<=pags; i++ ) {
                  for ( j=1; j<=rows; j++ ) {
                    for ( k=1; k<=cols; k++ ) {
                      lndx=(i-1)*rows+j
                      dstx=(n-k)*dxa; dsty=(m-j)*dya; dstz=(l-i)*dza
                      if ( k-n > hx || k-n < -hx ) {
                        if ( k < n ) {
                          dstx-=xd }
                        else {
                          dstx+=xd
                      } }
                      if ( j-m > hy || j-m < -hy ) {
                        if ( j < m ) {
                          dsty-=yd }
                        else {
                          dsty+=yd
                      } }
                      if ( i-l > hz || i-l < -hz ) {
                        if ( i < l ) {
                          dstz-=zd }
                        else {
                          dstz+=zd
                      } }
                      dist2=dstx^2+dsty^2+dstz^2
                      if ( dist2 != 0 ) {
                        qxk=vals[lndx][k]*FACT
                        ftot=qxk/dist2
                        dist=sqrt(dist2)
                        if ( ftot != 0 ) {
                          ft[indx][lf+1]+=ftot*dstx/dist
                          ft[indx][lf+2]+=ftot*dsty/dist
                          ft[indx][lf+3]+=ftot*dstz/dist
                } } } } }
                ft[indx][lf+1]/=divf
                tx+=ft[indx][lf+1]
                ft[indx][lf+2]/=divf
                ty+=ft[indx][lf+2]
                ft[indx][lf+3]/=divf
                tz+=ft[indx][lf+3]
            } } }
            adjx=-tx/ibxs
            adjy=-ty/ibxs
            adjz=-tz/ibxs

            printf("\n\n")
            for ( i=1; i<=pags; i++ ) {
              for ( j=1; j<=rows; j++ ) {
                indx=(i-1)*rows+j
                for ( k=1; k<=cols; k++ ) {
```

```
                lf=(k-1)*4+1
                ft[indx][lf+1]+=adjx
                ft[indx][lf+2]+=adjy
                ft[indx][lf+3]+=adjz

ft[indx][lf]=sqrt((ft[indx][lf+1])^2+(ft[indx][lf+2])^2+(ft[indx][lf+3])^2)
                printf(" %9.6g %9.6g %9.6g %9.6g %9.6g %9.6g
%9.6g\n",k*dxa,j*dya,i*dza,ft[indx][lf],ft[indx][lf+1],ft[indx][lf+2],ft[indx][lf+3]
)
            } }
            print("\n\n")
          }

          tx=0; ty=0; tz=0
          for ( l=1; l<=pags; l++ ) {
            for ( m=1; m<=rows; m++ ) {
              indx=(l-1)*rows+m
              for ( n=1; n<=cols; n++ ) {
                lf=(n-1)*4+1
                if ( m == 1 ) {
                  ft[indx][lf+1]+=ft[indx][lf-3]
                  tx+=ft[indx][lf+1]
                  ft[indx][lf+2]+=ft[indx+rows-1][lf+2]
                  ty+=ft[indx][lf+2]
                  ft[indx][lf+3]+=ft[indx-rows][lf+3]
                  tz+=ft[indx][lf+3] }
                else {
                  if ( m == rows ) {
                    ft[indx][lf+1]+=ft[indx][lf-3]
                    tx+=ft[indx][lf+1]
                    ft[indx][lf+2]+=ft[indx-rows+1][lf+2]
                    ty+=ft[indx][lf+2]
                    ft[indx][lf+3]+=ft[indx-rows][lf+3]
                    tz+=ft[indx][lf+3] }
                  else {
                    ft[indx][lf+1]+=ft[indx][lf-3]
                    tx+=ft[indx][lf+1]
                    ft[indx][lf+2]+=ft[indx-1][lf+2]
                    ty+=ft[indx][lf+2]
                    ft[indx][lf+3]+=ft[indx-rows][lf+3]
                    tz+=ft[indx][lf+3]
          } } } } }
          adjx=-tx/ibxs
          adjy=-ty/ibxs
          adjz=-tz/ibxs

          printf("\n\n")
          for ( i=1; i<=pags; i++ ) {
            for ( j=1; j<=rows; j++ ) {
              indx=(i-1)*rows+j
              for ( k=1; k<=cols; k++ ) {
                lf=(k-1)*4+1
                ft[indx][lf+1]+=adjx
                ft[indx][lf+2]+=adjy
                ft[indx][lf+3]+=adjz

ft[indx][lf]=sqrt((ft[indx][lf+1])^2+(ft[indx][lf+2])^2+(ft[indx][lf+3])^2)
                printf(" %9.6g %9.6g %9.6g %9.6g %9.6g %9.6g
%9.6g\n",k*dxa,j*dya,i*dza,ft[indx][lf],ft[indx][lf+1],ft[indx][lf+2],ft[indx][lf+3]
)
            } }
            printf("\n\n")
        } }
    '   ${2} > ${3}
```

#!NOTE: The following script lchrgall. can be used to conveniently execute all 11 field calculation routines or any individual one with a simple command.

```
#!/bin/bash -l
#================================================================================
# integr8all.  ### calculate all the electric fields                          ###
#              ###                                                            ###
# parameters
# $1         Moiety to process
#   options: <null> - all
#            4A,4H,4N,4P,4W,4S,4C,4R,4T,4L,4Y
# $2         number of bins in the z direction
#   options: 2-?
#_______________________________________________________________________________

ARG2=${2:-'50'}

case ${1} in
  4A)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}A.xvg fylr${ARG2:0:1}A.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  4H)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}H.xvg fylr${ARG2:0:1}H.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  4N)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}N.xvg fylr${ARG2:0:1}N.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  4P)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}P.xvg fylr${ARG2:0:1}P.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  4W)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}W.xvg fylr${ARG2:0:1}W.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  4S)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}S.xvg fylr${ARG2:0:1}S.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  4C)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}C.xvg fylr${ARG2:0:1}C.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  4R)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}R.xvg fylr${ARG2:0:1}R.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  4T)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}T.xvg fylr${ARG2:0:1}T.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  4L)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}L.xvg fylr${ARG2:0:1}L.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  4A)
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}Y.xvg fylr${ARG2:0:1}Y.xvg
    s ../jobs/spltfld. ${ARG2:0:1} ${ARG1:0:1}
  ;;
  *)
```

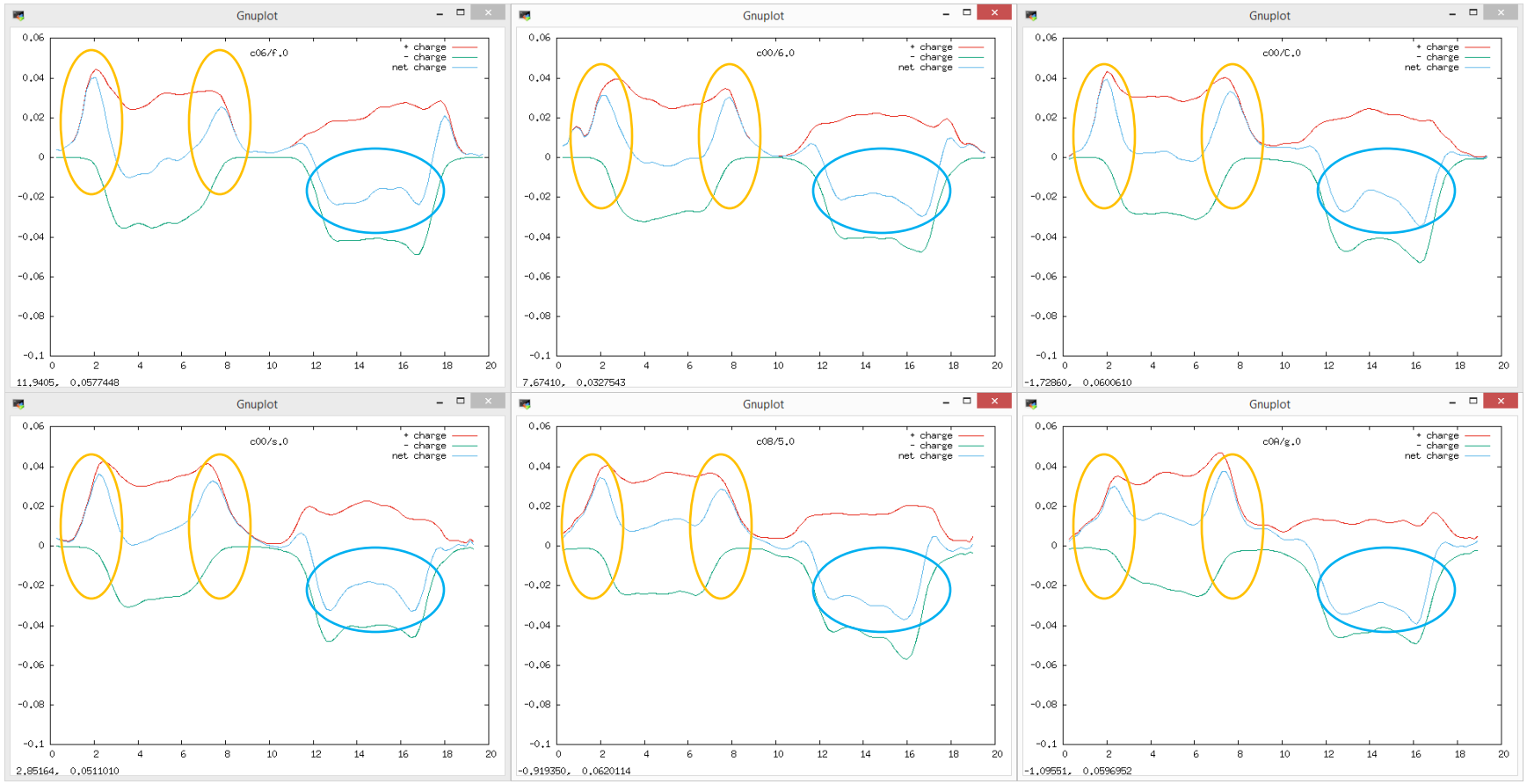
```
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}A.xvg fylr${ARG2:0:1}A.xvg
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}H.xvg fylr${ARG2:0:1}H.xvg
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}N.xvg fylr${ARG2:0:1}N.xvg
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}P.xvg fylr${ARG2:0:1}P.xvg
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}W.xvg fylr${ARG2:0:1}W.xvg
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}S.xvg fylr${ARG2:0:1}S.xvg
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}C.xvg fylr${ARG2:0:1}C.xvg
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}R.xvg fylr${ARG2:0:1}R.xvg
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}T.xvg fylr${ARG2:0:1}T.xvg
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}L.xvg fylr${ARG2:0:1}L.xvg
    s ../jobs/Bfintegr8.awk ${ARG2} chrg${ARG2:0:1}Y.xvg fylr${ARG2:0:1}Y.xvg
    for s in A H N P W S C R T L Y; do
      s ../jobs/spltfld. ${ARG2:0:1} ${s}
    done
  ;;
esac


#!/bin/bash -l
#================================================================================
# spltfld.   ### splits electric field output file from Bfintegr8.awk into   ###
#            ### local and vector summed sections                            ###
# parameters
# $1         vertical bins digit
#   options: typically, 5 (50) or 1 (100). Remember Bfintegr8.awk computational
#            cost scales as N^6. For the lipid systems (~113000 atoms), ~106
#            takes about 2 days to calculate. For now, that's my top end
#            practical capability.
# $2         Moiety Character
#   options: A,H,N,P,W,S,C,R,T,Y,L
#_______________________________________________________________________________

if ! [ $#0 <> 2 ]; then
  echo "Required vertical bins digit and moiety character not entered"
else
  head -n 22201 fylr${1}${2}.xvg > fyld${1}${2}.xvg
  tail -n 22152 fylr${1}${2}.xvg > fylt${1}${2}.xvg
fi


#!/usr/bin/gnuplot
#================================================================================
# plfldmaps  ### plots the electric field and potential maps                  ###
#            ###                                                            ###
# parameters
# none       N/A
#   options: N/A
#_______________________________________________________________________________

#set term pngcairo size 640,1160
set term pngcairo size 432,864 font ",12" # 4.5x9 at 96dpi
set xlabel "X-Axis Coordinate (nm)" offset 3,-1.5
set ylabel "Y-A (nm)" rotate parallel offset -12,0.5
set zlabel "Z-Axis Coordinate (nm)" rotate parallel offset 3.5
set xrange [0:8]
set yrange [0:8]
set zrange [0:20]
set xyplane 0
set xtics 0,1 offset 0,-0.5
set ytics 0,1 out
set ztics 0,2 offset 1
set view 87,10,1.4,1

set palette model RGB
#set palette defined (0 1 1 1, 2.00 0 0 0, 3.33 0 0 1, 4.66 0.2 0.8 0.8, 6 0.1 0.8
0.1, 7.33 0.8 0.8 0.2, 8.66 0.8 .2 0.8, 10 0.8 0.1 0.1)
```

```
#set palette defined (0 1 1 1, 2.00 0 0 0, 3.33 0 0 1, 4.66 0.4 1 1, 6 0.2 1 0.2,
7.33 1 1 0.4, 8.66 1 0.4 1, 10 1 0.2 0.2)
#set palette defined (0 1 1 1, 0.83 0 0 0, 1.66 0.58 0 0.83, 2.5 0 0.61 0.44, 3.33
0.33 0.70 0.91, 4.16 0.9 0.61 0, 5 0.95 0.91 0.30, 5.83 0 0.4 0.7, 6.66 0.90 0.13
0.07, 7.5 0 0 0, 8.33 0.58 0 0.83, 9.16 0 0.61 0.44, 10 0.33 0.70 0.91)
set palette defined (0 1 1 1, 0.83 0 0 0, 1.66 0.58 0 0.83, 2.5 0 0.61 0.44, 3.33
0.33 0.70 0.91, 4.16 0.765 0.384 0, 5 0.95 0.91 0.30, 5.83 0 0 1, 6.66 1 0 0, 7.5 0
0 0, 8.33 0.75 0 1, 9.16 0 1 0, 10 1 0.666 0)
set cbrange [0:13]
set colorbox user o .93,.25 size .038,.472

array
fy[22]=["fyld5A","fylt5A","fyld5H","fylt5H","fyld5N","fylt5N","fyld5P","fylt5P","fyl
d5W","fylt5W","fyld5S","fylt5S","fyld5C","fylt5C","fyld5R","fylt5R","fyld5T","fylt5T
","fyld5Y","fylt5Y","fyld5L","fylt5L"]
array
fn[22]=["fyld5A.xvg","fylt5A.xvg","fyld5H.xvg","fylt5H.xvg","fyld5N.xvg","fylt5N.xvg
","fyld5P.xvg","fylt5P.xvg","fyld5W.xvg","fylt5W.xvg","fyld5S.xvg","fylt5S.xvg","fyl
d5C.xvg","fylt5C.xvg","fyld5R.xvg","fylt5R.xvg","fyld5T.xvg","fylt5T.xvg","fyld5Y.xv
g","fylt5Y.xvg","fyld5L.xvg","fylt5L.xvg"]
#array
a[22]=[777,730,110,115,198,978,1071,551,551,872,191,307,203,18,25,66,226,187,158,160
,240,56]
array
a[22]=[1000,800,125,125,225,1250,1250,700,700,1100,250,307,203,18,25,66,226,187,158,
160,240,56]
#array
a[22]=[140,140,140,140,140,140,140,140,140,140,140,22,22,22,22,22,22,22,22,22,22,22]
array ti[11]=["Net Electric Flux","Ester
Groups","Choline","Phosphate","Water","Sodium","Chlorine","SCMTR Headgroup","SCMTR
Tails","Lipid Tails","Lipids"]
array type[2]=["Field Map","Potential Plot"]
array lynw[2]=[0.75,1]

do for [ofst=1:1] {
  do for [i=1:11] {

    fl="c08-9-R50-".fy[(i-1)*2+ofst]
    set output fl.".png"
#    set label 2 at screen 0.7,0.96 fl

    splot for [j=0:49] fy[(i-1)*2+ofst].".xvg" index j every 1:1:0:0:440:0 u
1:2:($3+.2845):($5*a[i+(ofst-1)*11]):($6*a[i+(ofst-1)*11]):($7*a[i+(ofst-
1)*11]):($4*a[i+(ofst-1)*11]*23) with vectors not lw lynw[ofst] lc palette

    set label 1 at screen 0.15,0.96 ti[i]." ".type[ofst]
    pause 1
    unset output
} }


#!/bin/bash -l
#================================================================================
# chrg2pdb.awk ### converts number density and charge files to .pdb file for ###
#              ### display in vmd                                              ###
# parameters
# $1         N/A
#   options: N/A
#________________________________________________________________________________

ARG1=${1:-'50'}
ARG2=${2:-'dens2H.xvg'}
ARG3=${3:-'chrg2H.xvg'}
ARG4=${4:-'chrg2H.pdb'}

if [ -s ${ARG4} ]; then s ../jobs/roll. ${ARG4}; fi
```

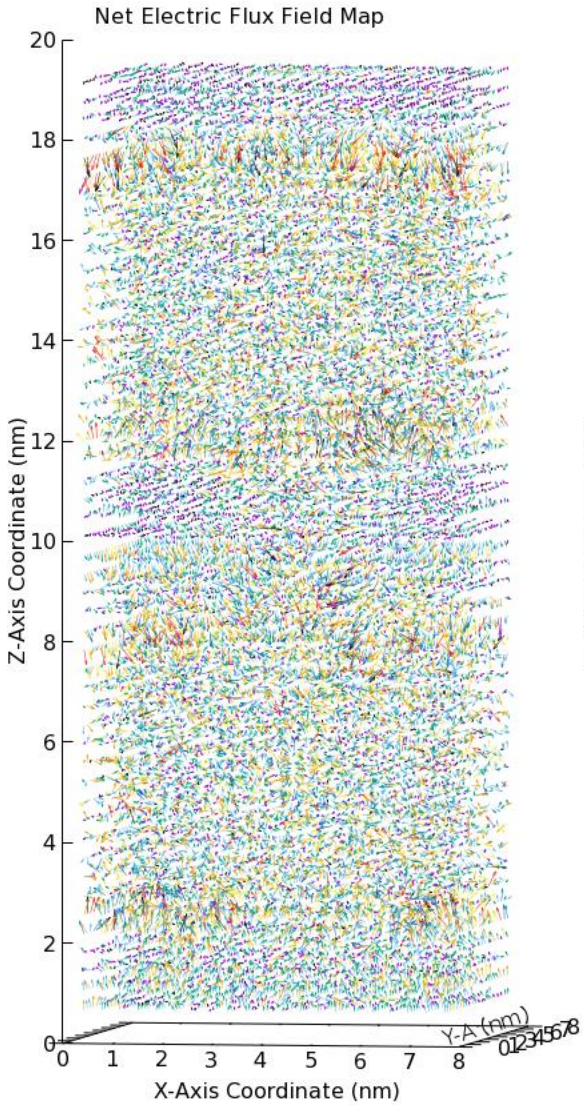
```
declare -a boxdims
boxdims=($(cat ../boxdims.xvg))

gawk    -v PREC="quad" \
        -v xd=${boxdims[0]} -v yd=${boxdims[1]} -v zd=${boxdims[2]} \
        -v pags=${ARG1} -v dfyl=${ARG2} -v cfyl=${ARG3} \
    '
BEGIN   { dlns=0; clns=0
          dza=zd/pags
          cols=int(xd/dza)+1; rows=int(yd/dza)+1
          dxa=xd/cols; dya=yd/rows
          mult=pags/50; dmlt=mult^3
# on the line below, originally dxa*10*mult, etc. to convert nm to ångströms
# I discovered I could cut the dimensions in half when exporting to pdb at higher
# resolutions without VMD thinking the data points were bonded. This is the reason
for
# the /2 additions.
          dx=dxa*10/2*mult; dy=dya*10/2*mult; dz=zd/pags*10/2*mult
          iy=cols
          ibxs=cols*rows*pags
          Volm=dxa*dya*dza
        }

BEGINFILES
        /^[^\n#@]/ \
        { if ( FILENAME == dfyl ) {
            dlns++
            dens[dlns][""]
            split( $0, dens[dlns])
          }
          if ( FILENAME == cfyl ) {
            clns++
            chgs[clns][""]
            split( $0, chgs[clns])
        } }
ENDFILES

END     \
        { for ( i=1; i<=cols; i++ ) {
            x[i]=dx*(i-1)
          }
          for ( i=1; i<=rows; i++ ) {
            y[i]=dy*(i-1)
          }
          for ( i=1; i<=pags; i++ ) {
            z[i]=dz*(i-1)
          }
          printf("HEAD_1%9.3f%9.3f%9.3f  90.00  90.00  90.00 P 1
1\n",xd*10,yd*10,zd*10)
          printf("REMARK 001 %7d bins, %3d cols, %3d rows, %3d
pags\n",ibxs,cols,rows,pags)
          printf("REMARK 002 %9.6g nm^3/bin\n",volm)
          ipos=1; irsn=1;
          for ( k=0; k<pags*rows; k++ ) {
            for ( i=1; i<=cols; i++ ) {
              if ( dens[k+1][i] != 0 ) {
                printf("HETATM%5d  C2  HED X%4d    %8.3f%8.3f%8.3f%6.2f%6.2f C
0\n",ipos,irsn,x[i],y[k%iy+1],z[int(k/iy)+1],dens[k+1][i]*dmlt,chgs[k+1][i]*1000)
                ipos++;
                if ( ipos == 100000 ) {
                  ipos=1; irsn++
          } } } }
          printf("END\n")
        }
    '   ${ARG2} ${ARG3} \
        > ${ARG4}
```

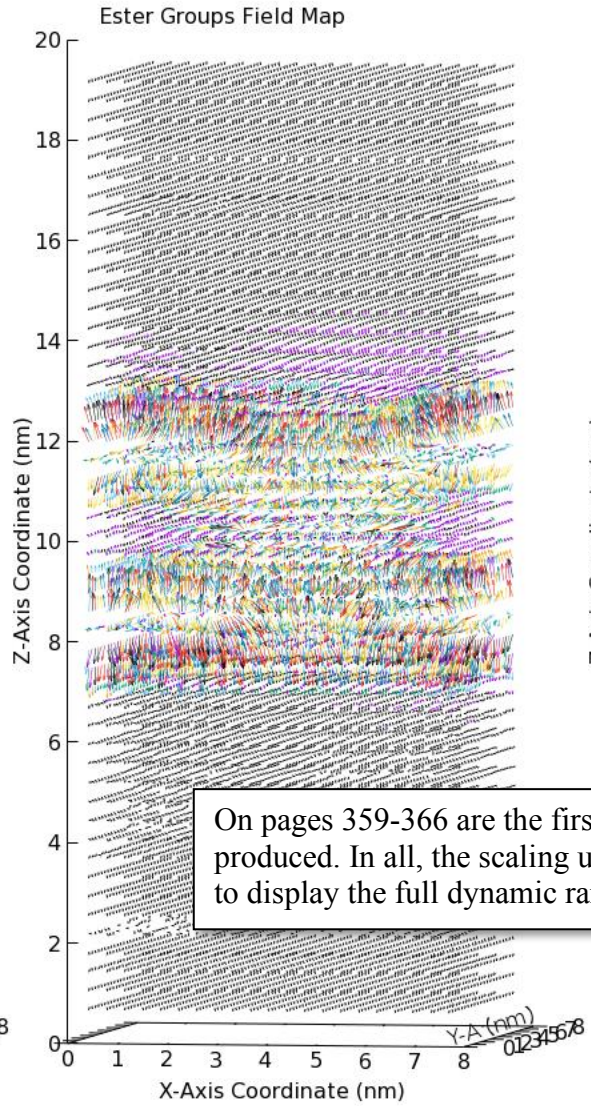
VISUALIZATION IMAGES

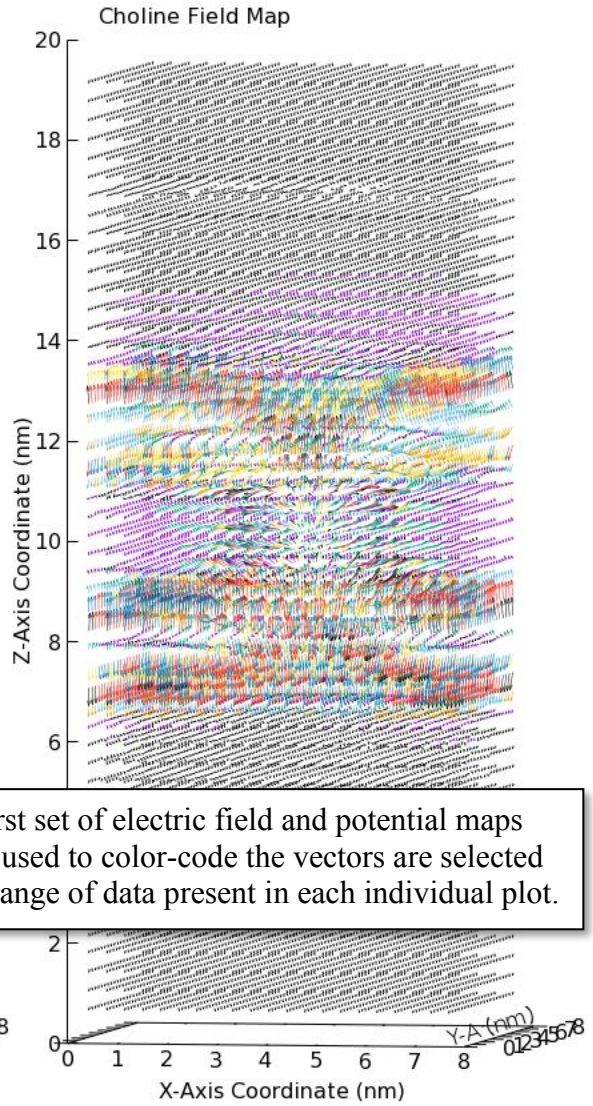
The following visualizations were produced as part of the basic research to discern as much as possible regarding coulombic distributions and fields in the membrane systems. The significance of each or each type is explained in the captions or preceding text.



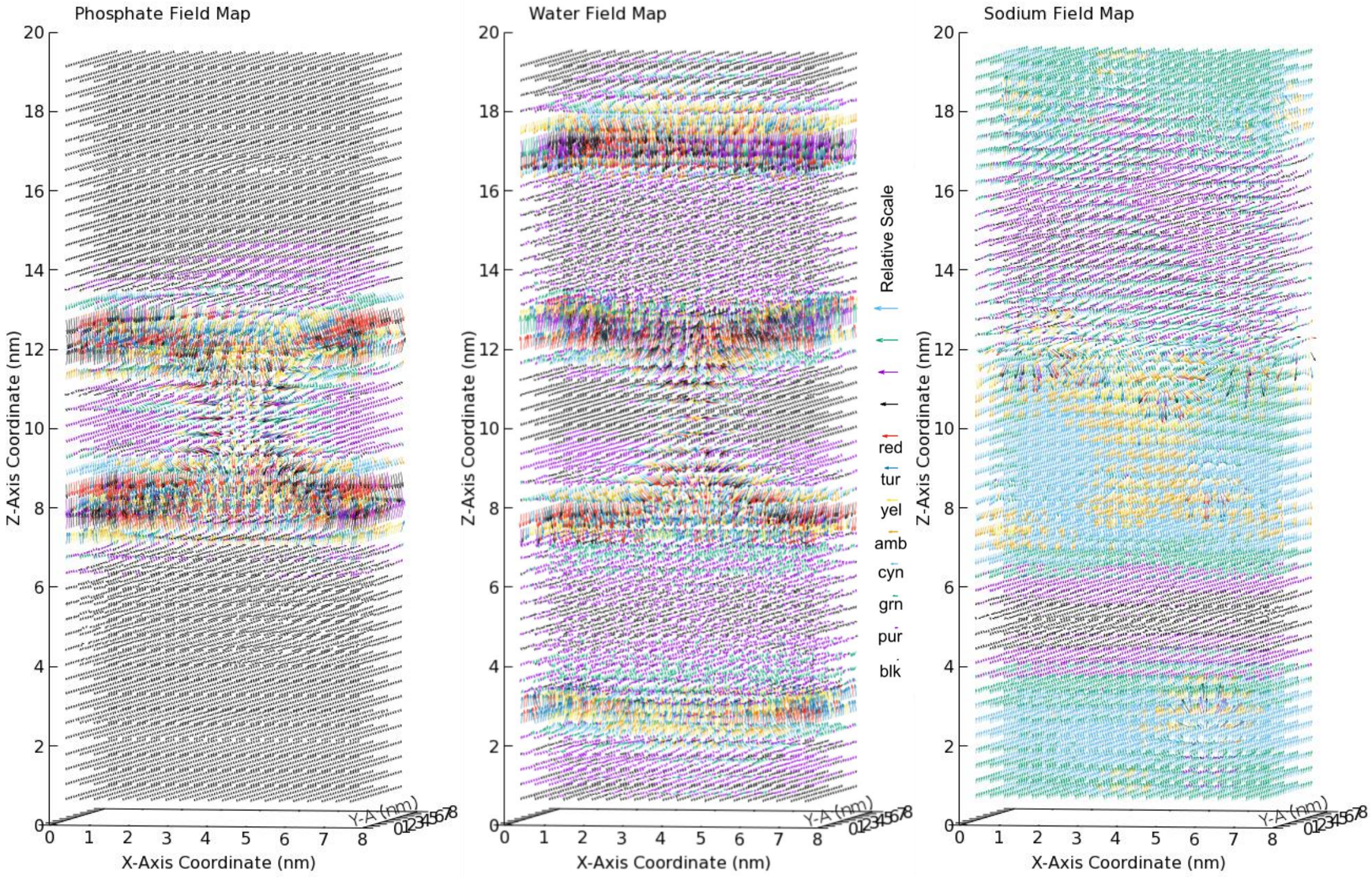
Figs. 10 and 11 revealed much larger field potentials in the membrane than the conventional one-dimensional potential map would hint at. This initiated other approaches to understand the overall picture of the electrostatic interactions. These plots show the charge distributions in the membrane systems differentiated by polarity, positive (red), negative (green) and net (cyan) for charge imbalances +/-4 to +/- 9. In the above plots, notice in the overall charge distributions the tendency for positive charges (mostly cations in this case) to aggregate next to the membranes (~@ 2 and 8 nm) while the negative charges (mostly anions) are much more evenly distributed throughout the bath region. The potassium cation (also important physiologically) begs similar analysis.

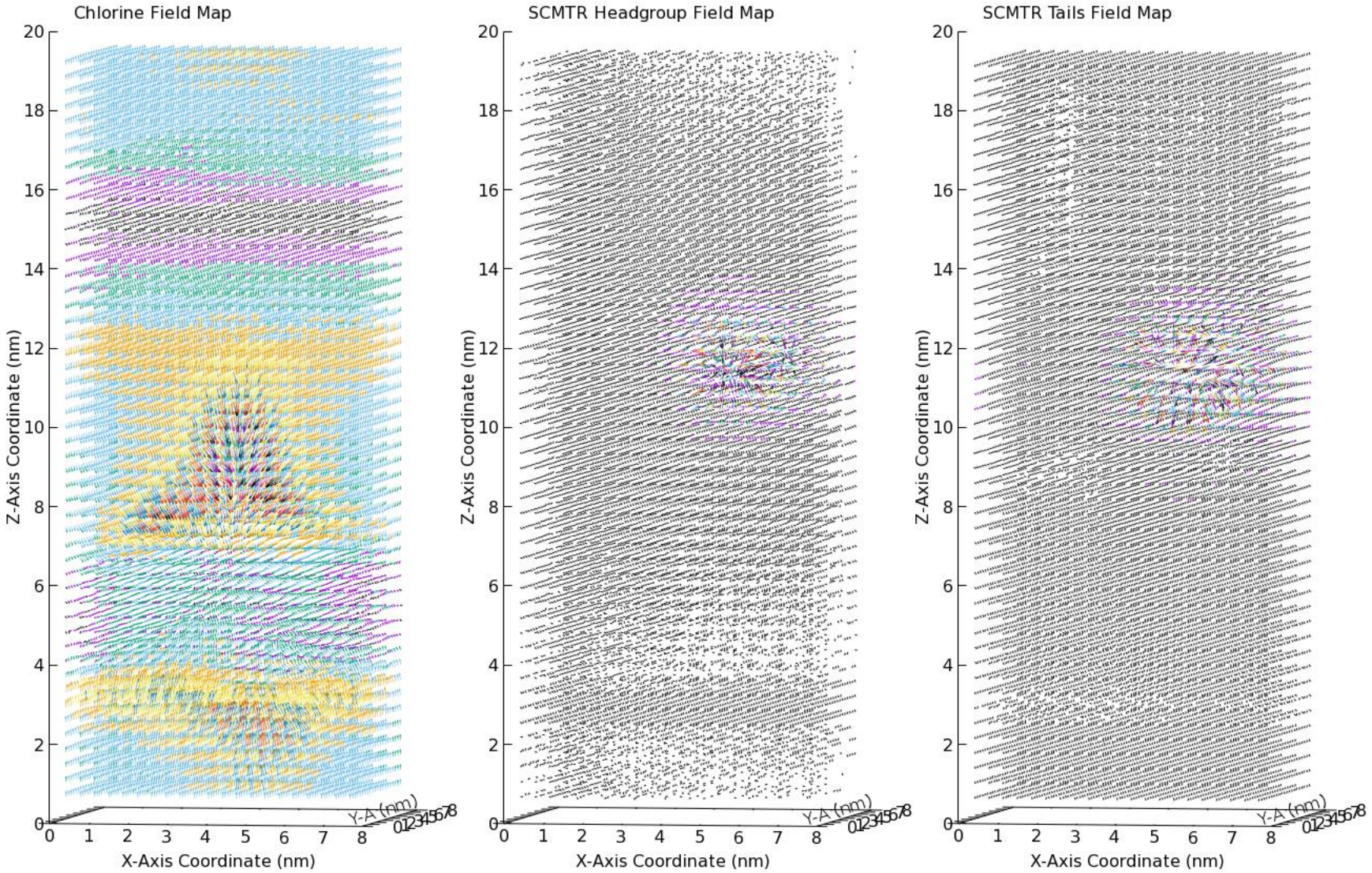


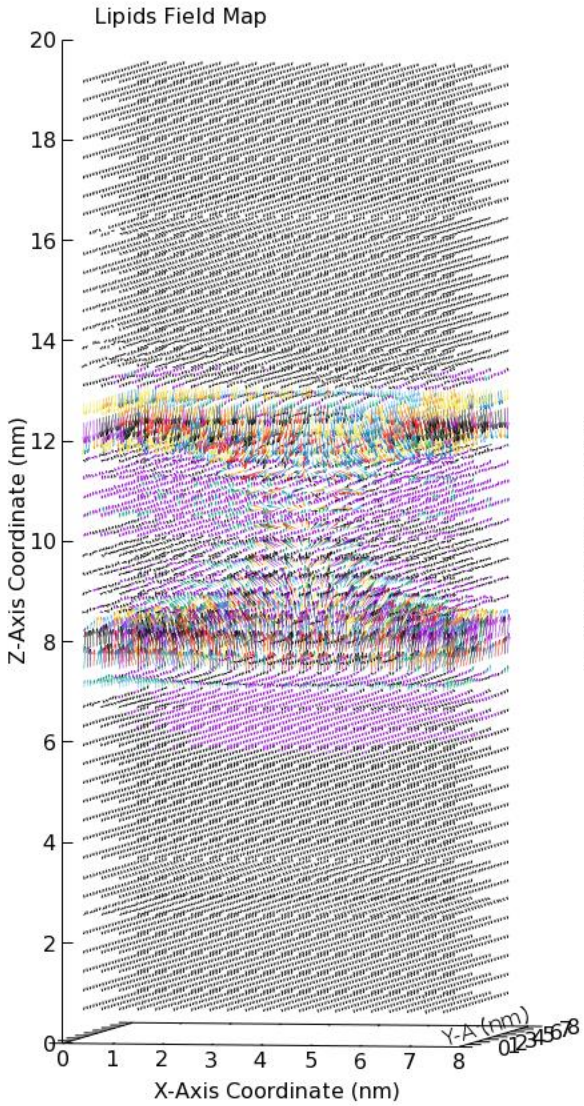


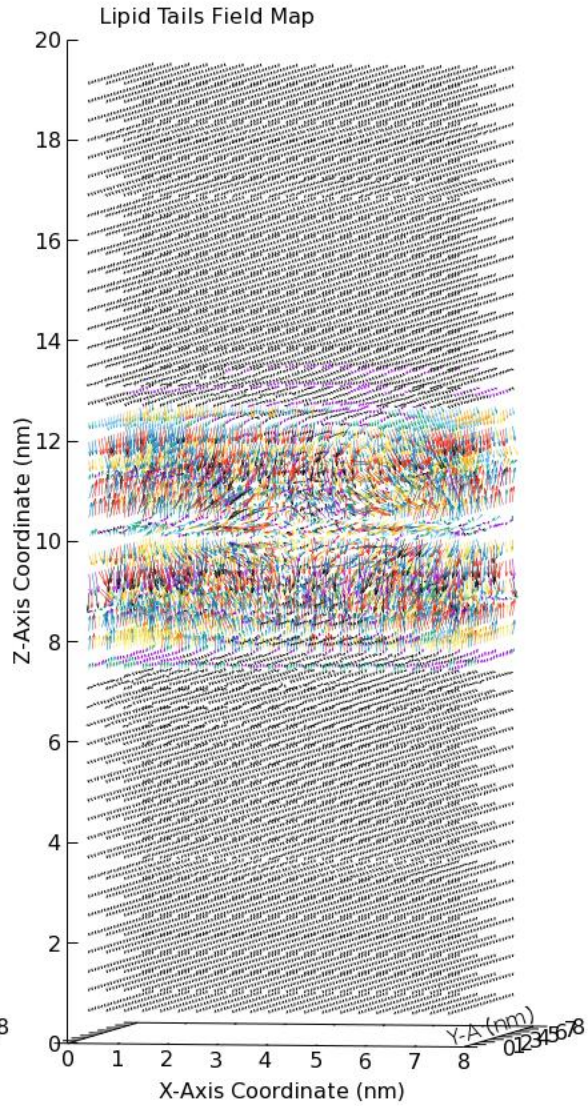


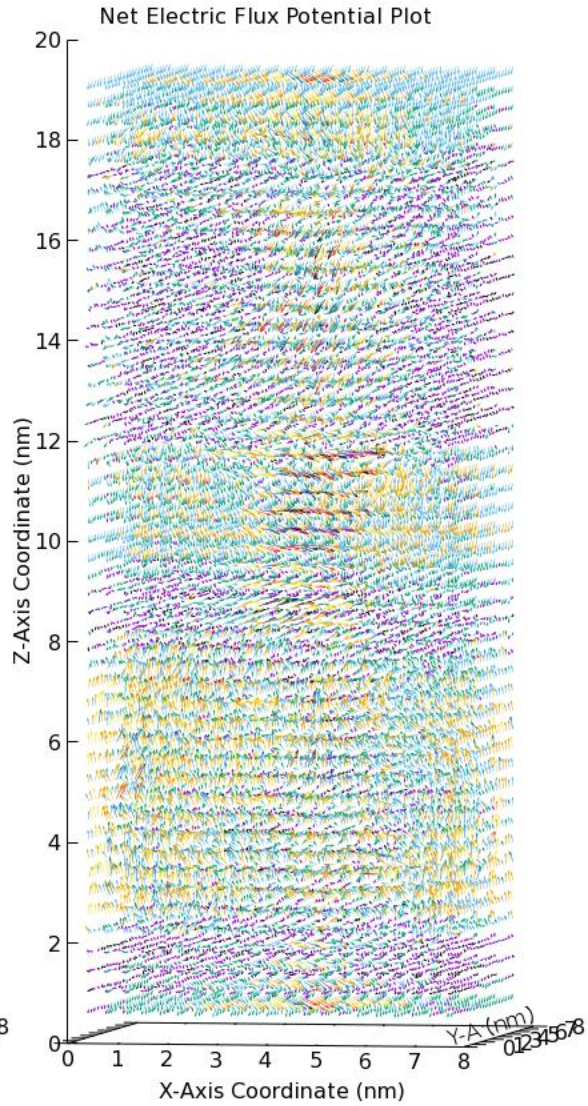
On pages 359-366 are the first set of electric field and potential maps produced. In all, the scaling used to color-code the vectors are selected to display the full dynamic range of data present in each individual plot.

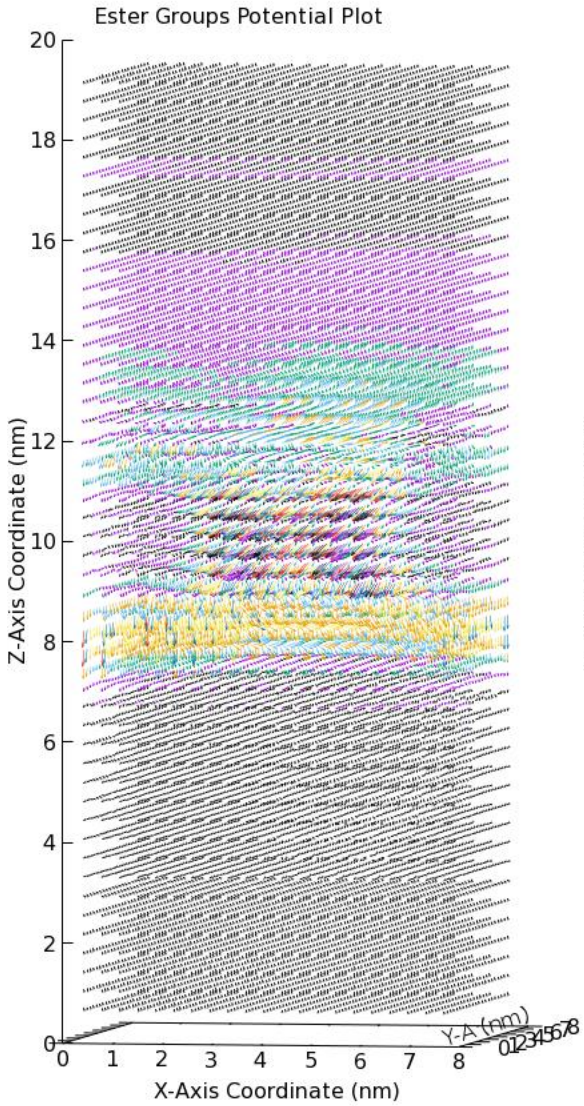

Phosphate Field Map
Water Field Map
Sodium Field Map
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)
Relative Scale
red
tur
yel
amb
cyn
grn
pur
blk

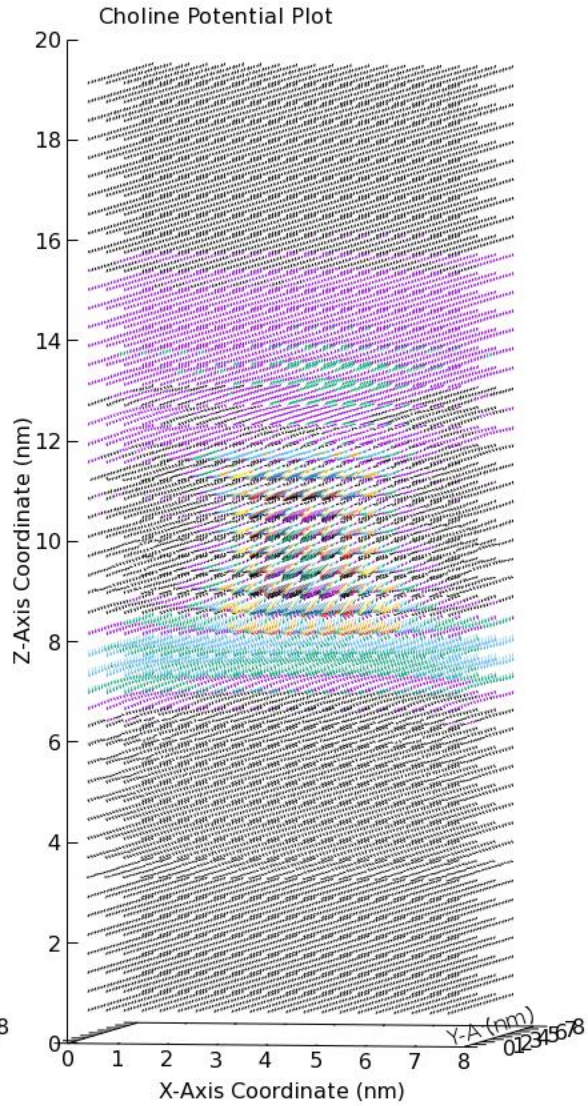

Chlorine Field Map
SCMTR Headgroup Field Map
SCMTR Tails Field Map
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)
0
2
4
6
8
10
12
14
16
18
20
0
1
2
3
4
5
6
7
8

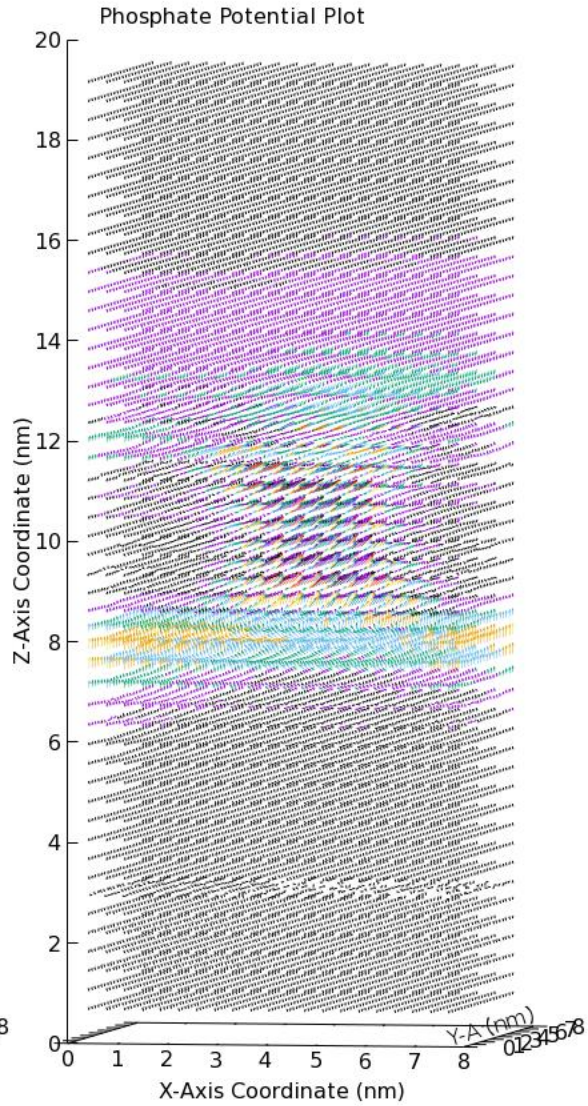

Lipids Field Map
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)

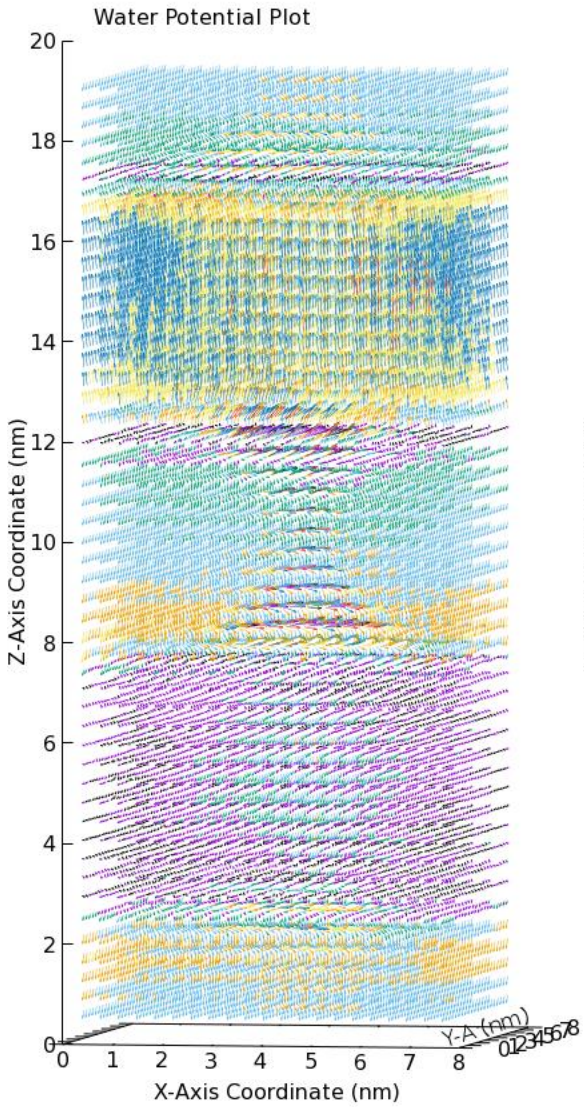

Lipid Tails Field Map
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)

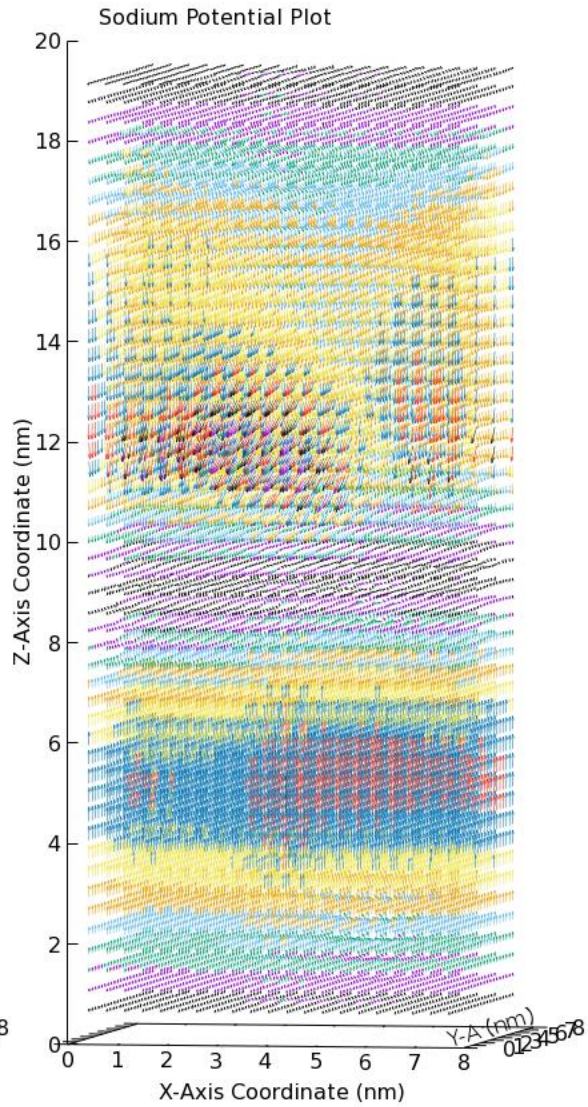

Net Electric Flux Potential Plot
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)

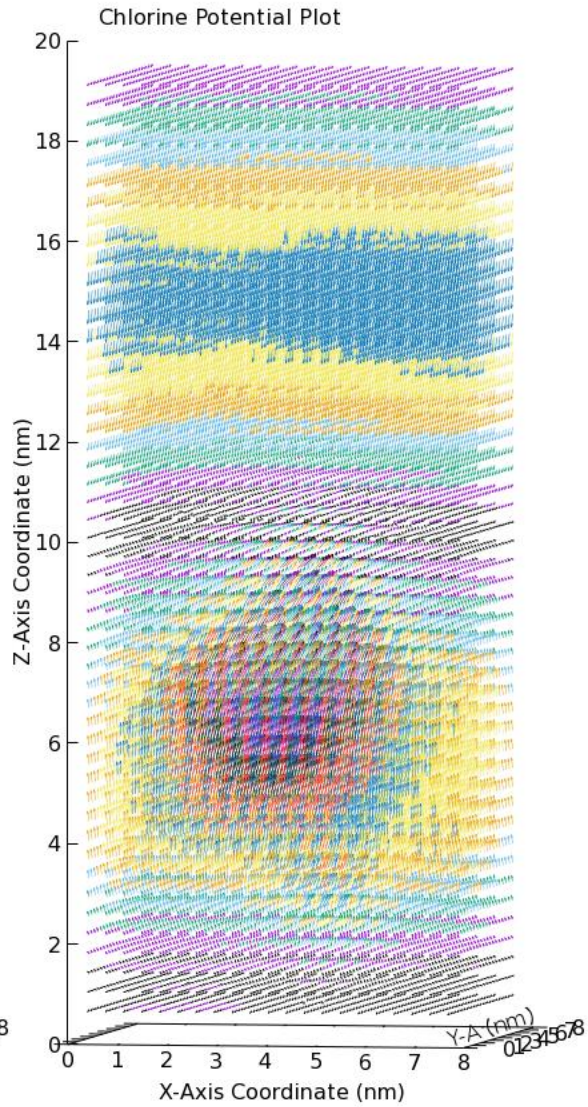

Ester Groups Potential Plot
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)

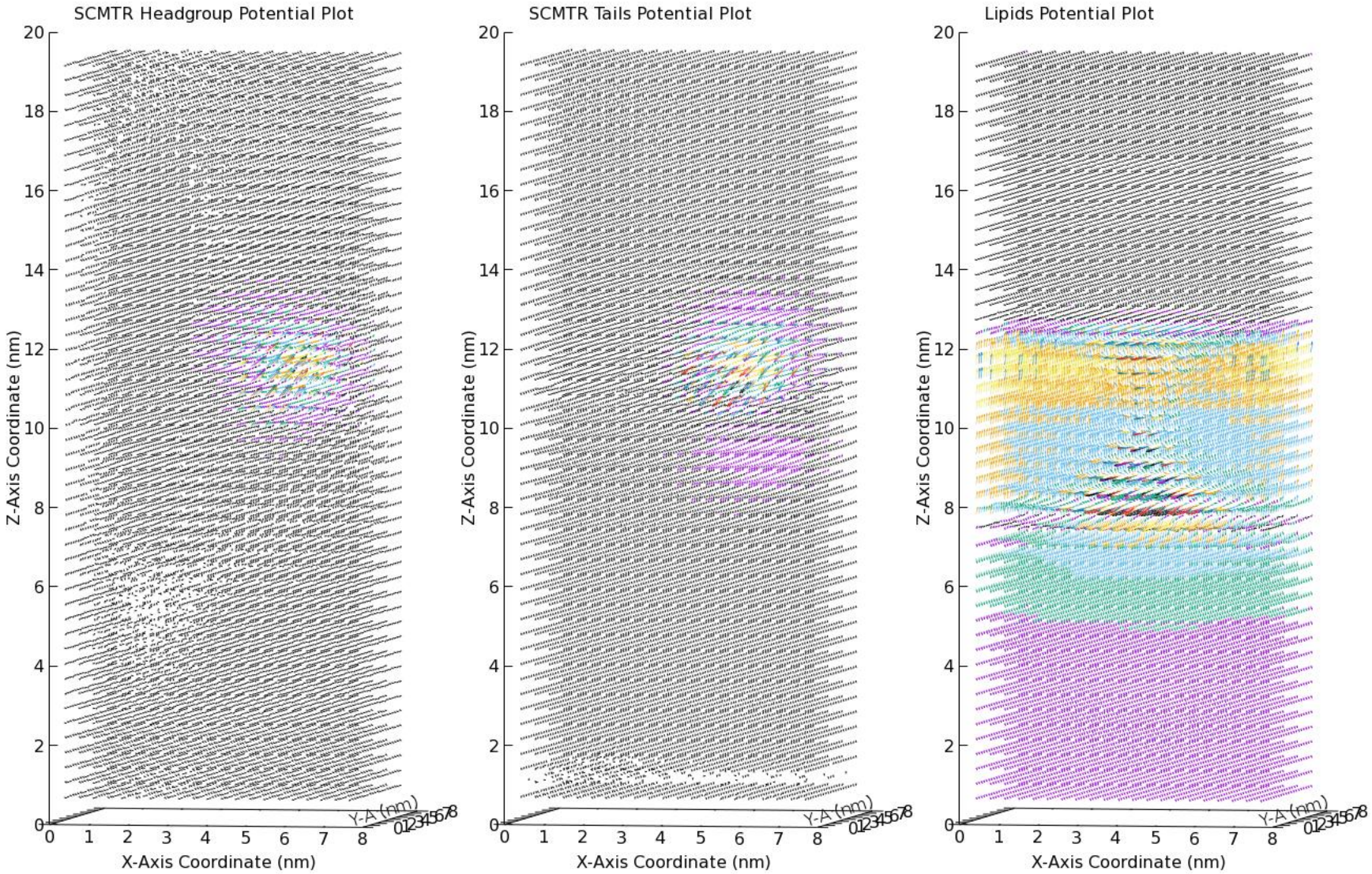

Choline Potential Plot
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)

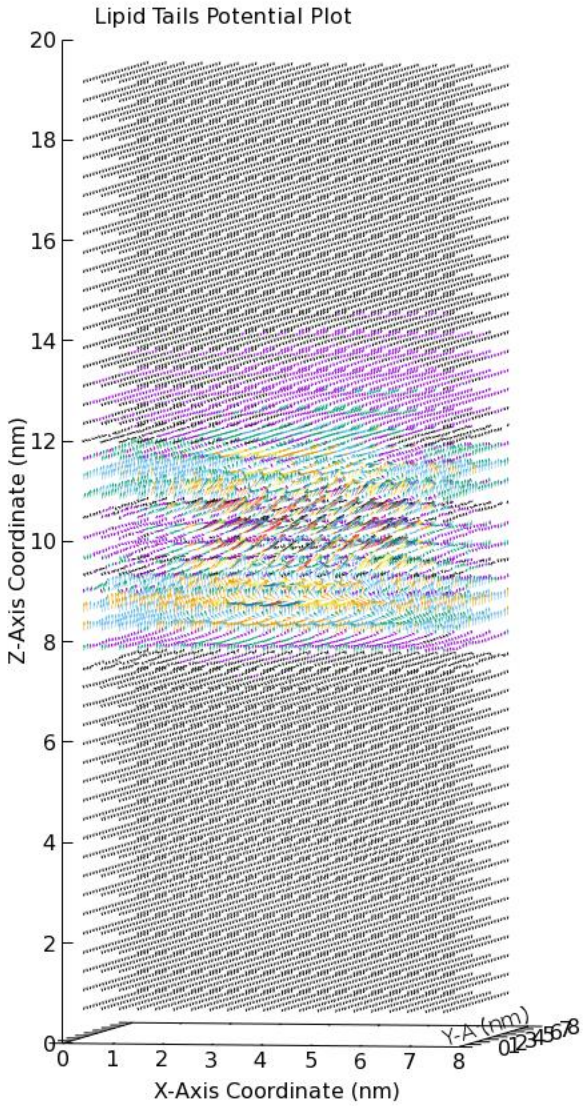

Phosphate Potential Plot
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)

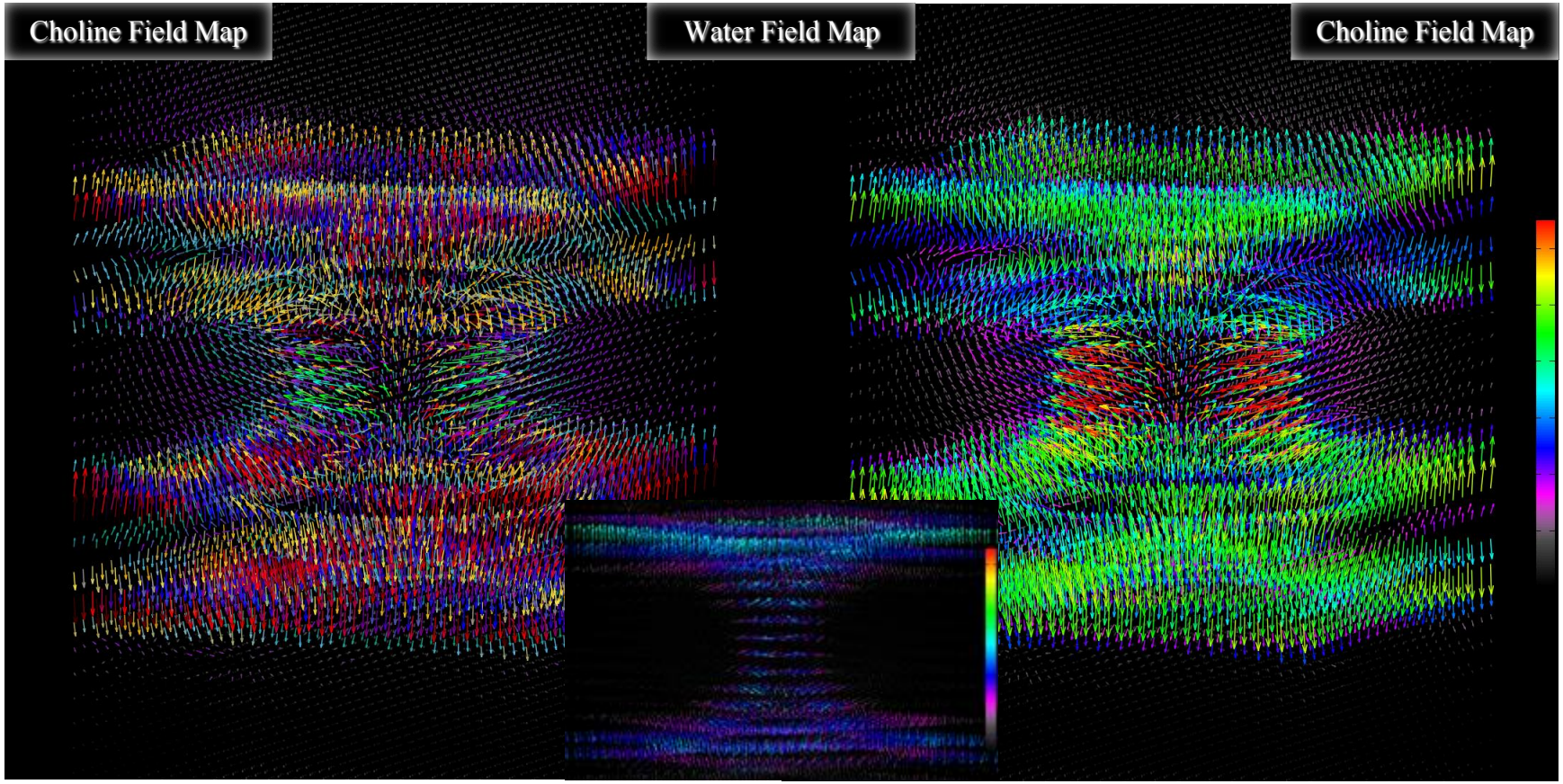

Water Potential Plot
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)


Sodium Potential Plot
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)

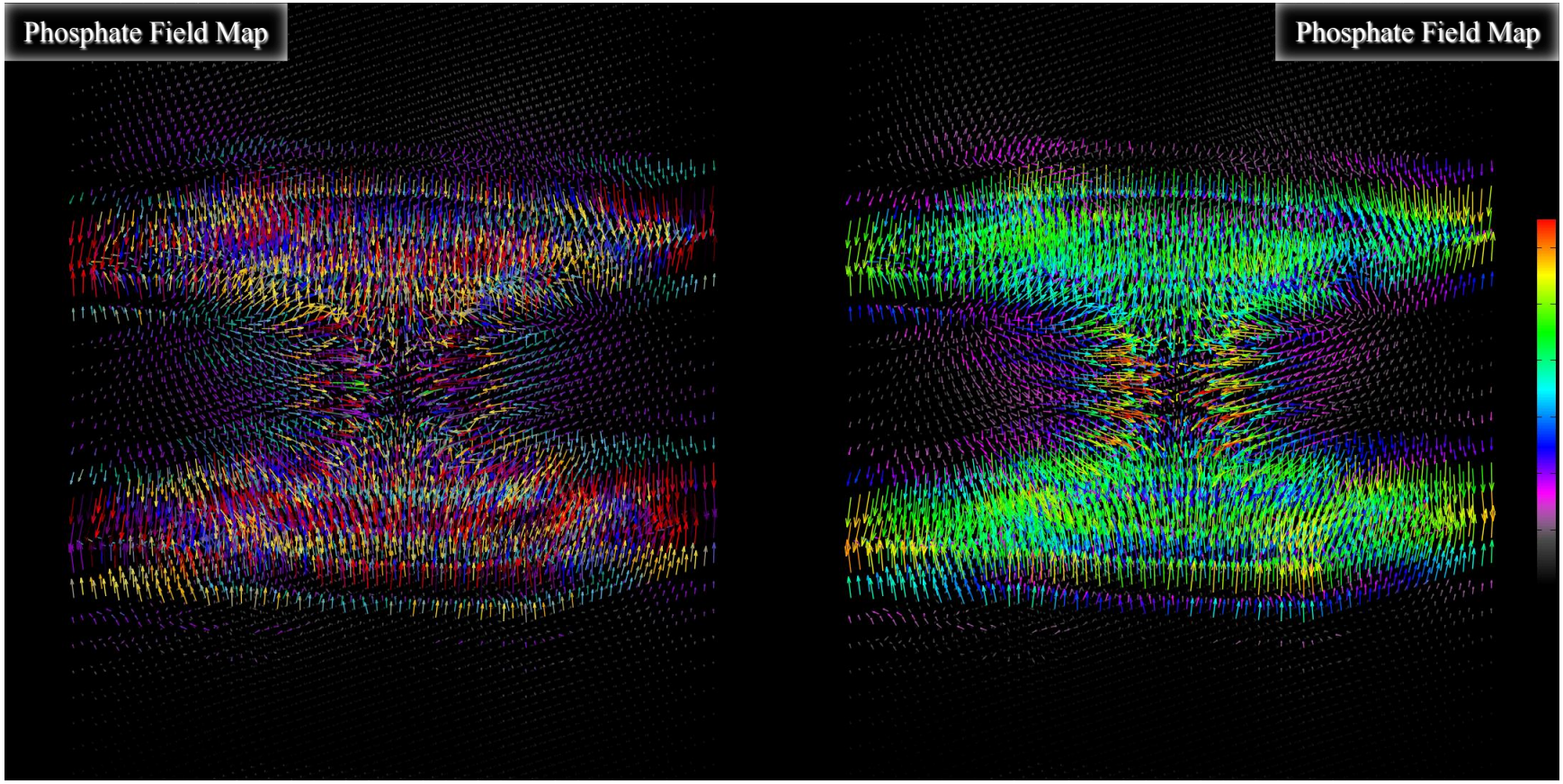

Chlorine Potential Plot
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)


SCMTR Headgroup Potential Plot
SCMTR Tails Potential Plot
Lipids Potential Plot
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-Axis (nm)
0
2
4
6
8
10
12
14
16
18
20
0
1
2
3
4
5
6
7
8

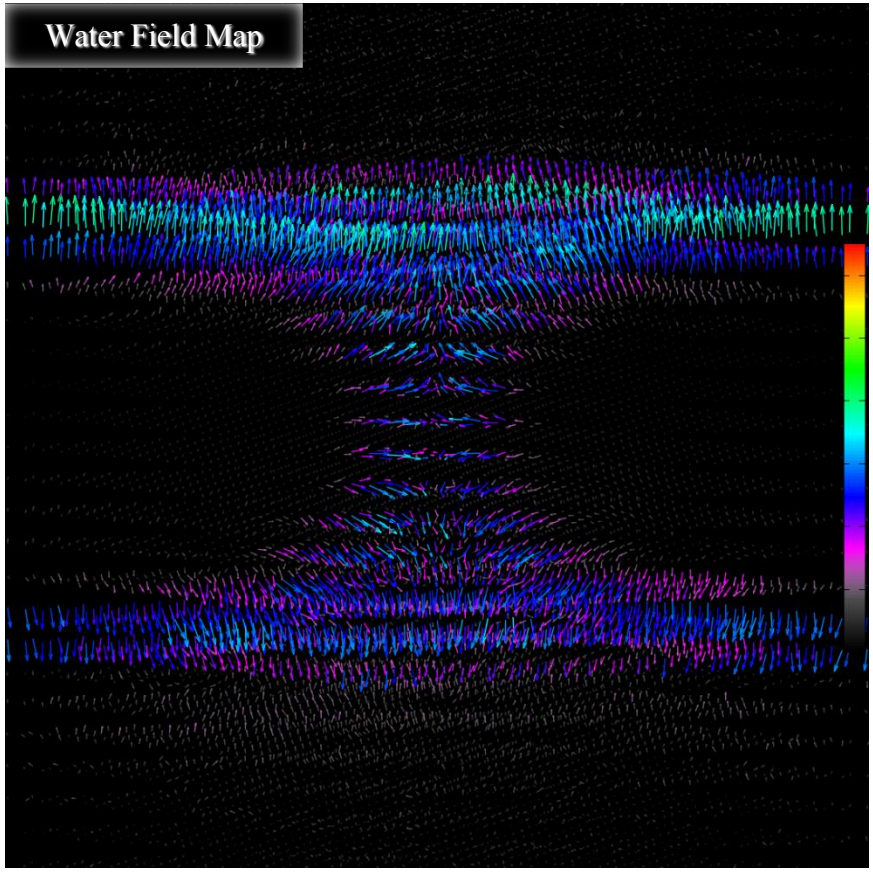

Lipid Tails Potential Plot
Z-Axis Coordinate (nm)
X-Axis Coordinate (nm)
Y-A (nm)
0
2
4
6
8
10
12
14
16
18
20
0 1 2 3 4 5 6 7 8



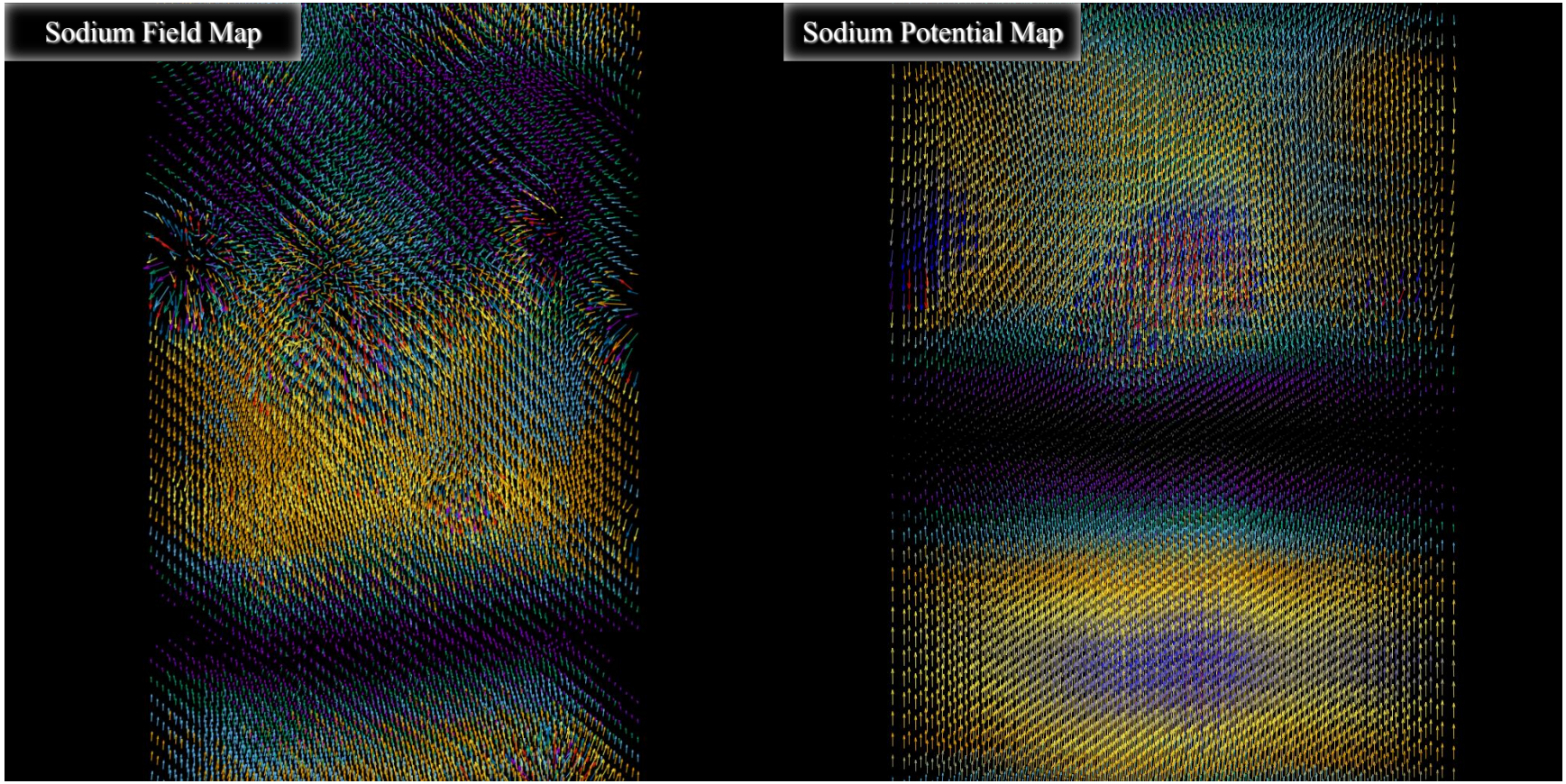
As mentioned several times, different scaling for color pallets are used for various purposes. In the left panel is a color scaling developed to hyper-accentuate slight differences in electric field strength. The right panel is a scaling used to compare electric fields of various moieties with a consistent scaling. Of all the moieties, by far those with the strongest coulombic interactions are the zwitterionic elements of the DOPC molecules, the choline and phosphate moieties. Because they are the strongest, they require the full palette range to plot. Compare the relative field strengths in the right panel to the central inset panel of water plotted at the same scale. Although of different magnitudes, each is the precise magnitude required to establish equilibrium, and in that respect, equally important to quantify and understand.



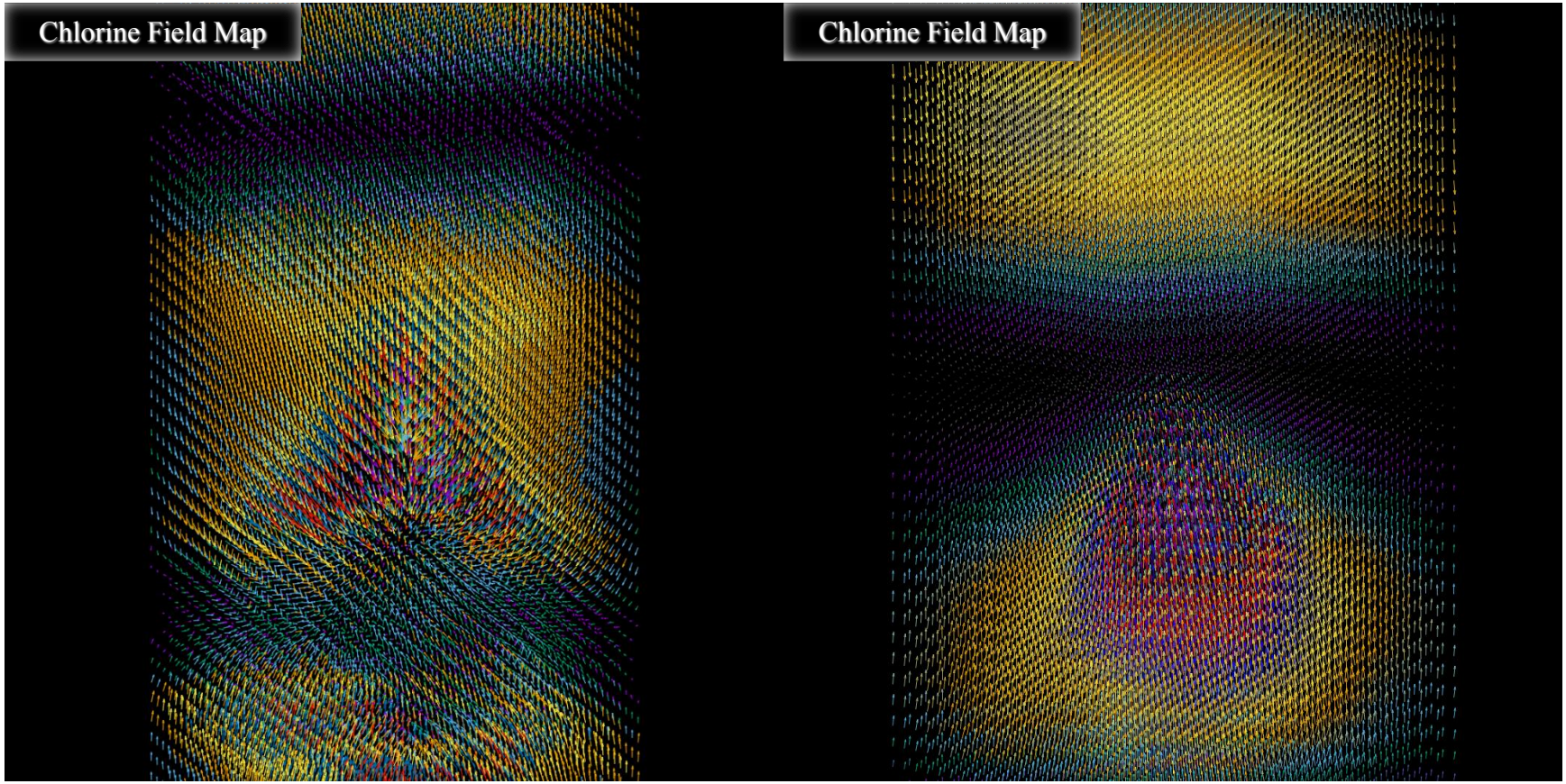
Here are the same series of images for the phosphate moiety. Observe the thickness of the choline layer on the previous page compared to the thickness of the phosphate moiety here. This is another indication of how the choline envelope surrounds the phosphate clusters.



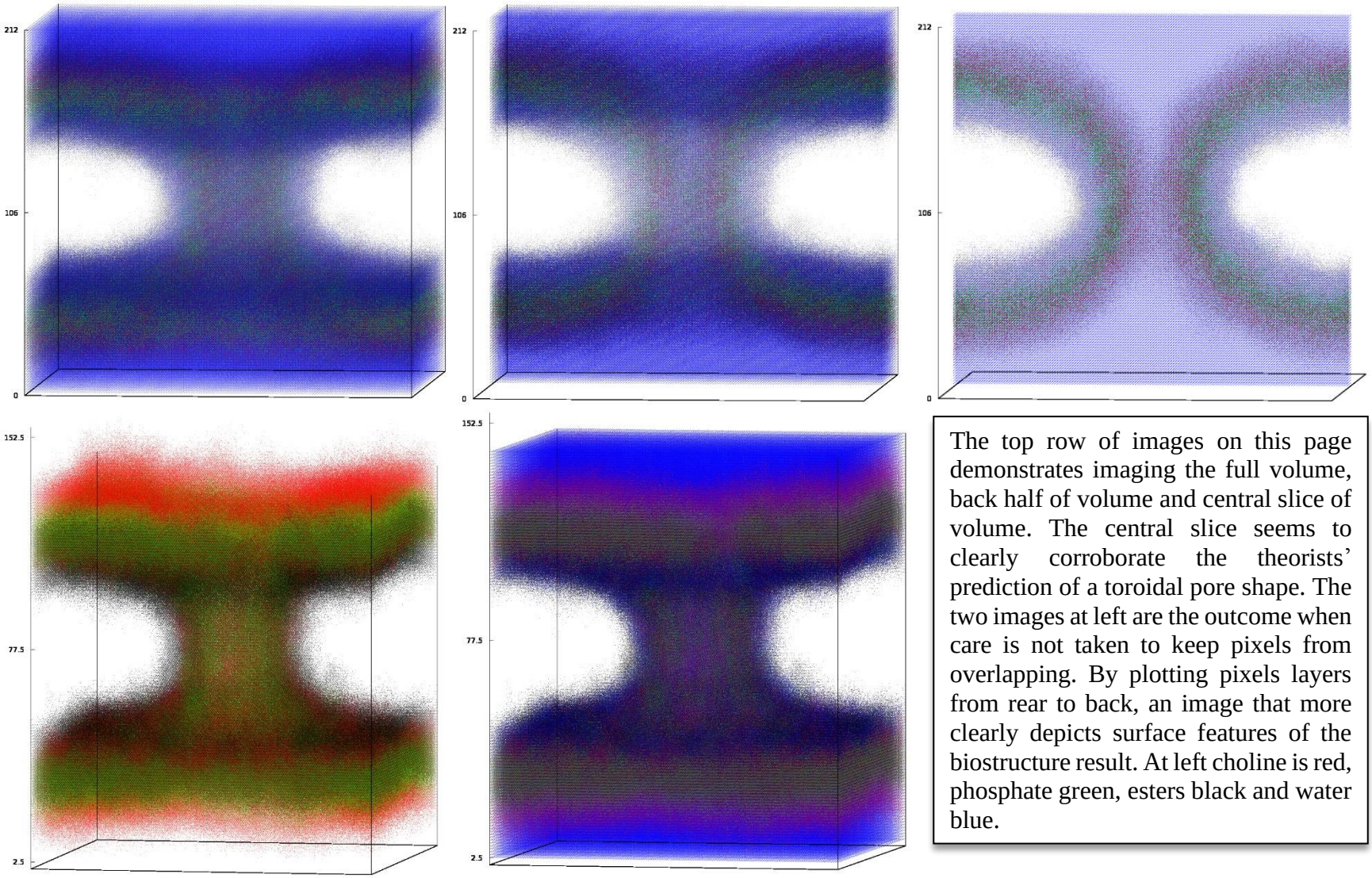
A final observation on one of the uniform scale images (on GitHub, these images are the B53 series). Note the slight assymetry of electric field strengths between the top leaflet layer and the bottom leaflet layer. This is due to the fact that relative to this membrane, the net negatively charged bath was above it during the simulation. Yet, even with this asymmetry, it is clear that without it, there would still be a net field in both leaflets pointing away from the hydrophobic region. To my committee: Since the field being revealed here is due only to common alignment of water dipoles, what does that reveal about the coulombic balance of charges between the zwitterionic elements of the lipids? I don't anticipate anybody seeing what I believe answers the next question, but I believe the data supporting the hypothesis is in this dissertation, ripe for experimental validation. How could a molecule be designed to want to embed itself in a DOPC membrane?



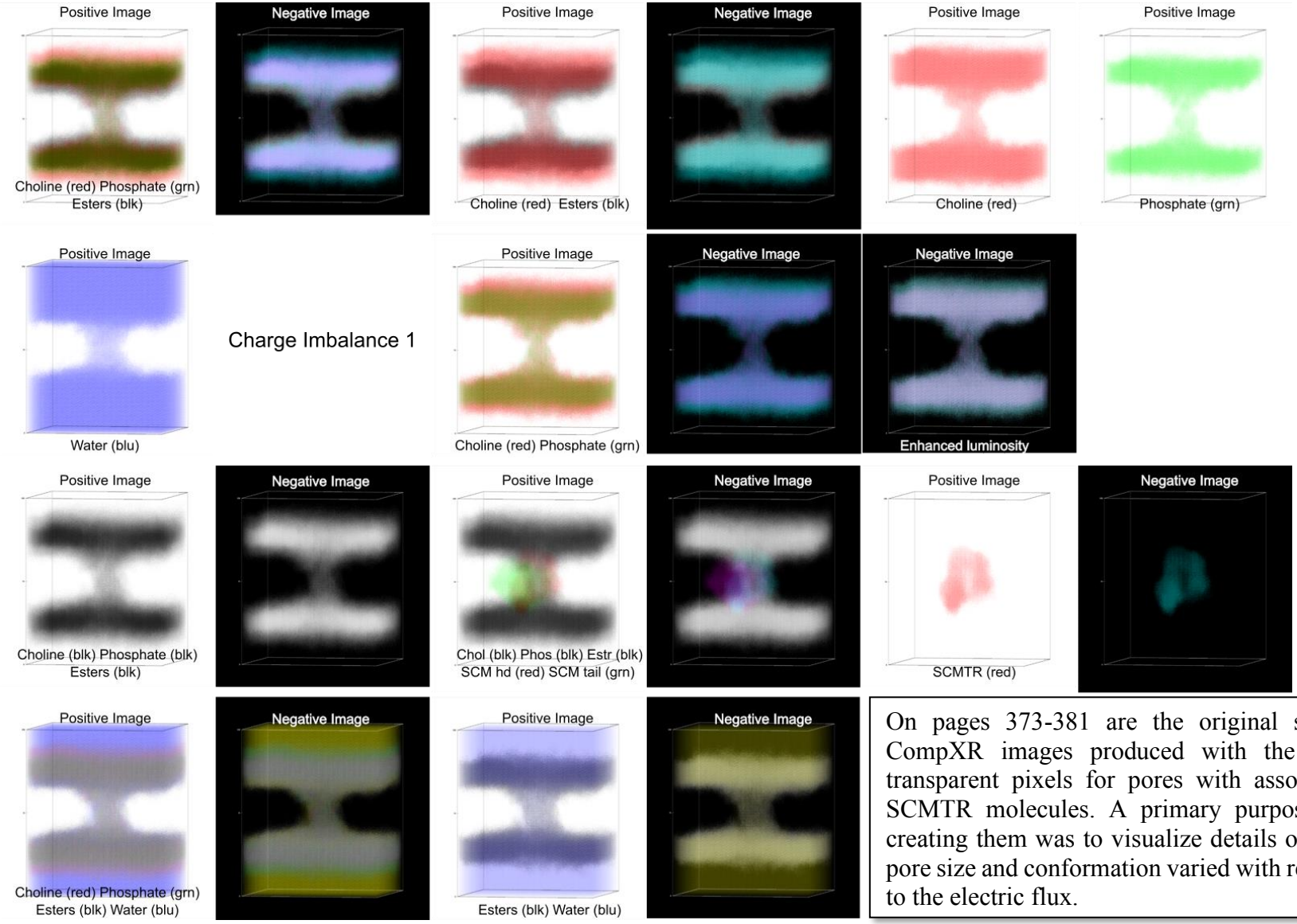
Here are the field map and potential map for the sodium cations. These images and the two on the next page for the chorine anions both include some expected features and some unexpected features. Movies of all the images individually have been produced to fully visualize the non-symmetrical aspects visible in the images. They need to be seen to be fully appreciated. These movie files are not small by GitHub standards which has a maximum size constraint for any individual file. The intent is to make these movies available through the UofL network infrastructure that will be available in perpetuity for the files (almost all movie files) that exceed the GitHub maximum size constraints.



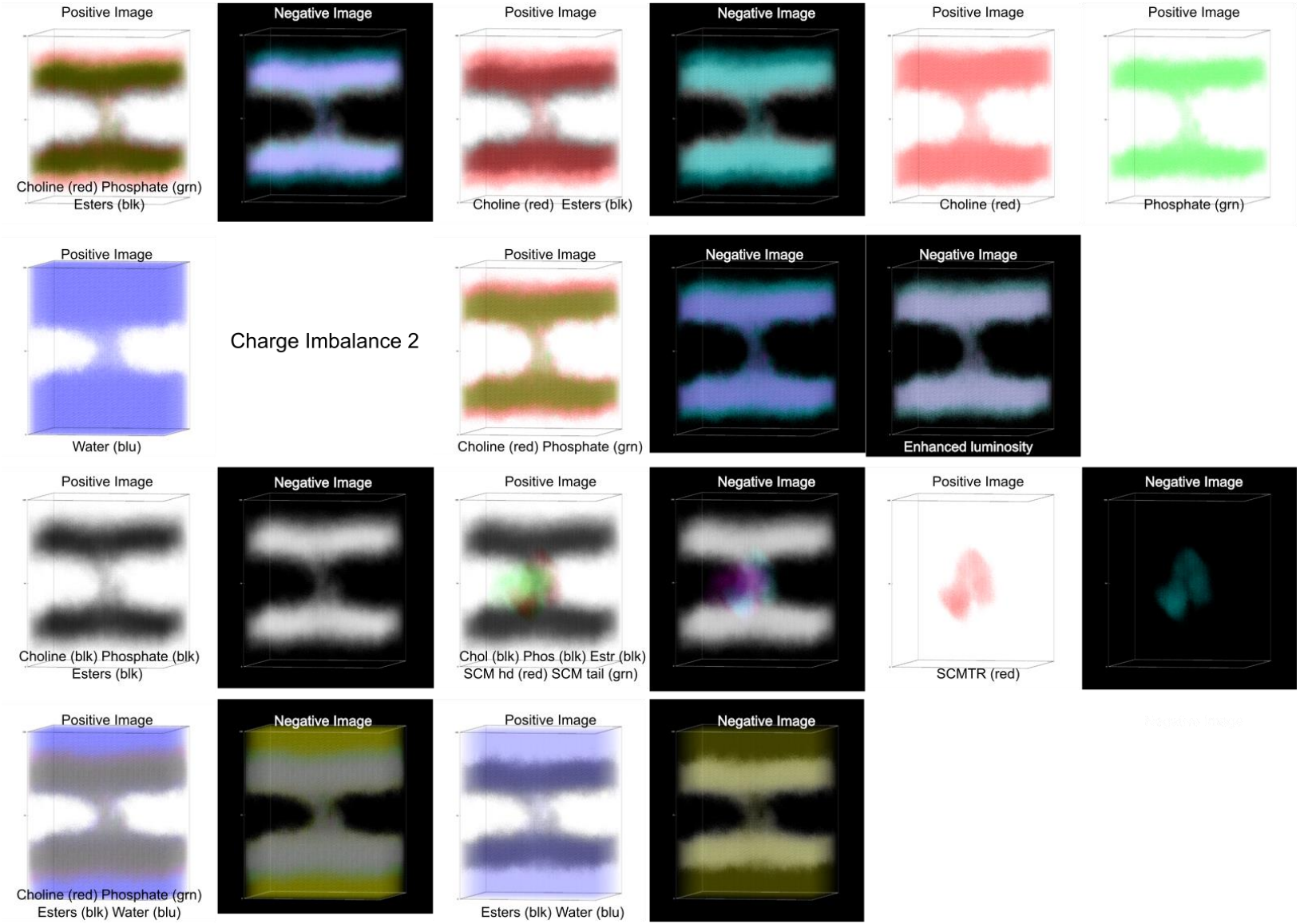
Carefully compare these images for the chorine anions with those of the sodium cations on the previous page. This is a simple comparison of the behavior of monovalent anions and cations. Is it reasonable to anticipate observing nearly identical behavior simply with opposite polarity? The images reveal the outcome of the simulations to be much more unique than anticipated.

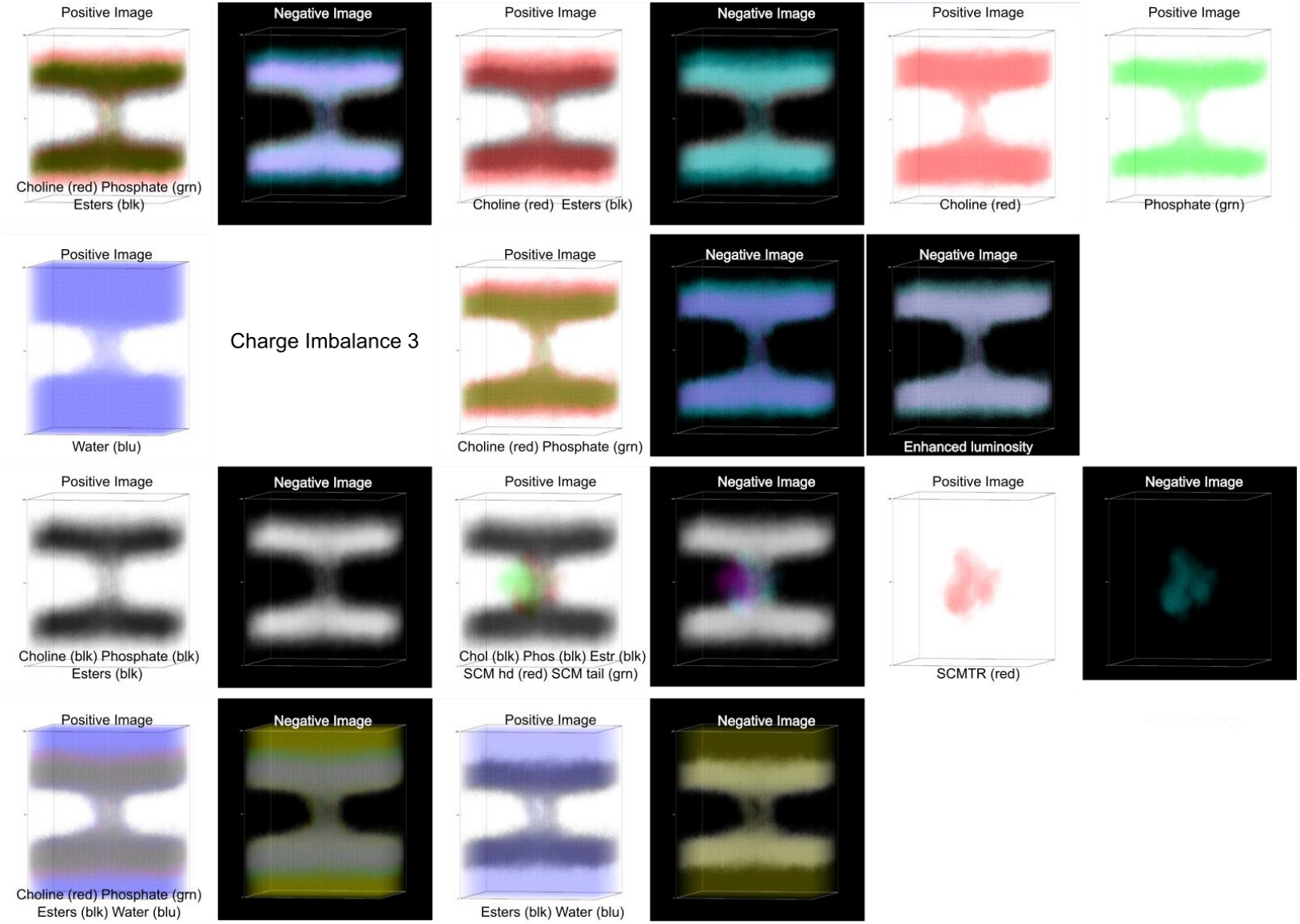


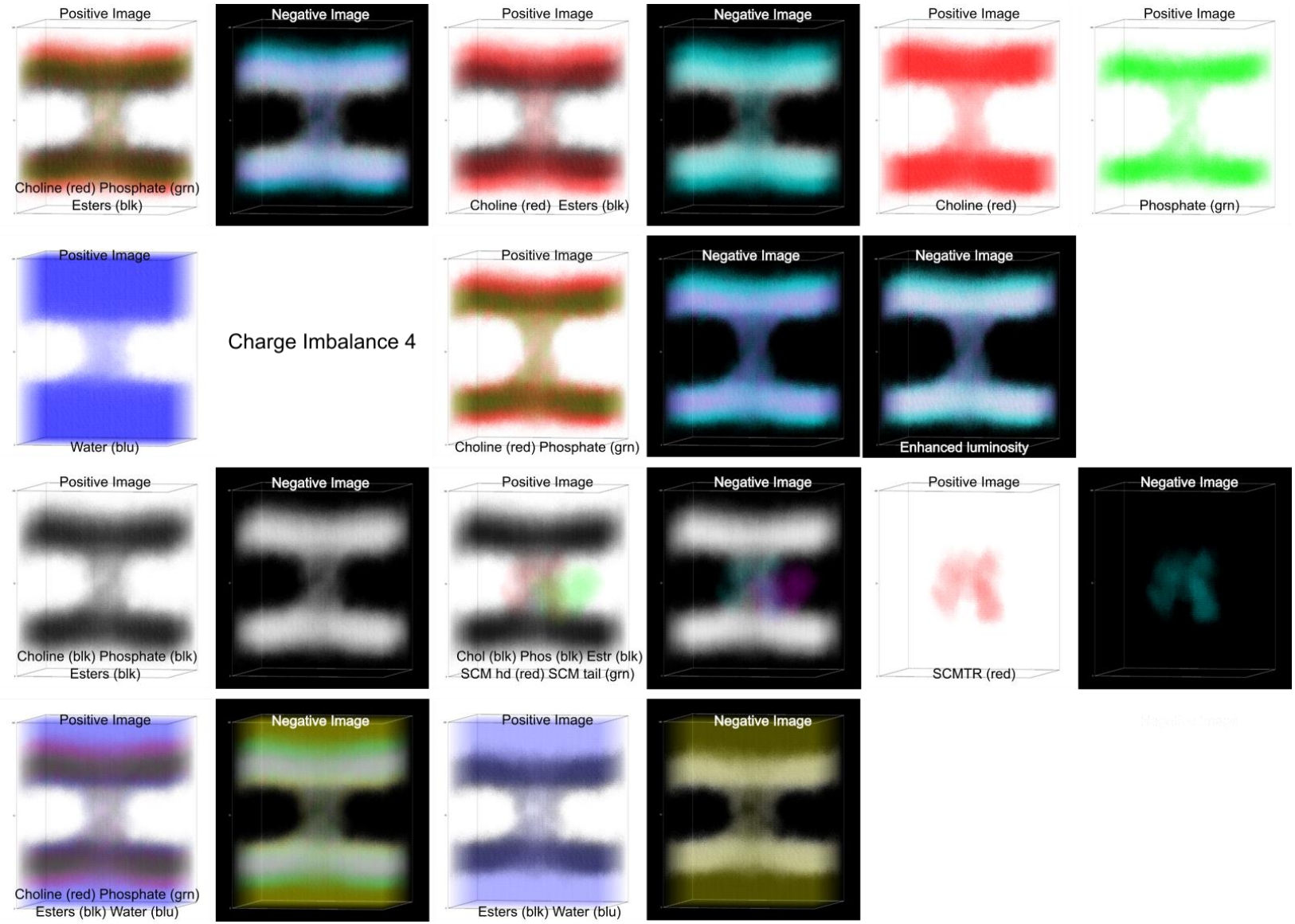
The top row of images on this page demonstrates imaging the full volume, back half of volume and central slice of volume. The central slice seems to clearly corroborate the theorists' prediction of a toroidal pore shape. The two images at left are the outcome when care is not taken to keep pixels from overlapping. By plotting pixels layers from rear to back, an image that more clearly depicts surface features of the biostructure result. At left choline is red, phosphate green, esters black and water blue.

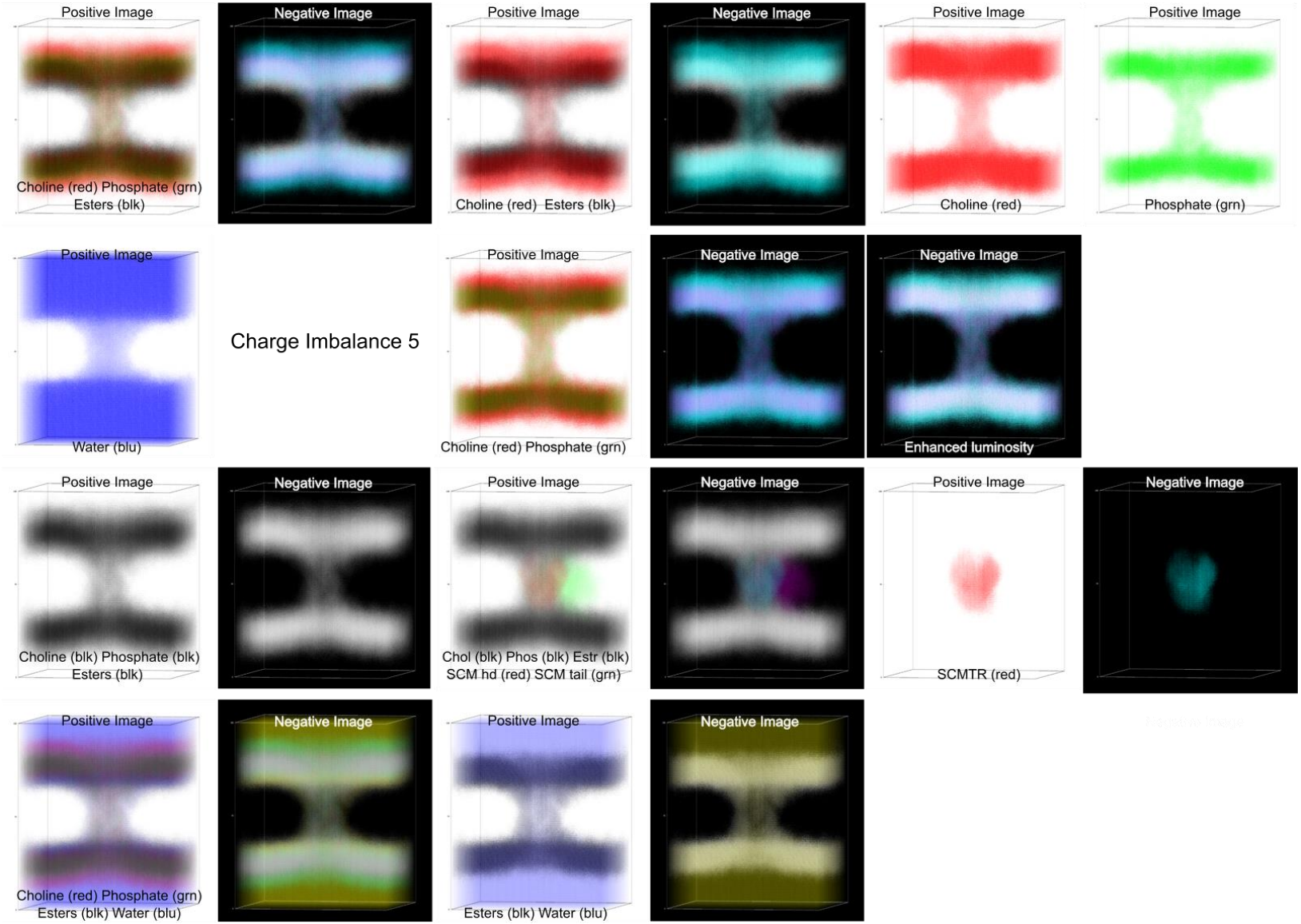


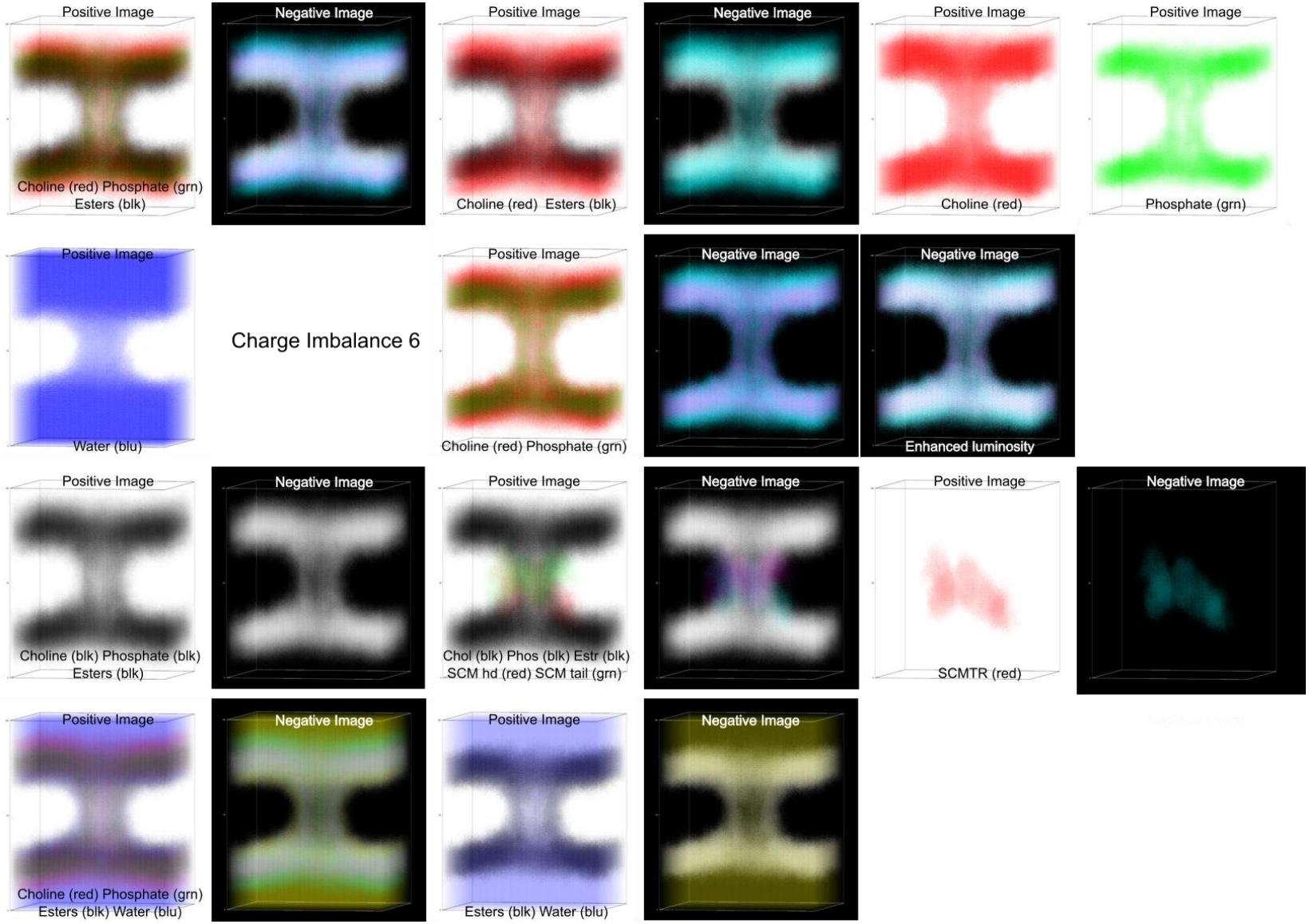
On pages 373-381 are the original set of CompXR images produced with the non-transparent pixels for pores with associated SCMTR molecules. A primary purpose for creating them was to visualize details of how pore size and conformation varied with respect to the electric flux.

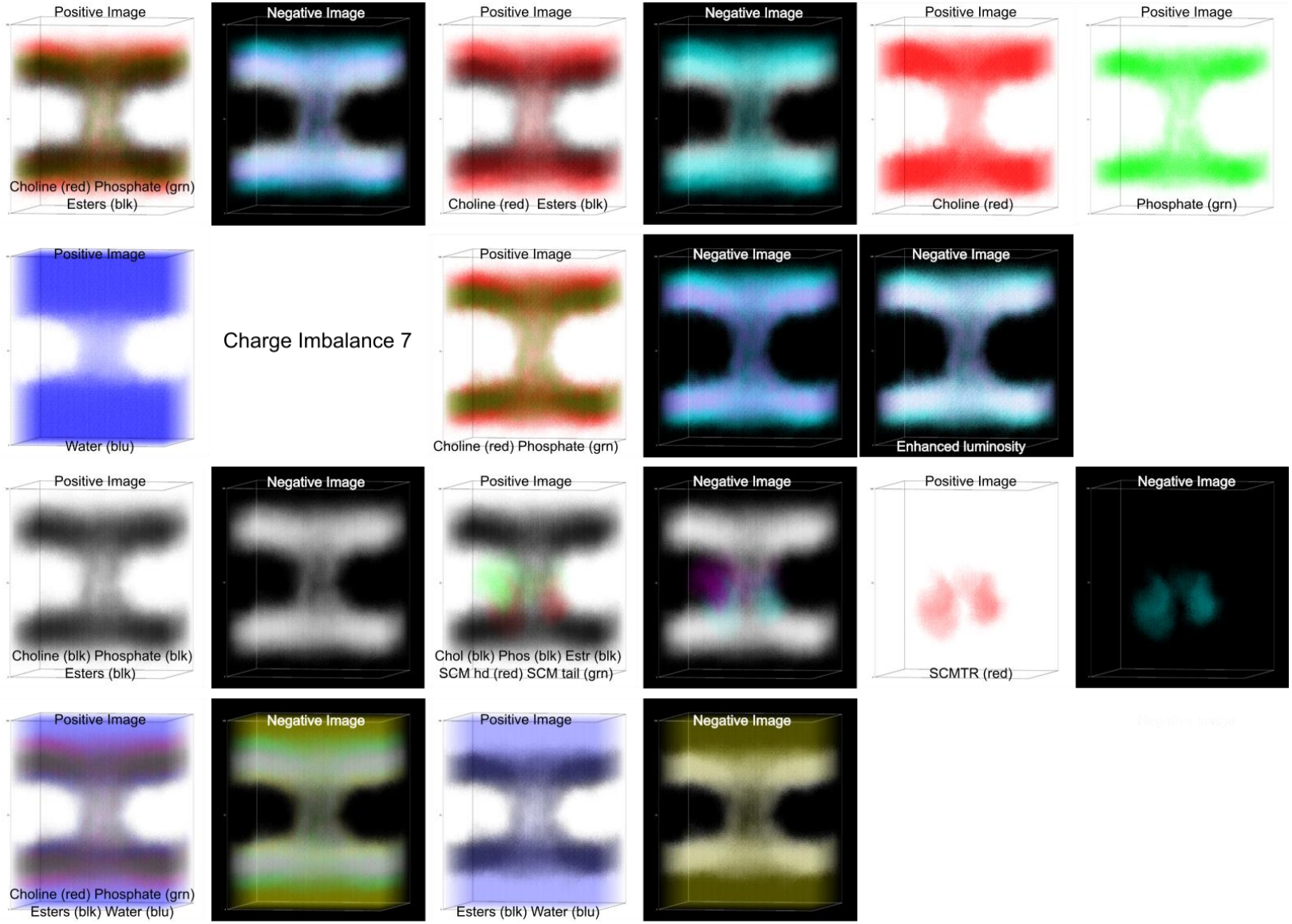

Positive Image
Choline (red) Phosphate (grn)
Esters (blk)
Negative Image
Positive Image
Choline (red) Esters (blk)
Negative Image
Positive Image
Choline (red)
Positive Image
Phosphate (grn)
Positive Image
Water (blu)
Charge Imbalance 2
Positive Image
Choline (red) Phosphate (grn)
Negative Image
Negative Image
Enhanced luminosity
Positive Image
Choline (blk) Phosphate (blk)
Esters (blk)
Negative Image
Positive Image
Chol (blk) Phos (blk) Estr (blk)
SCM hd (red) SCM tail (grn)
Negative Image
Positive Image
SCMTR (red)
Negative Image
Positive Image
Choline (red) Phosphate (grn)
Esters (blk) Water (blu)
Negative Image
Positive Image
Esters (blk) Water (blu)
Negative Image

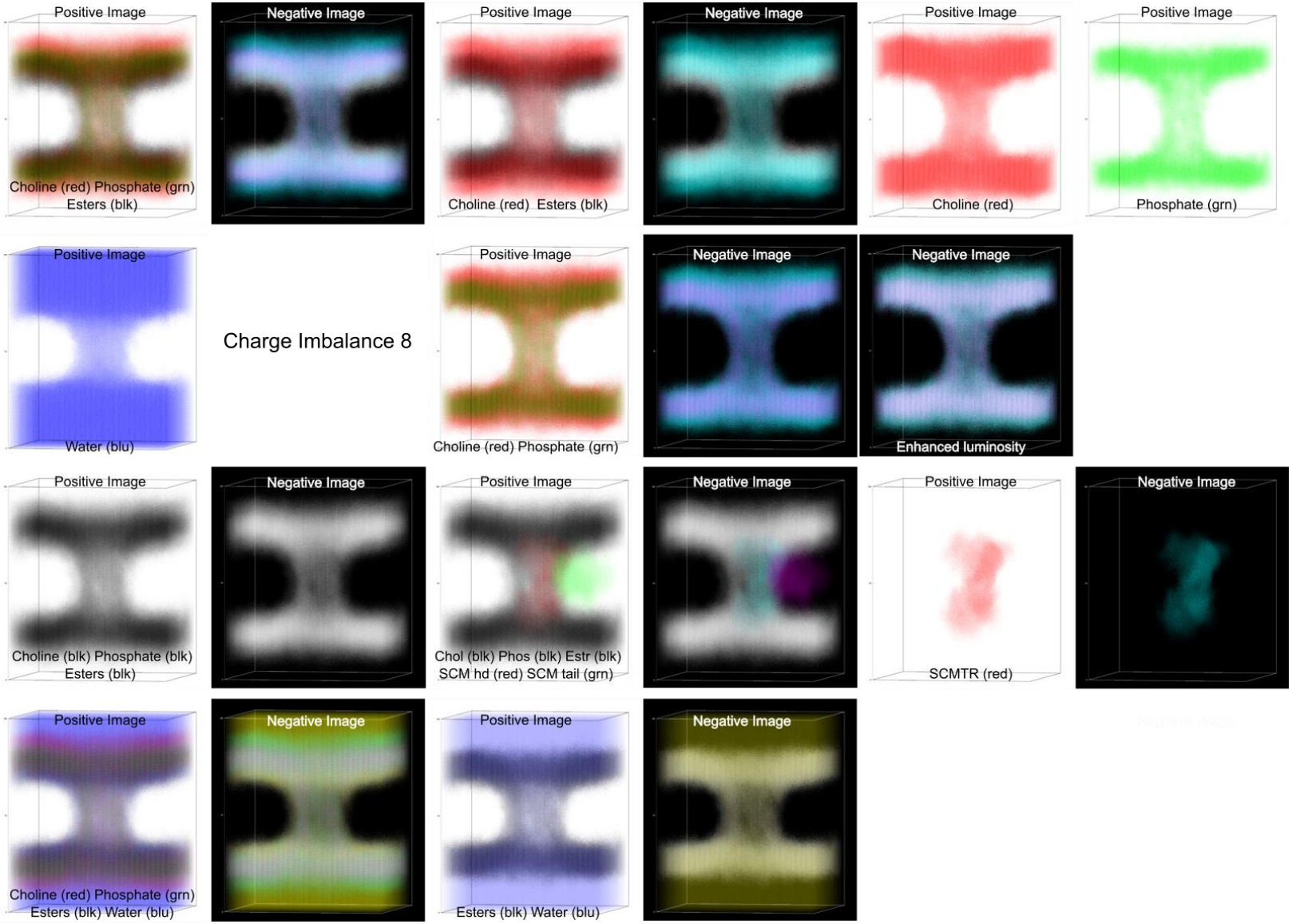

Positive Image
Negative Image
Positive Image
Negative Image
Positive Image
Positive Image
Choline (red) Phosphate (grn)
Esters (blk)
Choline (red) Esters (blk)
Choline (red)
Phosphate (grn)
Positive Image
Charge Imbalance 3
Positive Image
Negative Image
Negative Image
Water (blu)
Choline (red) Phosphate (grn)
Enhanced luminosity
Positive Image
Negative Image
Positive Image
Negative Image
Positive Image
Negative Image
Choline (blk) Phosphate (blk)
Esters (blk)
Chol (blk) Phos (blk) Estr (blk)
SCM hd (red) SCM tail (grn)
SCMTR (red)
Positive Image
Negative Image
Positive Image
Negative Image
Choline (red) Phosphate (grn)
Esters (blk) Water (blu)
Esters (blk) Water (blu)

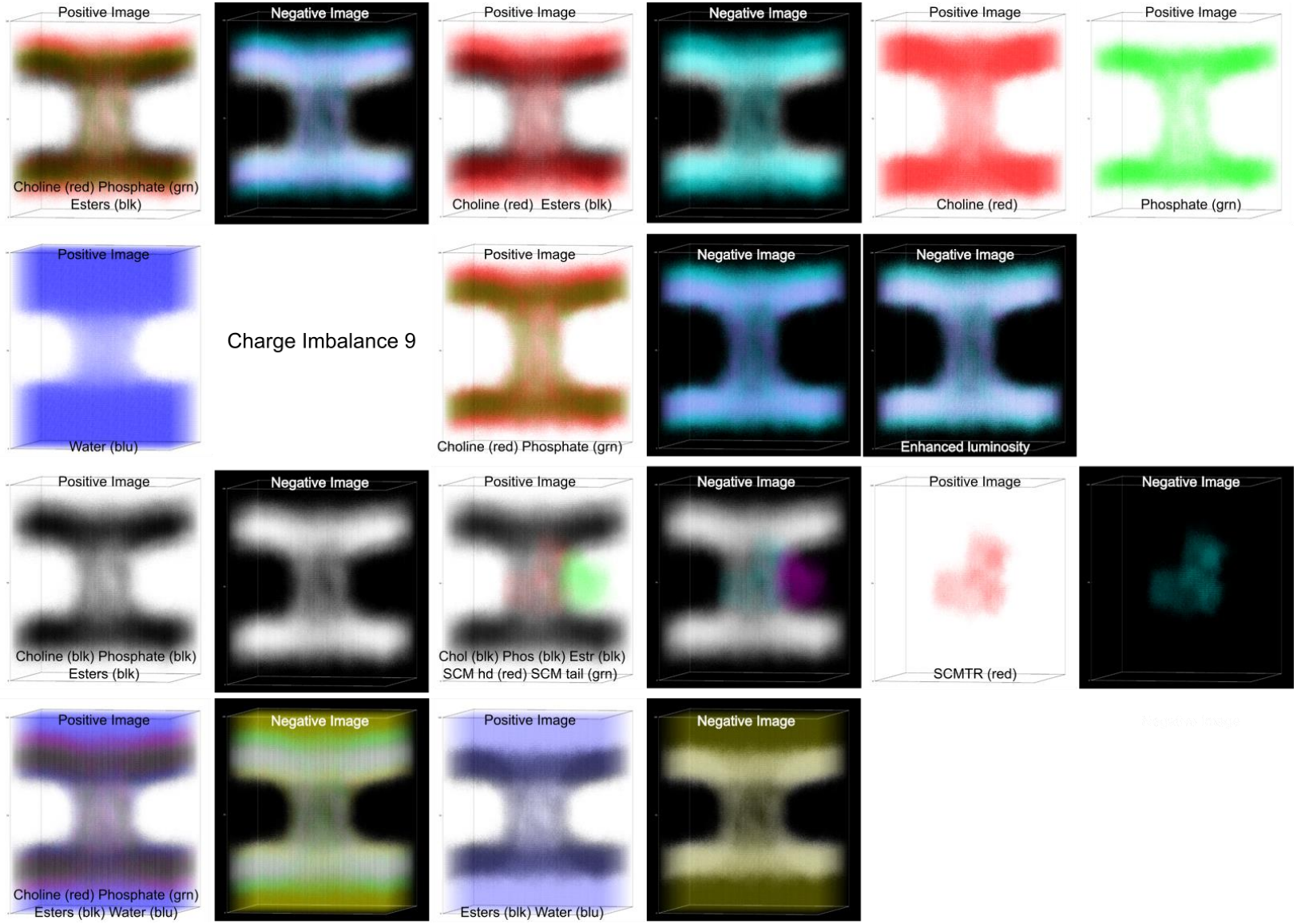

Positive Image
Negative Image
Positive Image
Negative Image
Positive Image
Positive Image
Choline (red) Phosphate (grn)
Esters (blk)
Choline (red) Esters (blk)
Choline (red)
Phosphate (grn)
Positive Image
Charge Imbalance 4
Positive Image
Negative Image
Negative Image
Water (blu)
Choline (red) Phosphate (grn)
Enhanced luminosity
Positive Image
Negative Image
Positive Image
Negative Image
Positive Image
Negative Image
Choline (blk) Phosphate (blk)
Esters (blk)
Chol (blk) Phos (blk) Estr (blk)
SCM hd (red) SCM tail (grn)
SCMTR (red)
Positive Image
Negative Image
Positive Image
Negative Image
Choline (red) Phosphate (grn)
Esters (blk) Water (blu)
Esters (blk) Water (blu)


Positive Image
Negative Image
Positive Image
Negative Image
Positive Image
Positive Image
Choline (red) Phosphate (grn)
Esters (blk)
Choline (red) Esters (blk)
Choline (red)
Phosphate (grn)
Positive Image
Charge Imbalance 5
Positive Image
Negative Image
Negative Image
Water (blu)
Choline (red) Phosphate (grn)
Enhanced luminosity
Positive Image
Negative Image
Positive Image
Negative Image
Positive Image
Negative Image
Choline (blk) Phosphate (blk)
Esters (blk)
Chol (blk) Phos (blk) Estr (blk)
SCM hd (red) SCM tail (grn)
SCMTR (red)
Positive Image
Negative Image
Positive Image
Negative Image
Choline (red) Phosphate (grn)
Esters (blk) Water (blu)
Esters (blk) Water (blu)


Positive Image
Choline (red) Phosphate (grn)
Esters (blk)
Negative Image
Positive Image
Choline (red) Esters (blk)
Negative Image
Positive Image
Choline (red)
Positive Image
Phosphate (grn)
Positive Image
Water (blu)
Charge Imbalance 6
Positive Image
Choline (red) Phosphate (grn)
Negative Image
Negative Image
Enhanced luminosity
Positive Image
Choline (blk) Phosphate (blk)
Esters (blk)
Negative Image
Positive Image
Chol (blk) Phos (blk) Estr (blk)
SCM hd (red) SCM tail (grn)
Negative Image
Positive Image
SCMTR (red)
Negative Image
Positive Image
Choline (red) Phosphate (grn)
Esters (blk) Water (blu)
Negative Image
Positive Image
Esters (blk) Water (blu)
Negative Image


Positive Image
Choline (red) Phosphate (grn)
Esters (blk)
Negative Image
Positive Image
Choline (red) Esters (blk)
Negative Image
Positive Image
Choline (red)
Positive Image
Phosphate (grn)
Positive Image
Water (blu)
Charge Imbalance 7
Positive Image
Choline (red) Phosphate (grn)
Negative Image
Negative Image
Enhanced luminosity
Positive Image
Choline (blk) Phosphate (blk)
Esters (blk)
Negative Image
Positive Image
Chol (blk) Phos (blk) Estr (blk)
SCM hd (red) SCM tail (grn)
Negative Image
Positive Image
SCMTR (red)
Negative Image
Positive Image
Choline (red) Phosphate (grn)
Esters (blk) Water (blu)
Negative Image
Positive Image
Esters (blk) Water (blu)
Negative Image


Positive Image
Negative Image
Positive Image
Negative Image
Positive Image
Positive Image
Choline (red) Phosphate (grn)
Esters (blk)
Choline (red) Esters (blk)
Choline (red)
Phosphate (grn)
Positive Image
Charge Imbalance 8
Positive Image
Negative Image
Negative Image
Water (blu)
Choline (red) Phosphate (grn)
Enhanced luminosity
Positive Image
Negative Image
Positive Image
Negative Image
Positive Image
Negative Image
Choline (blk) Phosphate (blk)
Esters (blk)
Chol (blk) Phos (blk) Estr (blk)
SCM hd (red) SCM tail (grn)
SCMTR (red)
Positive Image
Negative Image
Positive Image
Negative Image
Choline (red) Phosphate (grn)
Esters (blk) Water (blu)
Esters (blk) Water (blu)


Positive Image
Choline (red) Phosphate (grn)
Esters (blk)
Negative Image
Positive Image
Choline (red) Esters (blk)
Negative Image
Positive Image
Choline (red)
Positive Image
Phosphate (grn)
Positive Image
Water (blu)
Charge Imbalance 9
Positive Image
Choline (red) Phosphate (grn)
Negative Image
Negative Image
Enhanced luminosity
Positive Image
Choline (blk) Phosphate (blk)
Esters (blk)
Negative Image
Positive Image
Chol (blk) Phos (blk) Estr (blk)
SCM hd (red) SCM tail (grn)
Negative Image
Positive Image
SCMTR (red)
Negative Image
Positive Image
Choline (red) Phosphate (grn)
Esters (blk) Water (blu)
Negative Image
Positive Image
Esters (blk) Water (blu)
Negative Image

Appendix C

Section 3: Simulation Convergence Details, Free Energy Surface Plots and Radial Distribution Plots

Simulation Convergence Details

Simulation convergence was evaluated using the native contacts CV. The values at each simulation time represent the free energy change in the transition from unfolded to folded, hence negative values represent an energy minimum in the folded state. The free energy change of this transition at 5.0 μs are the values reported in the free energy surface plots.

TRPC		Simulation Time (μs)				Std. Dev.	ΔA	
4.0	**4.2**	**4.4**	**4.6**	**4.8**	**5.0**	Kcal/mol	Kcal/mol	BenchM?
0.571	-0.005	-0.018	-0.446	-0.254	-0.050	0.342	0.204	1

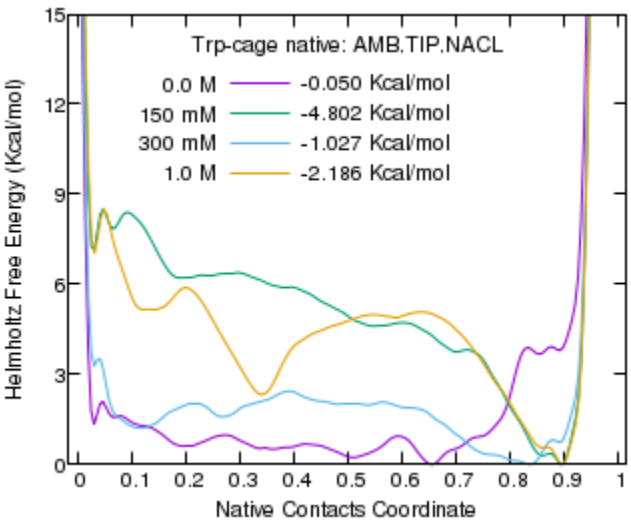
TRPC		Simulation Time (μs)				Std. Dev.	ΔA	
4.0	**4.2**	**4.4**	**4.6**	**4.8**	**5.0**	Kcal/mol	Kcal/mol	BenchM?
-1.698	-2.397	-3.082	-3.762	-4.338	-4.802	1.180	-0.464	0
-0.116	-0.107	-0.089	-0.481	-0.680	-1.027	0.385	-0.348	1
-5.539	-4.134	-3.940	-3.427	-2.187	-2.186	1.279	0.000	1
0.650	-0.045	-0.560	-0.880	-1.495	-1.886	0.932	-0.390	1
-2.271	-1.487	-0.754	-0.346	-0.346	-0.342	0.796	0.004	1
0.814	0.316	-0.115	-0.670	-0.882	-1.420	0.822	-0.539	1
0.592	0.048	-0.266	-0.741	-1.024	-0.857	0.619	0.167	1
0.571	-0.005	-0.018	-0.446	-0.254	-0.050	0.342	0.204	1
-3.074	-2.608	-2.013	-1.451	-0.728	-0.443	1.039	0.285	0
-0.535	-0.524	0.061	0.252	-0.112	-0.286	0.318	-0.174	1
-0.706	0.118	0.508	0.389	-0.040	-0.128	0.433	-0.088	1
0.650	-0.045	-0.560	-0.880	-1.495	-1.886	0.932	-0.390	1
0.467	0.510	0.905	0.949	0.888	0.673	0.212	-0.215	1
2.111	2.412	1.623	1.643	1.653	1.658	0.333	0.005	1
-1.432	-1.205	-0.945	-0.635	-0.039	0.204	0.649	0.243	1
5.262	5.313	4.405	4.476	4.375	4.217	0.483	-0.158	1
-0.999	-1.039	-0.890	-0.741	-0.480	-0.382	0.273	0.099	1
5.977	4.265	4.379	4.739	5.007	5.340	0.640	0.333	1
3.413	3.685	3.802	3.886	4.027	4.106	0.250	0.080	1
5.262	5.313	4.405	4.476	4.375	4.217	0.483	-0.158	1
2.457	2.684	3.005	3.285	3.413	3.615	0.445	0.202	1
-0.489	-1.209	-1.687	-1.638	-1.401	-0.942	0.455	0.459	1
-0.277	-0.313	-0.297	-0.250	-0.018	0.046	0.157	0.064	1
-4.965	-5.555	-5.966	-6.441	-6.865	-6.905	0.768	-0.040	1
-4.640	-5.231	-5.509	-6.074	-6.286	-5.675	0.593	0.611	1
-3.930	-4.064	-4.112	-4.113	-4.607	-3.589	0.329	1.018	1
1.962	1.930	0.471	-0.028	-0.644	-0.706	1.202	-0.063	1
-4.965	-5.555	-5.966	-6.441	-6.865	-6.905	0.768	-0.040	1
-2.745	-3.426	-3.965	-4.100	-4.313	-4.429	0.636	-0.116	1
0.046	-0.203	-0.979	-1.780	-2.429	-2.898	1.195	-0.469	0
-0.061	-0.500	-0.823	-1.412	-2.020	-2.263	0.869	-0.243	1
							Total	**29**

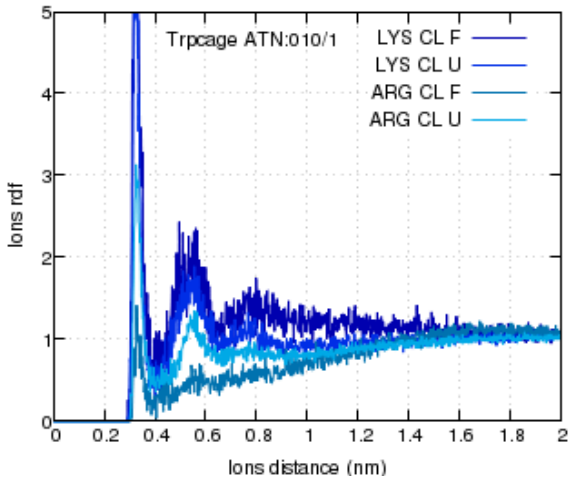
GB1		Simulation Time (μs)				Std. Dev.	ΔA	
4.0	**4.2**	**4.4**	**4.6**	**4.8**	**5.0**	Kcal/mol	Kcal/mol	BenchM?
4.724	4.801	4.919	5.028	5.118	4.557	0.206	-0.560	1
3.899	3.831	3.978	4.078	4.395	4.364	0.238	-0.031	1
3.560	3.817	4.015	4.016	4.397	4.568	0.370	0.171	1
-0.740	-0.267	-0.044	0.409	0.624	0.771	0.579	0.147	1
1.289	1.500	1.882	2.143	2.273	2.467	0.458	0.194	1
3.528	3.656	3.845	3.936	4.011	4.020	0.201	0.009	1
4.864	5.083	4.983	4.919	4.782	4.384	0.244	-0.397	1
20.609	21.153	21.970	22.485	22.636	23.959	1.185	1.323	0
3.072	3.290	3.774	4.265	4.397	4.534	0.609	0.137	1
4.458	4.150	4.457	4.699	5.023	5.354	0.436	0.331	1
-0.539	-0.069	0.349	0.441	0.532	0.585	0.437	0.053	1
-1.358	-1.400	-1.991	-1.996	-2.058	-1.874	0.316	0.184	1
3.044	1.781	1.180	0.676	0.351	0.318	1.049	-0.033	1
3.097	3.165	3.283	3.378	3.254	3.345	0.107	0.090	1
-0.282	-1.570	-2.019	-1.912	-1.743	-1.570	0.631	0.173	1
3.314	3.301	3.382	3.468	3.565	3.599	0.127	0.034	1
-2.933	-3.323	-3.630	-3.736	-4.250	-4.522	0.585	-0.273	1
5.051	5.474	5.345	4.079	2.272	1.601	1.663	-0.672	0
-5.072	-5.296	-5.571	-5.792	-5.155	-4.482	0.453	0.673	1
2.760	1.814	0.994	0.450	-0.035	-0.469	1.204	-0.433	0
2.254	2.600	2.787	3.055	3.326	3.177	0.399	-0.149	1
2.490	2.664	2.657	2.962	3.230	3.482	0.383	0.252	1
2.039	1.726	1.695	1.926	2.228	2.468	0.299	0.240	1
-4.336	-4.485	-4.793	-4.847	-3.093	-3.059	0.817	0.034	1
6.746	6.935	4.448	4.130	1.797	1.429	2.344	-0.369	1
-6.208	-6.503	-5.600	-5.189	-4.181	-3.354	1.210	0.828	0
3.307	3.740	3.869	3.956	3.921	3.840	0.240	-0.081	1
6.175	6.386	6.487	6.673	6.925	6.834	0.284	-0.090	1
6.162	6.399	5.837	3.505	1.778	1.040	2.352	-0.738	0
5.770	5.955	6.294	6.666	7.011	7.065	0.542	0.054	1
4.052	4.427	4.664	4.962	4.974	4.973	0.377	-0.001	1
4.288	4.712	5.066	5.427	5.604	5.621	0.537	0.016	1
							Total	**27**

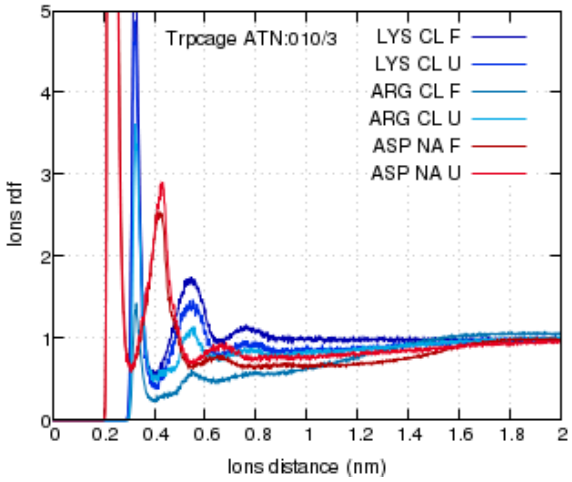
Free Energy Surface Plots and Radial Distribution Plots

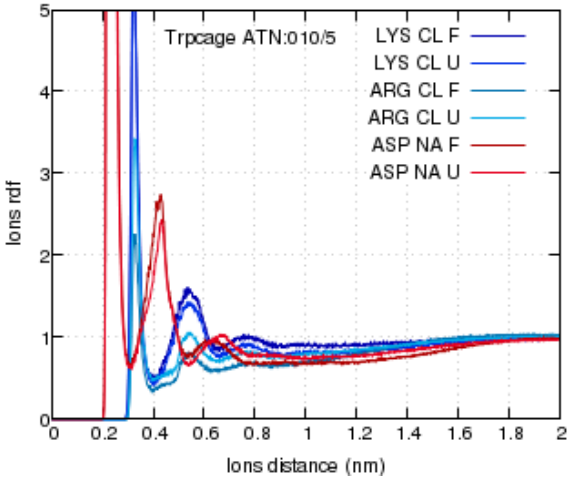
Radial distribution plots show data from single simulations whereas free energy surface plots can combine results for all 4 concentrations simulated. For convenience, the RDFs for the FF, WM, SS combination are shown after the FES plot for the same parameter combination. The FES plots can be compared to any set of the RDFs, Native, Helix, Radius of Gyration and Hydrophobic.

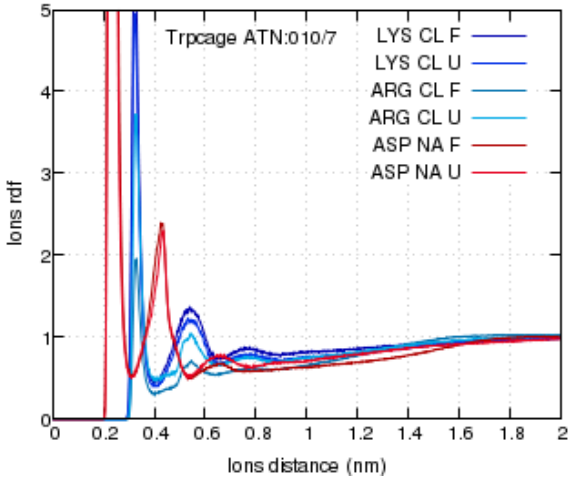
TRPC

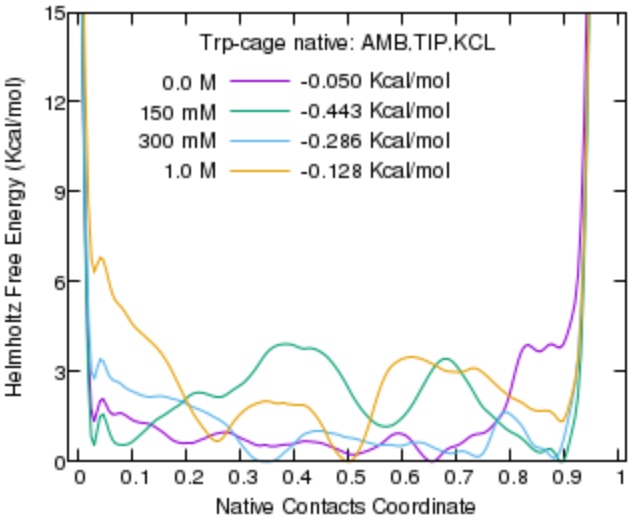

Trp-cage native: AMB.TIP.NACL
0.0 M -0.050 Kcal/mol
150 mM -4.802 Kcal/mol
300 mM -1.027 Kcal/mol
1.0 M -2.186 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

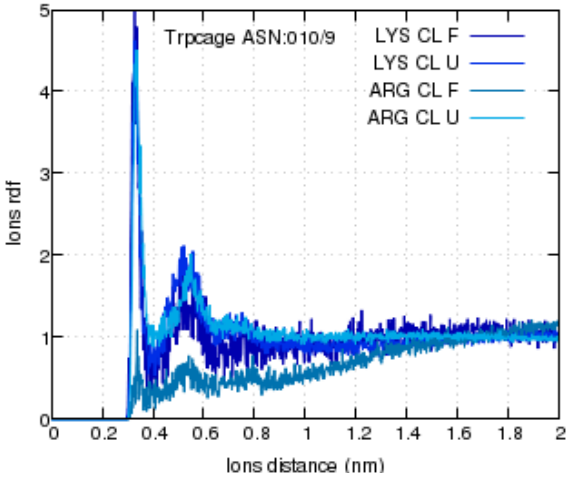

Trpcage ATN:010/1
LYS CL F
LYS CL U
ARG CL F
ARG CL U
Ions rdf
Ions distance (nm)

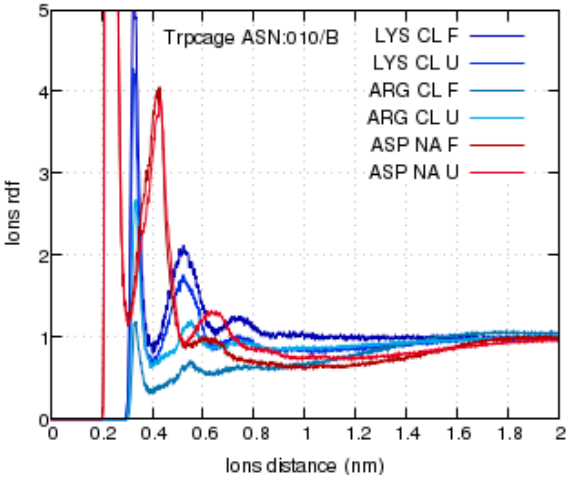

Trpcage ATN:010/3
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

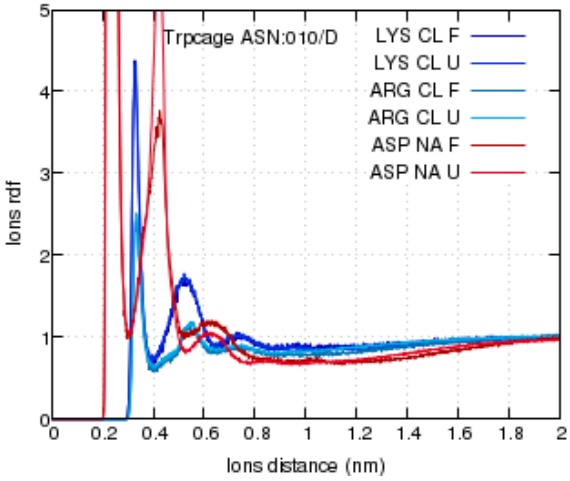

Trpcage ATN:010/5
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

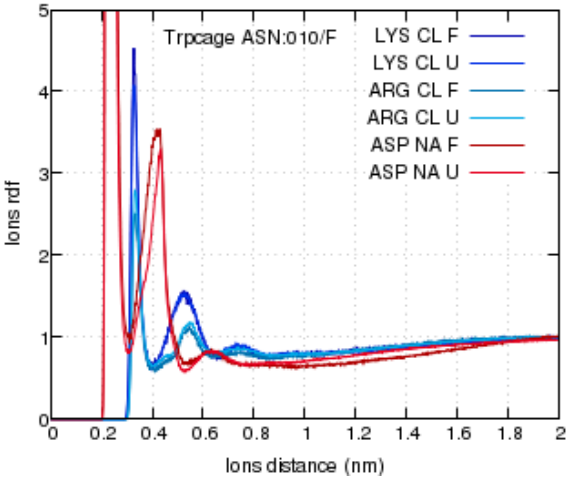

Trpcage ATN:010/7
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

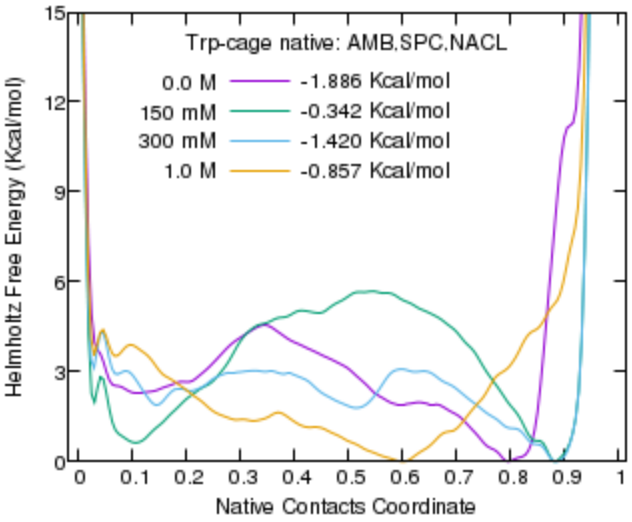

Trp-cage native: AMB.TIP.KCL
0.0 M -0.050 Kcal/mol
150 mM -0.443 Kcal/mol
300 mM -0.286 Kcal/mol
1.0 M -0.128 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

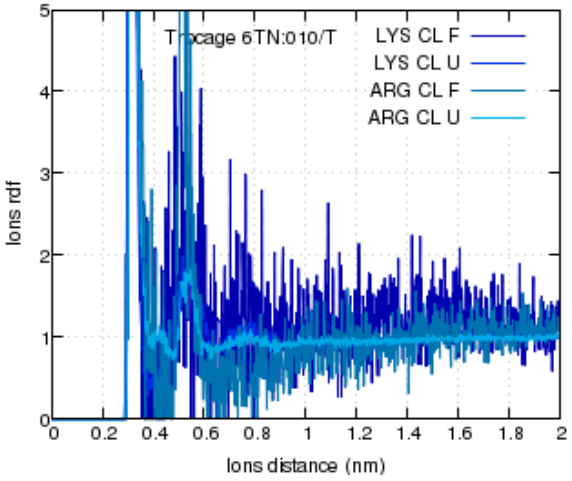

Trpcage ASN:010/9
LYS CL F
LYS CL U
ARG CL F
ARG CL U
Ions rdf
Ions distance (nm)

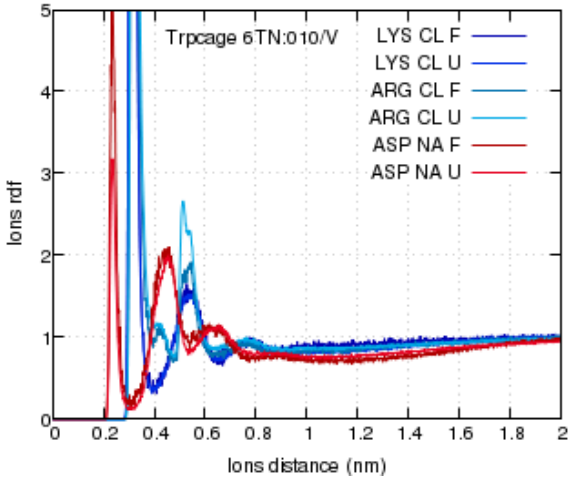

Trpcage ASN:010/B
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

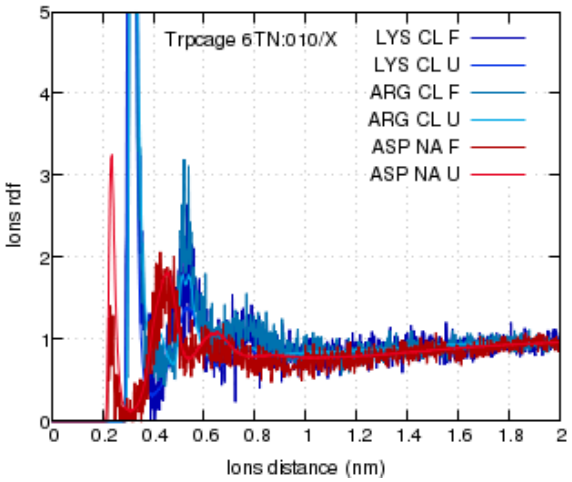

Trpcage ASN:010/D
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

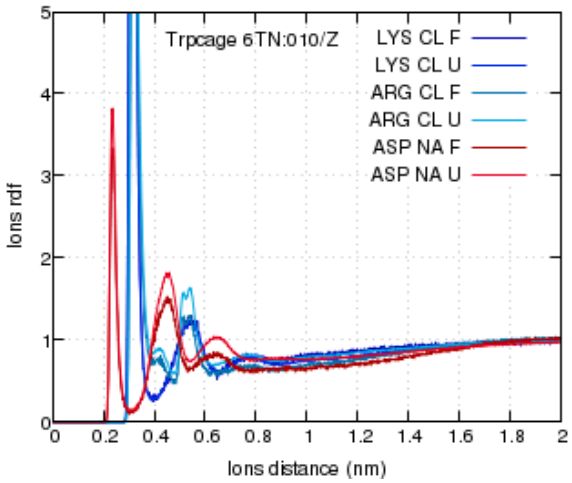

Trpcage ASN:010/F
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

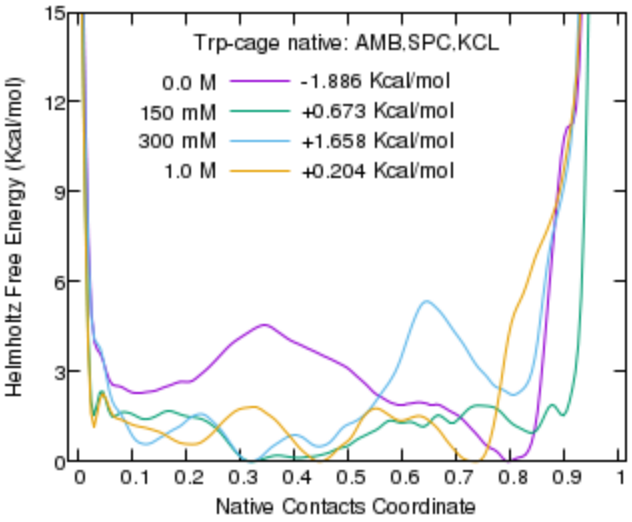

Trp-cage native: AMB.SPC.NACL
0.0 M -1.886 Kcal/mol
150 mM -0.342 Kcal/mol
300 mM -1.420 Kcal/mol
1.0 M -0.857 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

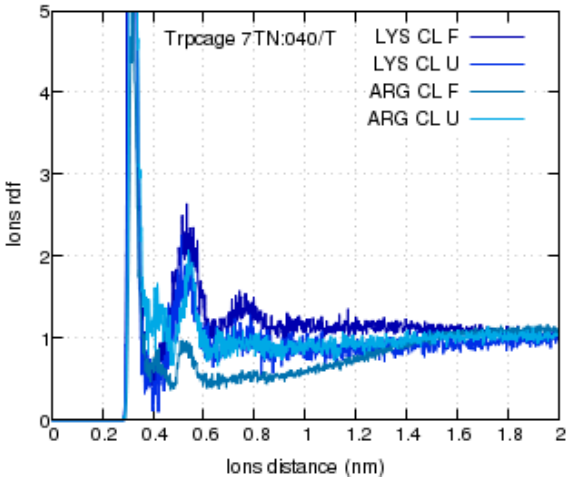

Trpcage 6TN:010/T
LYS CL F
LYS CL U
ARG CL F
ARG CL U
Ions rdf
Ions distance (nm)

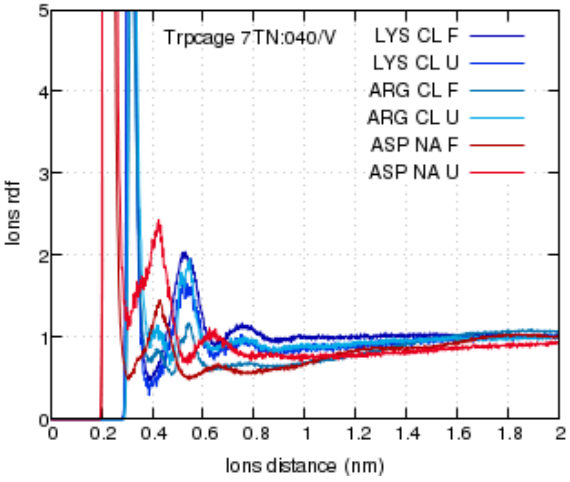

Trpcage 6TN:010/V
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

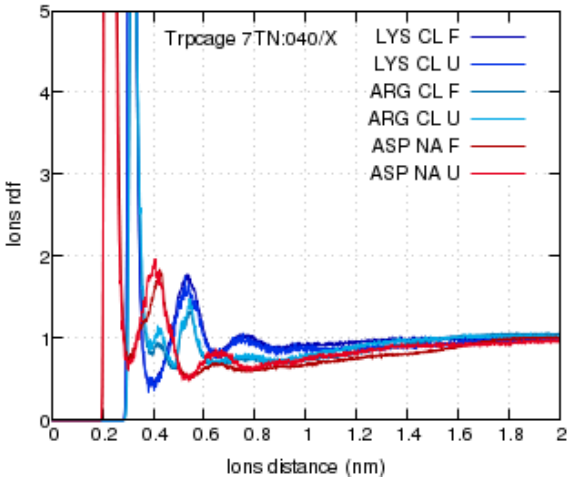

Trpcage 6TN:010/X
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

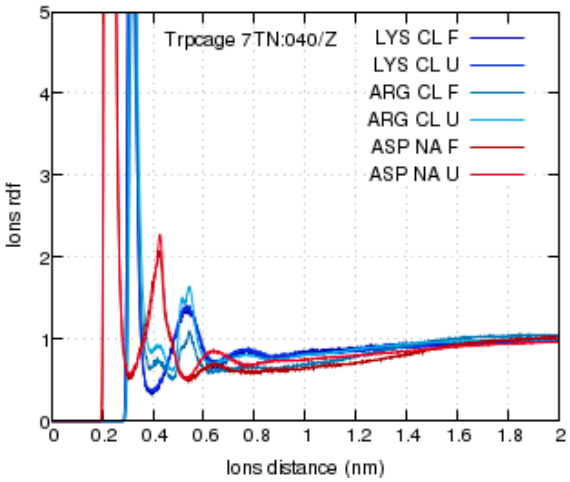

Trpcage 6TN:010/Z
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

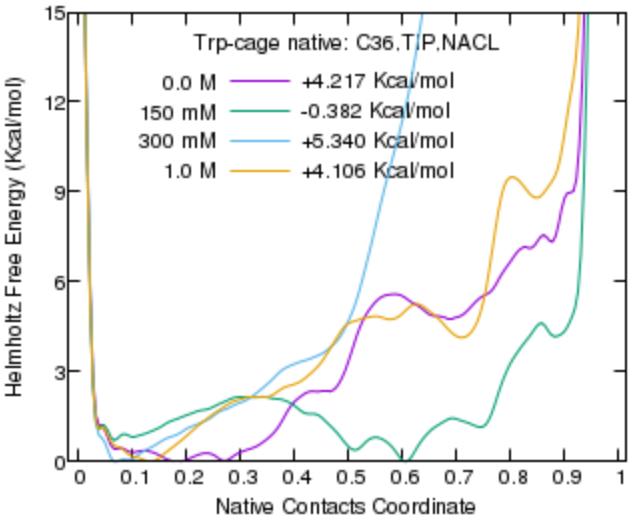

Trp-cage native: AMB.SPC.KCL
0.0 M -1.886 Kcal/mol
150 mM +0.673 Kcal/mol
300 mM +1.658 Kcal/mol
1.0 M +0.204 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

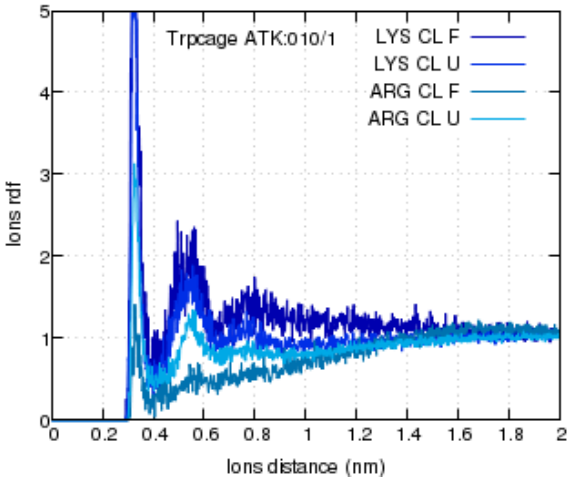

Trpcage 7TN:040/T
LYS CL F
LYS CL U
ARG CL F
ARG CL U
Ions rdf
Ions distance (nm)

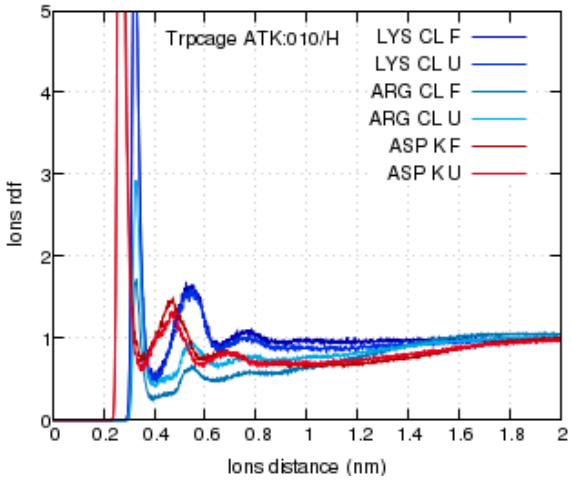

Trpcage 7TN:040/V
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

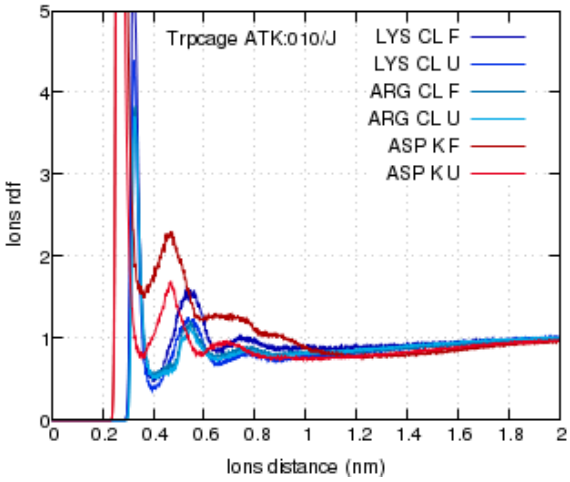

Trpcage 7TN:040/X
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

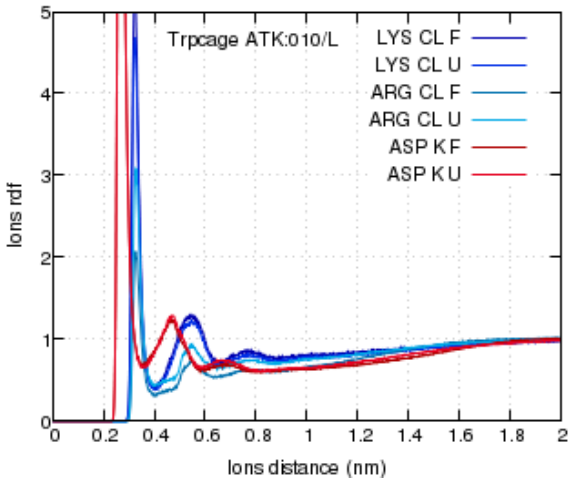

Trpcage 7TN:040/Z
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP NA F
ASP NA U
Ions rdf
Ions distance (nm)

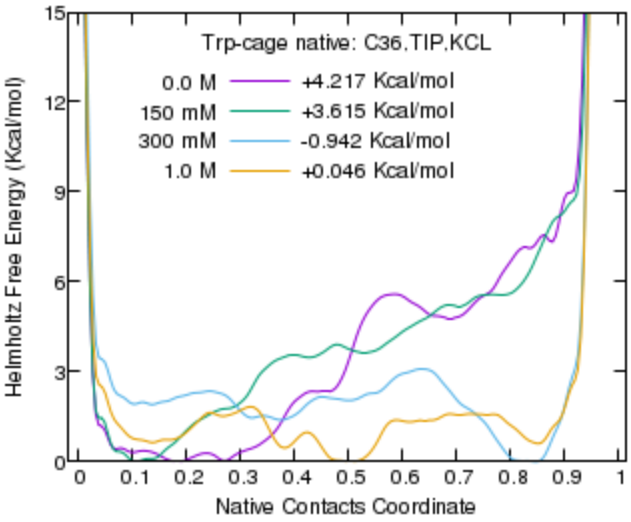

Trp-cage native: C36.T/P.NACL
0.0 M +4.217 Kcal/mol
150 mM -0.382 Kcal/mol
300 mM +5.340 Kcal/mol
1.0 M +4.106 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

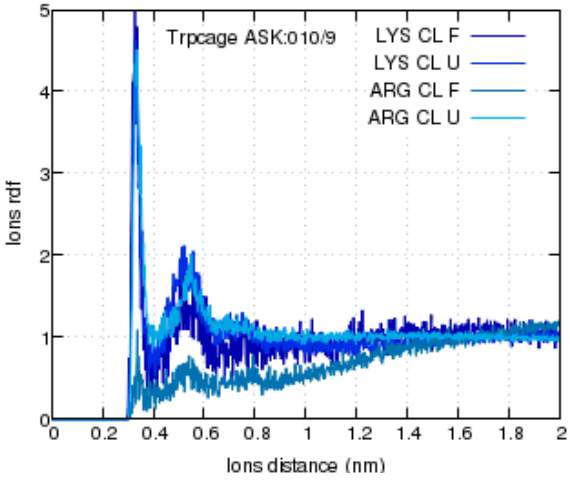

Trpcage ATK:010/1
LYS CL F
LYS CL U
ARG CL F
ARG CL U
Ions rdf
Ions distance (nm)

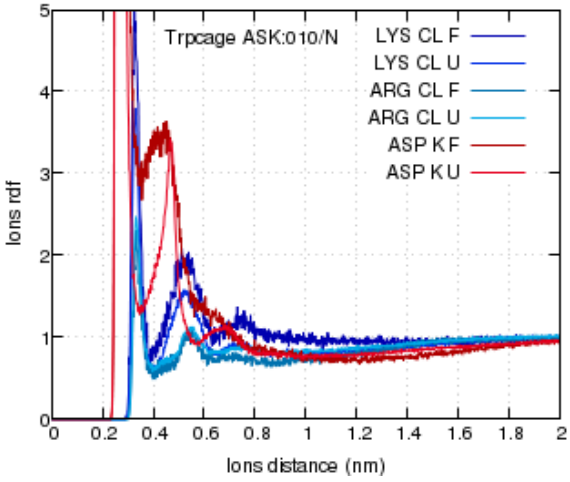

Trpcage ATK:010/H
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

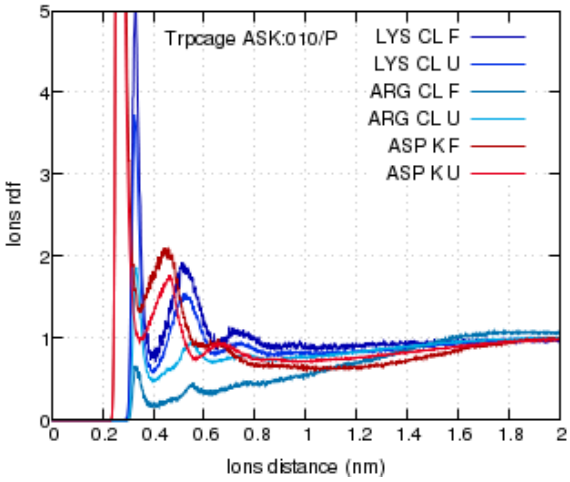

Trpcage ATK:010/J
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

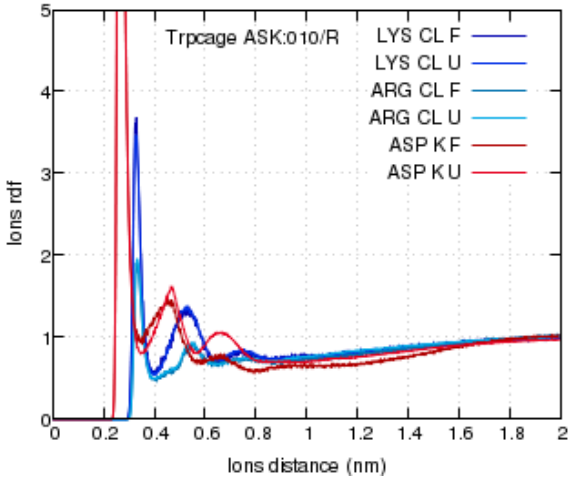

Trpcage ATK:010/L
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

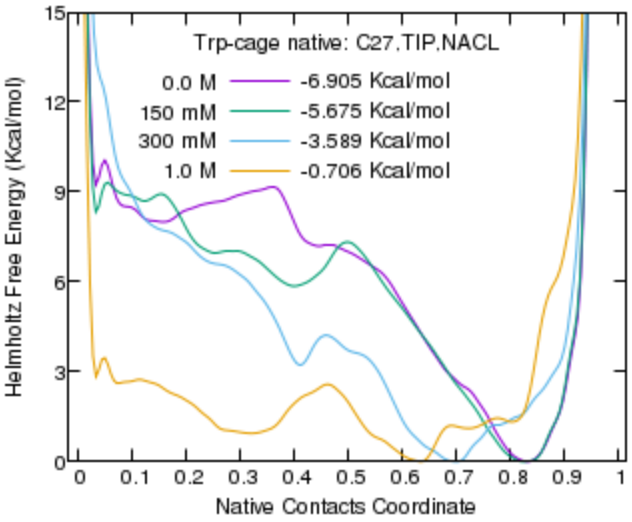

Trp-cage native: C36.TIP.KCL
0.0 M +4.217 Kcal/mol
150 mM +3.615 Kcal/mol
300 mM -0.942 Kcal/mol
1.0 M +0.046 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

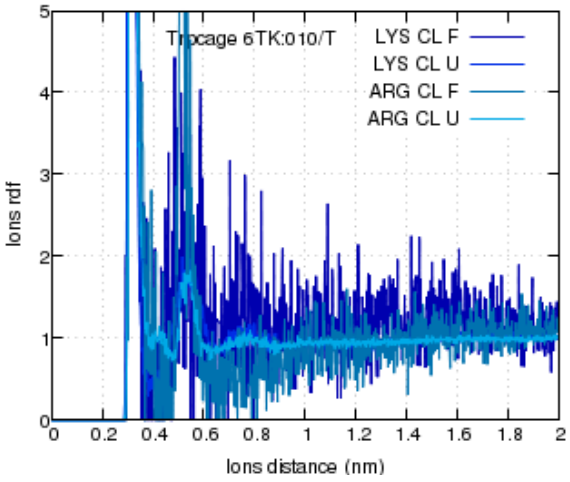

Trpcage ASK:010/9
LYS CL F
LYS CL U
ARG CL F
ARG CL U
Ions rdf
Ions distance (nm)

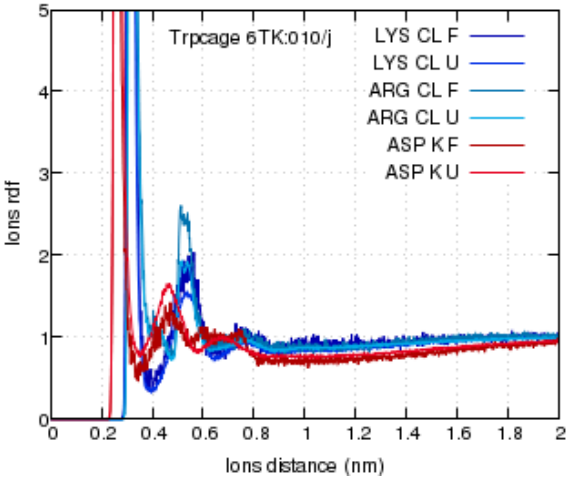

Trpcage ASK:010/N
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

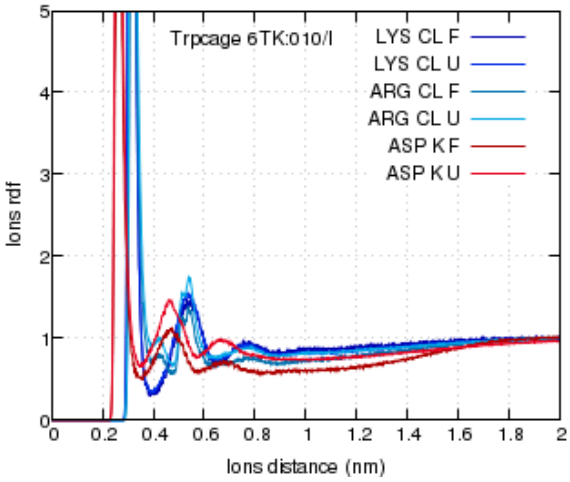

Trpcage ASK:010/P
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

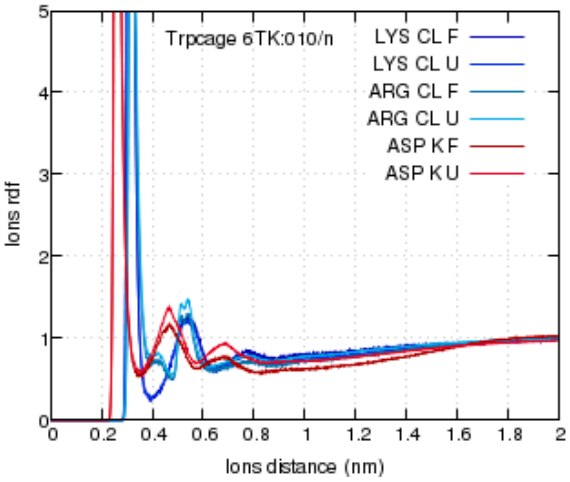

Trpcage ASK:010/R
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

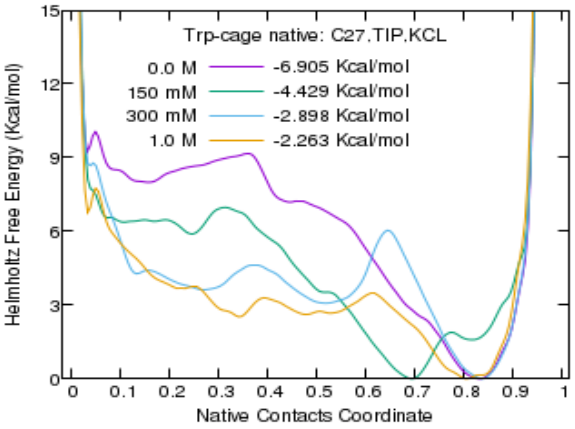

Trp-cage native: C27.TIP.NACL
0.0 M -6.905 Kcal/mol
150 mM -5.675 Kcal/mol
300 mM -3.589 Kcal/mol
1.0 M -0.706 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

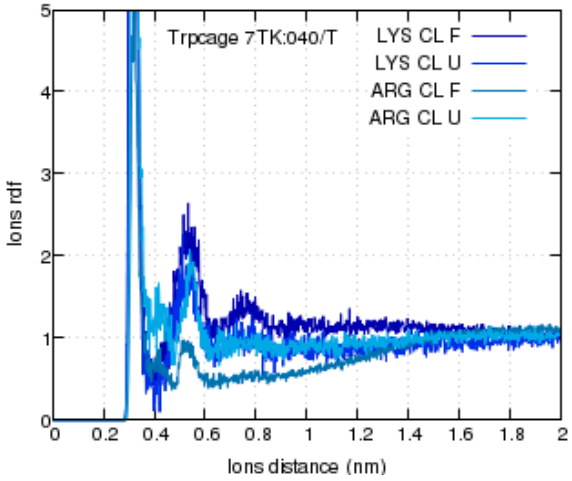

Trpcage 6TK:010/T
LYS CL F
LYS CL U
ARG CL F
ARG CL U
Ions rdf
Ions distance (nm)

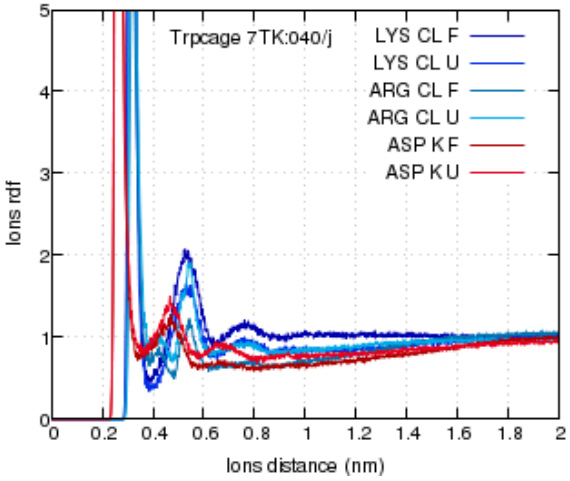

Trpcage 6TK:010/j
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

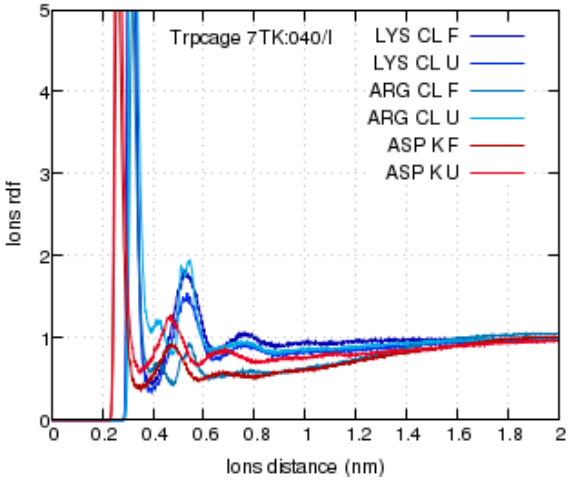

Trpcage 6TK:010/l
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

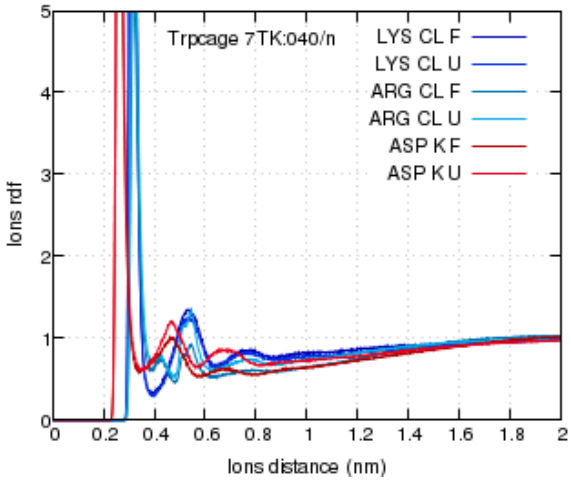

Trpcage 6TK:010/n
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

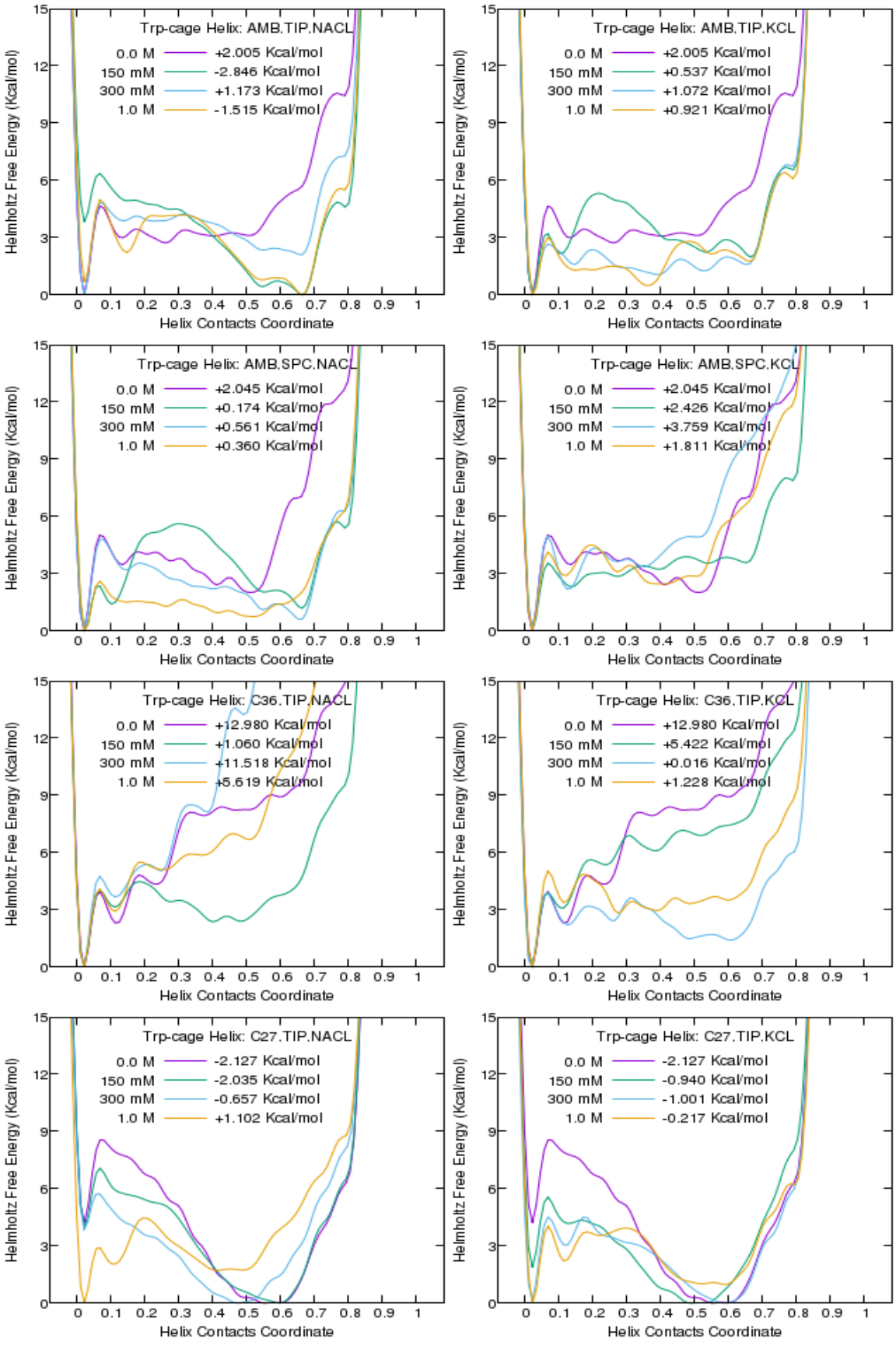

Trp-cage native: C27.TIP.KCL
0.0 M -6.905 Kcal/mol
150 mM -4.429 Kcal/mol
300 mM -2.898 Kcal/mol
1.0 M -2.263 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

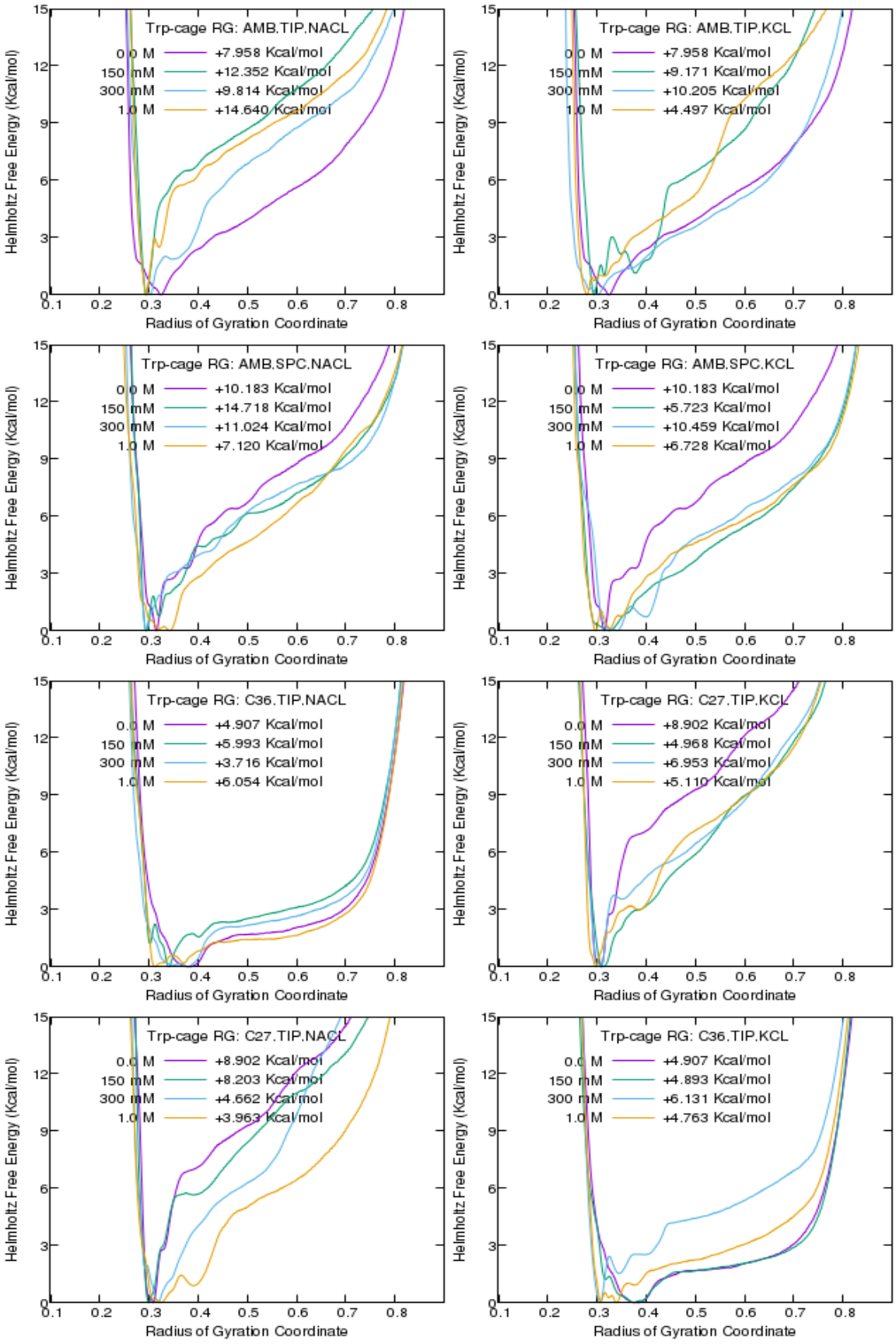

Trpcage 7TK:040/T
LYS CL F
LYS CL U
ARG CL F
ARG CL U
Ions rdf
Ions distance (nm)

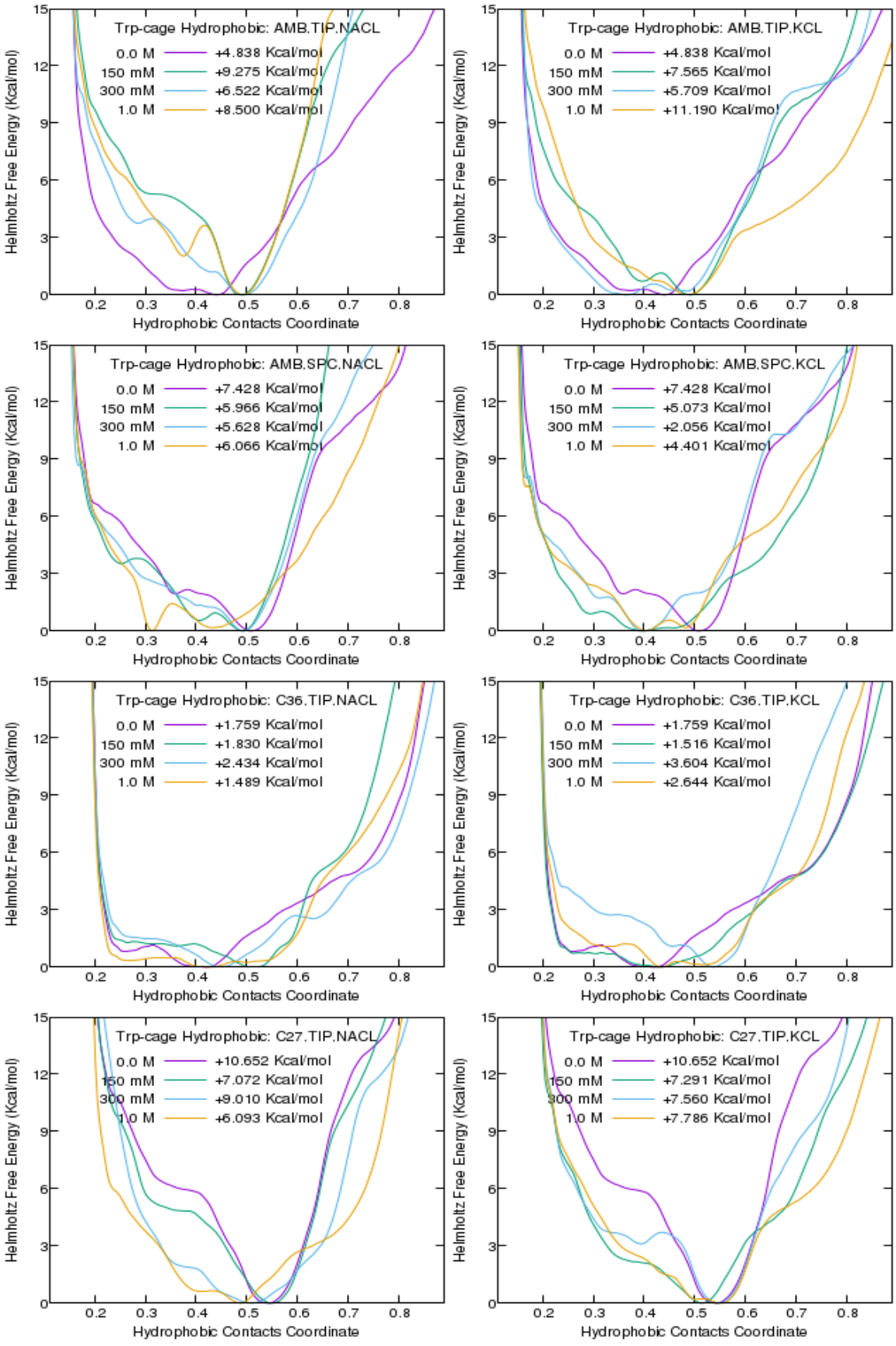

Trpcage 7TK:040/j
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)


Trpcage 7TK:040/l
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

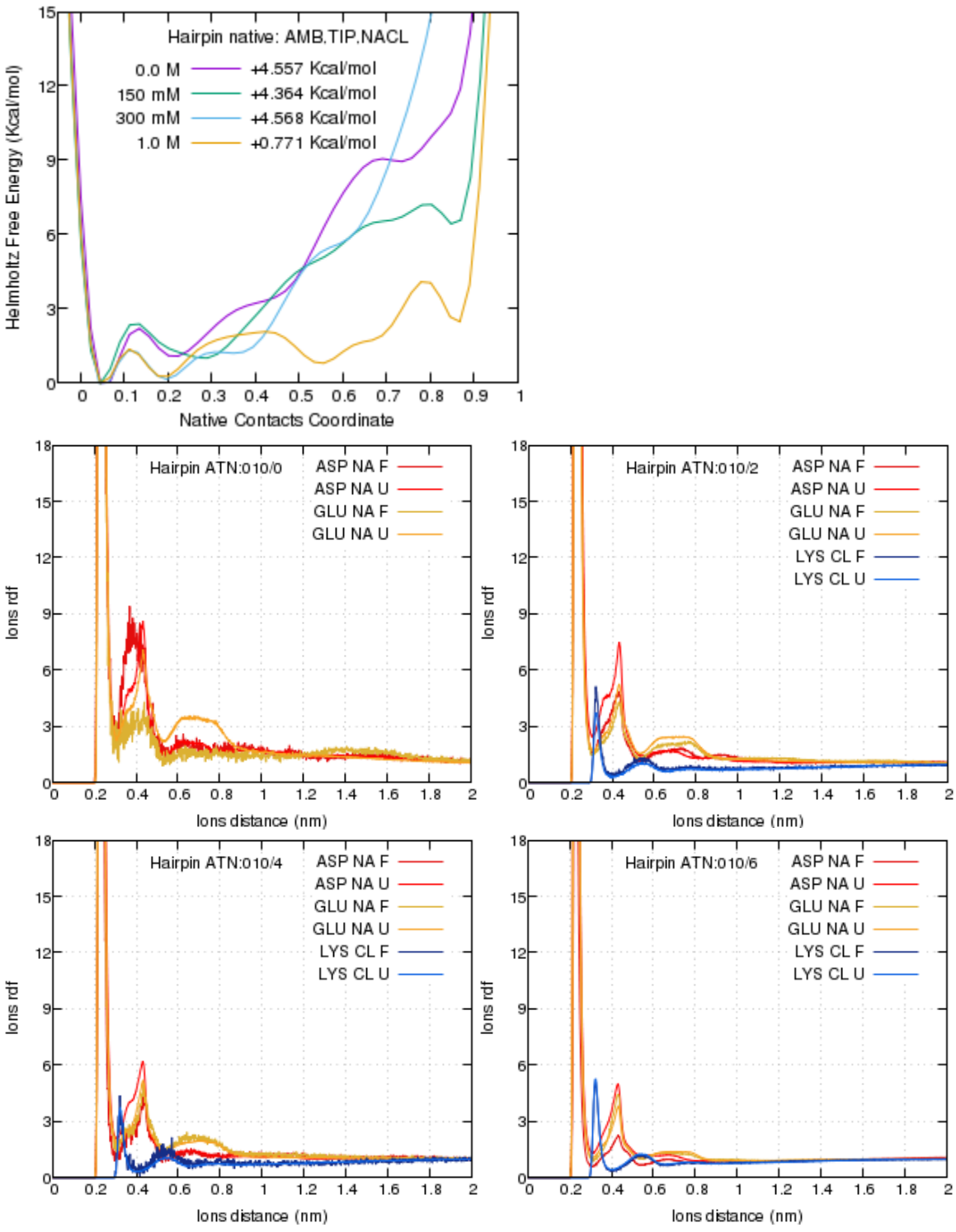

Trpcage 7TK:040/n
LYS CL F
LYS CL U
ARG CL F
ARG CL U
ASP K F
ASP K U
Ions rdf
Ions distance (nm)

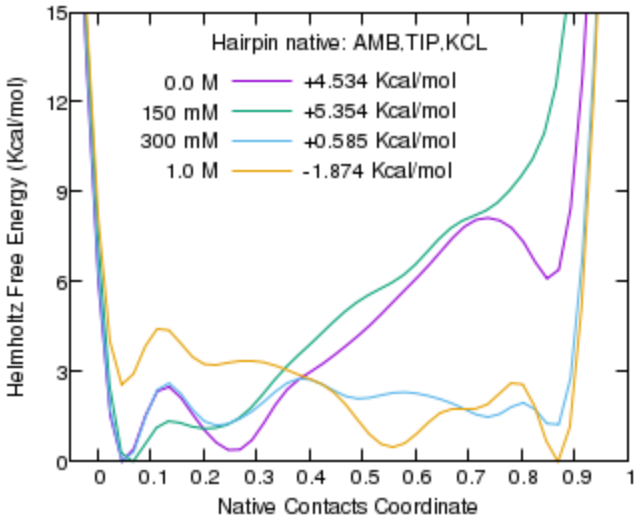

Trp-cage Helix: AMB.TIP.NACL
0.0 M +2.005 Kcal/mol
150 mM -2.846 Kcal/mol
300 mM +1.173 Kcal/mol
1.0 M -1.515 Kcal/mol
Trp-cage Helix: AMB.TIP.KCL
0.0 M +2.005 Kcal/mol
150 mM +0.537 Kcal/mol
300 mM +1.072 Kcal/mol
1.0 M +0.921 Kcal/mol
Trp-cage Helix: AMB.SPC.NACL
0.0 M +2.045 Kcal/mol
150 mM +0.174 Kcal/mol
300 mM +0.561 Kcal/mol
1.0 M +0.360 Kcal/mol
Trp-cage Helix: AMB.SPC.KCL
0.0 M +2.045 Kcal/mol
150 mM +2.426 Kcal/mol
300 mM +3.759 Kcal/mol
1.0 M +1.811 Kcal/mol
Trp-cage Helix: C36.TIP.NACL
0.0 M +12.980 Kcal/mol
150 mM +1.060 Kcal/mol
300 mM +11.518 Kcal/mol
1.0 M +5.619 Kcal/mol
Trp-cage Helix: C36.TIP.KCL
0.0 M +12.980 Kcal/mol
150 mM +5.422 Kcal/mol
300 mM +0.016 Kcal/mol
1.0 M +1.228 Kcal/mol
Trp-cage Helix: C27.TIP.NACL
0.0 M -2.127 Kcal/mol
150 mM -2.035 Kcal/mol
300 mM -0.657 Kcal/mol
1.0 M +1.102 Kcal/mol
Trp-cage Helix: C27.TIP.KCL
0.0 M -2.127 Kcal/mol
150 mM -0.940 Kcal/mol
300 mM -1.001 Kcal/mol
1.0 M -0.217 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Helix Contacts Coordinate

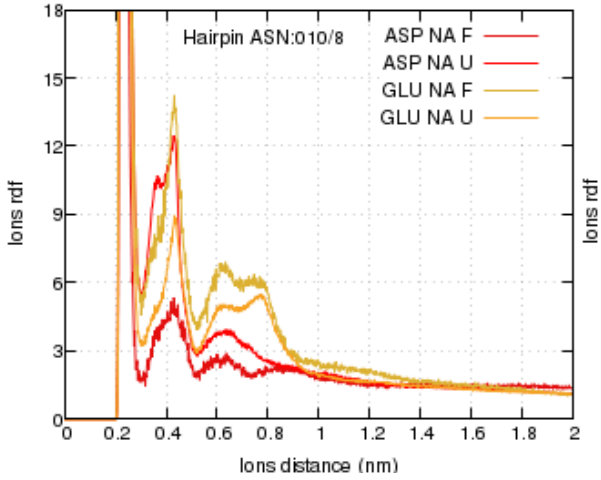

Trp-cage RG: AMB.TIP.NACL
0.0 M +7.958 Kcal/mol
150 mM +12.352 Kcal/mol
300 mM +9.814 Kcal/mol
1.0 M +14.640 Kcal/mol
Trp-cage RG: AMB.TIP.KCL
0.0 M +7.958 Kcal/mol
150 mM +9.171 Kcal/mol
300 mM +10.205 Kcal/mol
1.0 M +4.497 Kcal/mol
Trp-cage RG: AMB.SPC.NACL
0.0 M +10.183 Kcal/mol
150 mM +14.718 Kcal/mol
300 mM +11.024 Kcal/mol
1.0 M +7.120 Kcal/mol
Trp-cage RG: AMB.SPC.KCL
0.0 M +10.183 Kcal/mol
150 mM +5.723 Kcal/mol
300 mM +10.459 Kcal/mol
1.0 M +6.728 Kcal/mol
Trp-cage RG: C36.TIP.NACL
0.0 M +4.907 Kcal/mol
150 mM +5.993 Kcal/mol
300 mM +3.716 Kcal/mol
1.0 M +6.054 Kcal/mol
Trp-cage RG: C27.TIP.KCL
0.0 M +8.902 Kcal/mol
150 mM +4.968 Kcal/mol
300 mM +6.953 Kcal/mol
1.0 M +5.110 Kcal/mol
Trp-cage RG: C27.TIP.NACL
0.0 M +8.902 Kcal/mol
150 mM +8.203 Kcal/mol
300 mM +4.662 Kcal/mol
1.0 M +3.963 Kcal/mol
Trp-cage RG: C36.TIP.KCL
0.0 M +4.907 Kcal/mol
150 mM +4.893 Kcal/mol
300 mM +6.131 Kcal/mol
1.0 M +4.763 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Radius of Gyration Coordinate
0 3 6 9 12 15
0.1 0.2 0.3 0.4 0.5 0.6 0.7 0.8

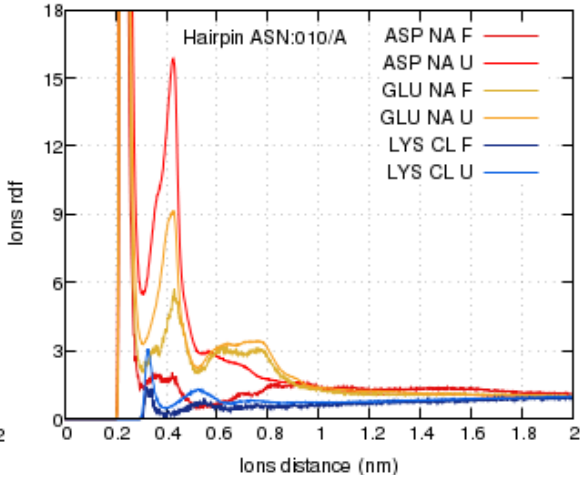

Trp-cage Hydrophobic: AMB.TIP.NACL
0.0 M +4.838 Kcal/mol
150 mM +9.275 Kcal/mol
300 mM +6.522 Kcal/mol
1.0 M +8.500 Kcal/mol
Trp-cage Hydrophobic: AMB.TIP.KCL
0.0 M +4.838 Kcal/mol
150 mM +7.565 Kcal/mol
300 mM +5.709 Kcal/mol
1.0 M +11.190 Kcal/mol
Trp-cage Hydrophobic: AMB.SPC.NACL
0.0 M +7.428 Kcal/mol
150 mM +5.966 Kcal/mol
300 mM +5.628 Kcal/mol
1.0 M +6.066 Kcal/mol
Trp-cage Hydrophobic: AMB.SPC.KCL
0.0 M +7.428 Kcal/mol
150 mM +5.073 Kcal/mol
300 mM +2.056 Kcal/mol
1.0 M +4.401 Kcal/mol
Trp-cage Hydrophobic: C36.TIP.NACL
0.0 M +1.759 Kcal/mol
150 mM +1.830 Kcal/mol
300 mM +2.434 Kcal/mol
1.0 M +1.489 Kcal/mol
Trp-cage Hydrophobic: C36.TIP.KCL
0.0 M +1.759 Kcal/mol
150 mM +1.516 Kcal/mol
300 mM +3.604 Kcal/mol
1.0 M +2.644 Kcal/mol
Trp-cage Hydrophobic: C27.TIP.NACL
0.0 M +10.652 Kcal/mol
150 mM +7.072 Kcal/mol
300 mM +9.010 Kcal/mol
1.0 M +6.093 Kcal/mol
Trp-cage Hydrophobic: C27.TIP.KCL
0.0 M +10.652 Kcal/mol
150 mM +7.291 Kcal/mol
300 mM +7.560 Kcal/mol
1.0 M +7.786 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Hydrophobic Contacts Coordinate
0
3
6
9
12
15
0.2
0.3
0.4
0.5
0.6
0.7
0.8

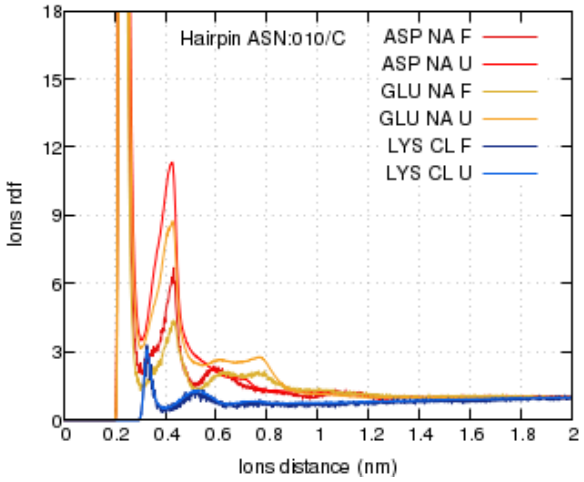
GB1 (Hairpin)

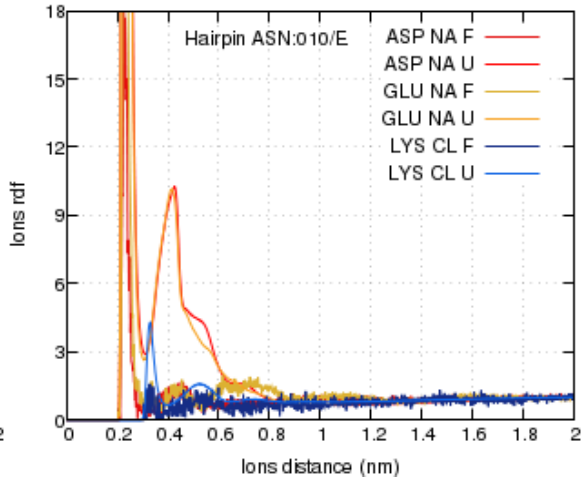

Hairpin native: AMB.TIP.NACL
0.0 M +4.557 Kcal/mol
150 mM +4.364 Kcal/mol
300 mM +4.568 Kcal/mol
1.0 M +0.771 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate
Hairpin ATN:010/0
Hairpin ATN:010/2
Hairpin ATN:010/4
Hairpin ATN:010/6
ASP NA F
ASP NA U
GLU NA F
GLU NA U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

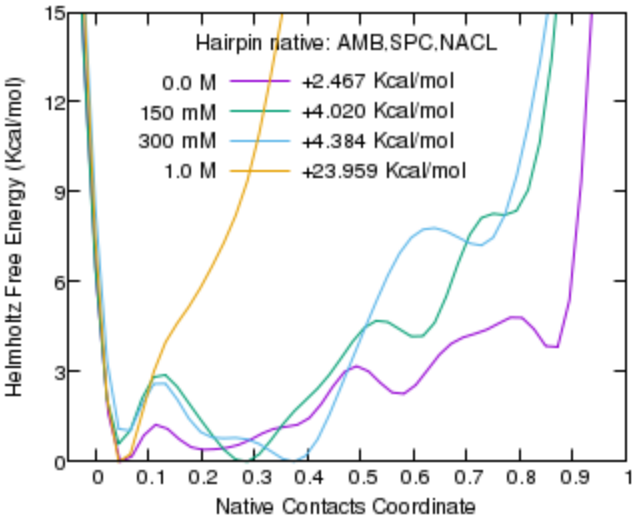

Hairpin native: AMB.TIP.KCL
0.0 M +4.534 Kcal/mol
150 mM +5.354 Kcal/mol
300 mM +0.585 Kcal/mol
1.0 M -1.874 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

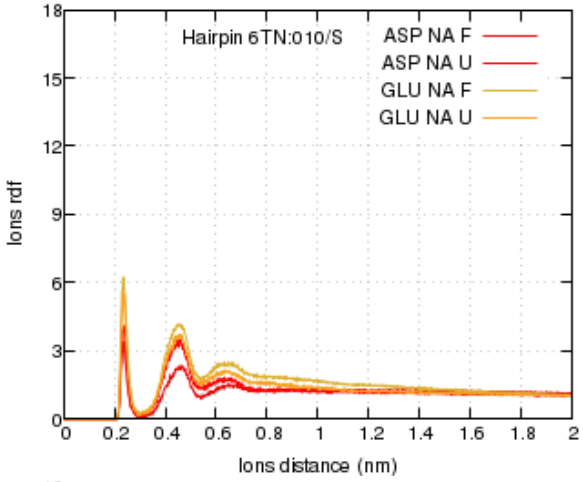

Hairpin ASN:010/8
ASP NA F
ASP NA U
GLU NA F
GLU NA U
Ions rdf
Ions distance (nm)

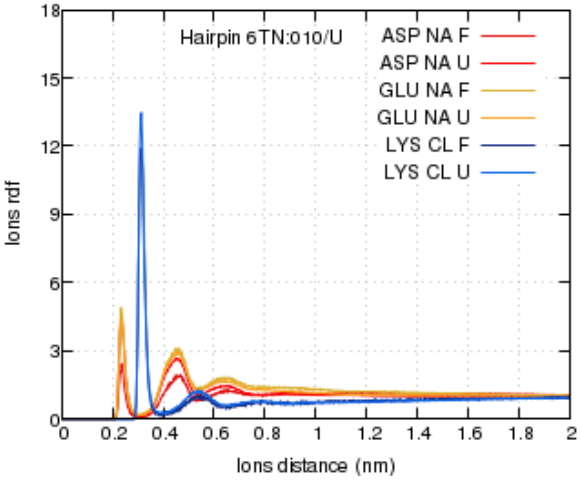

Hairpin ASN:010/A
ASP NA F
ASP NA U
GLU NA F
GLU NA U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

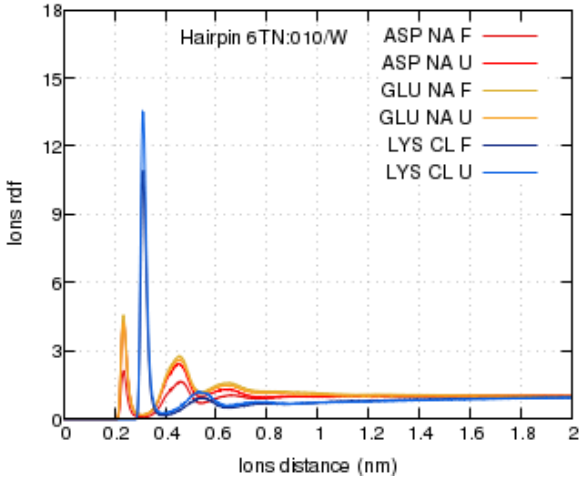

Hairpin ASN:010/C
ASP NA F
ASP NA U
GLU NA F
GLU NA U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

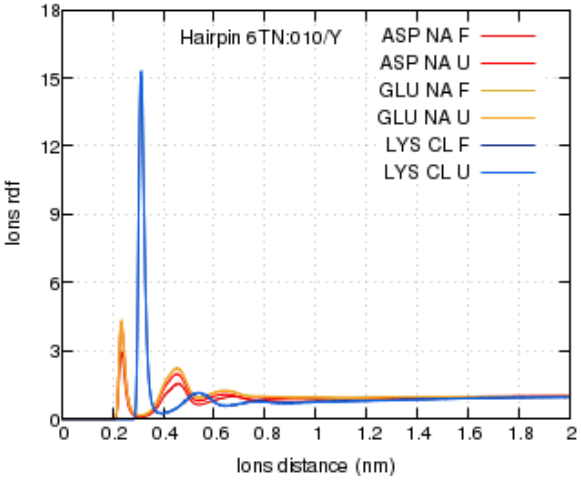

Hairpin ASN:010/E
ASP NA F
ASP NA U
GLU NA F
GLU NA U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

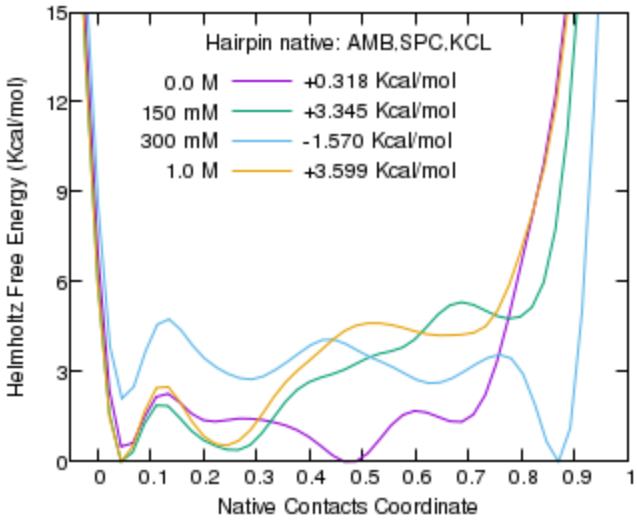

Hairpin native: AMB.SPC.NACL
0.0 M +2.467 Kcal/mol
150 mM +4.020 Kcal/mol
300 mM +4.384 Kcal/mol
1.0 M +23.959 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

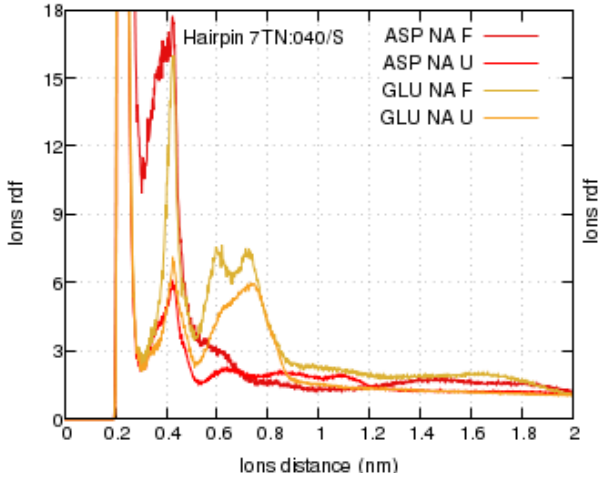

Hairpin 6TN:010/S
ASP NA F
ASP NA U
GLU NA F
GLU NA U
Ions rdf
Ions distance (nm)

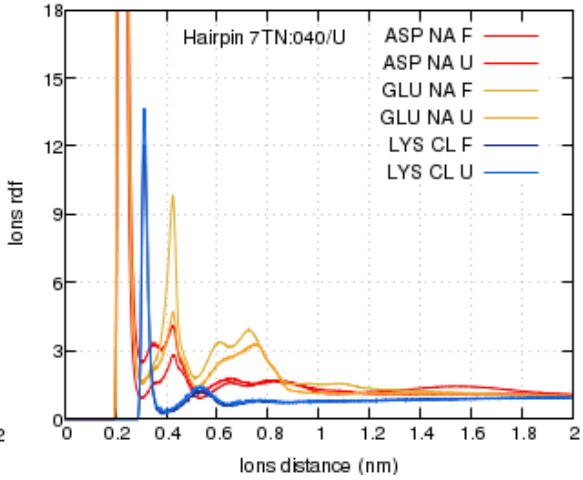

Hairpin 6TN:010/U
ASP NA F
ASP NA U
GLU NA F
GLU NA U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

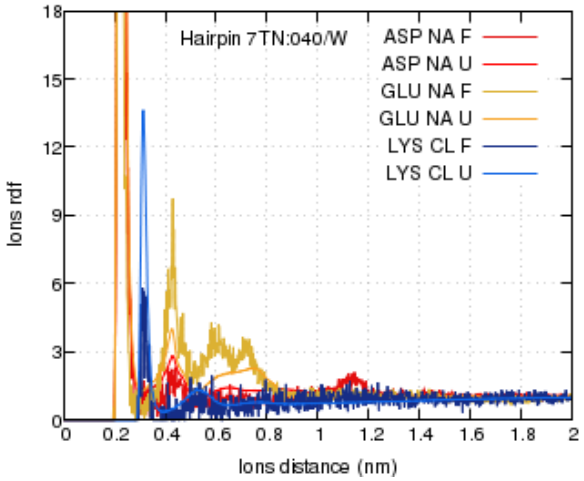

Hairpin 6TN:010/W
ASP NA F
ASP NA U
GLU NA F
GLU NA U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

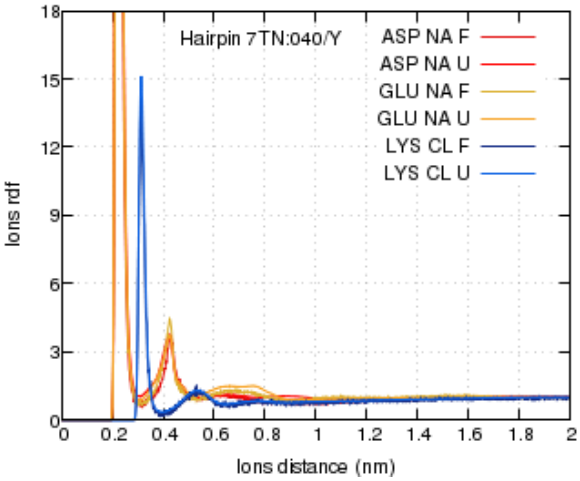

Hairpin 6TN:010/Y
ASP NA F
ASP NA U
GLU NA F
GLU NA U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

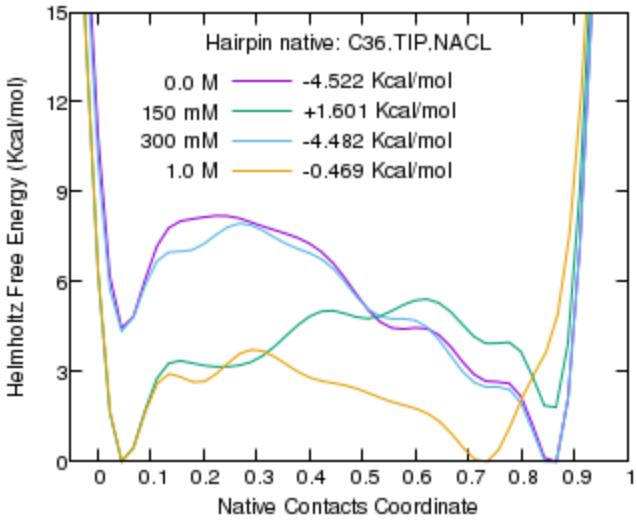

Hairpin native: AMB.SPC.KCL
0.0 M +0.318 Kcal/mol
150 mM +3.345 Kcal/mol
300 mM -1.570 Kcal/mol
1.0 M +3.599 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

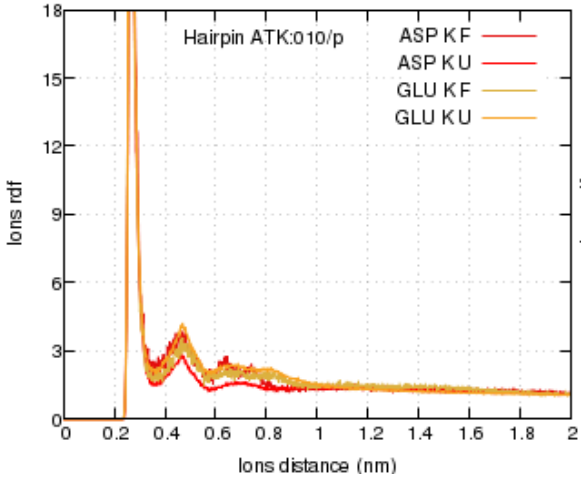

Hairpin 7TN:040/S
ASP NA F
ASP NA U
GLU NA F
GLU NA U
Ions rdf
Ions distance (nm)

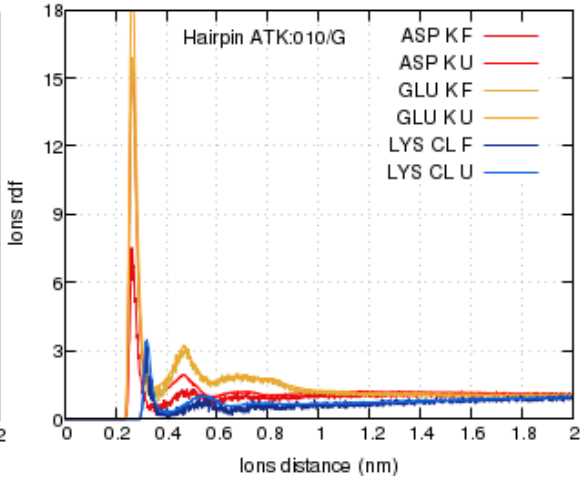

Hairpin 7TN:040/U
ASP NA F
ASP NA U
GLU NA F
GLU NA U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

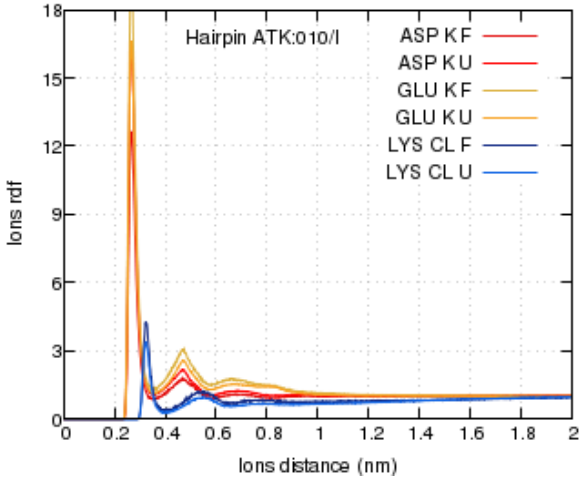

Hairpin 7TN:040/W
ASP NA F
ASP NA U
GLU NA F
GLU NA U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

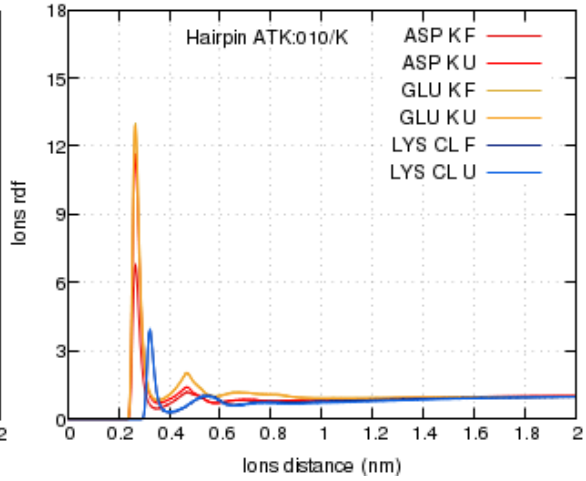

Hairpin 7TN:040/Y
ASP NA F
ASP NA U
GLU NA F
GLU NA U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

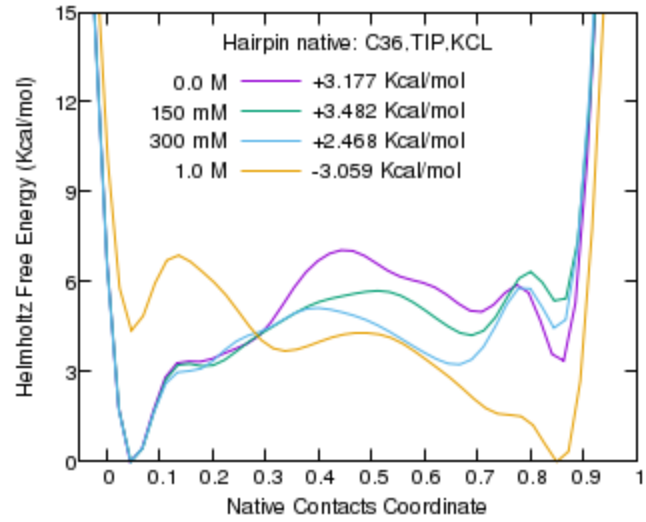

Hairpin native: C36.TIP.NACL
0.0 M -4.522 Kcal/mol
150 mM +1.601 Kcal/mol
300 mM -4.482 Kcal/mol
1.0 M -0.469 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

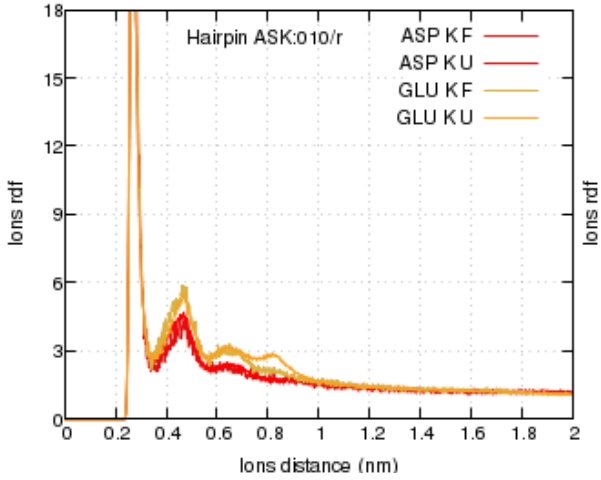

Hairpin ATK:010/p
ASP K F
ASP K U
GLU K F
GLU K U
Ions rdf
Ions distance (nm)

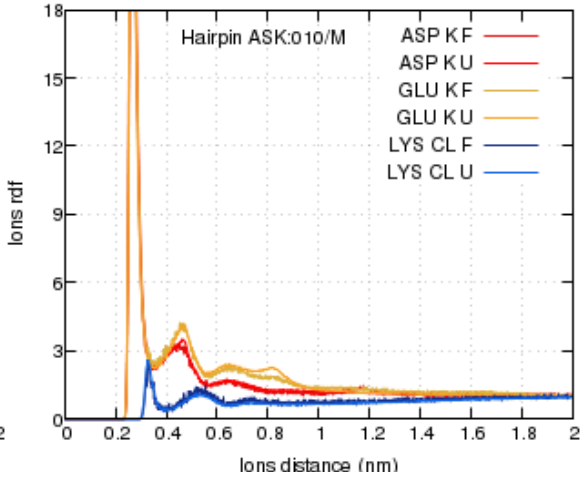

Hairpin ATK:010/G
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

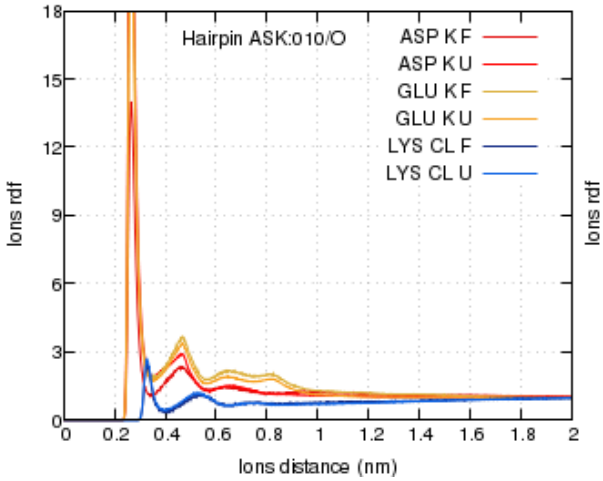

Hairpin ATK:010/I
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

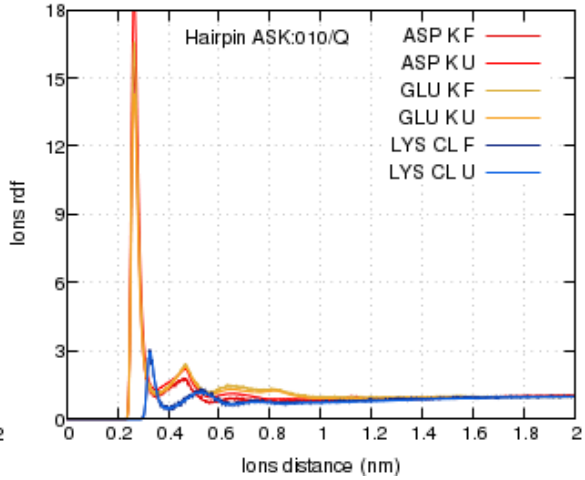

Hairpin ATK:010/K
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

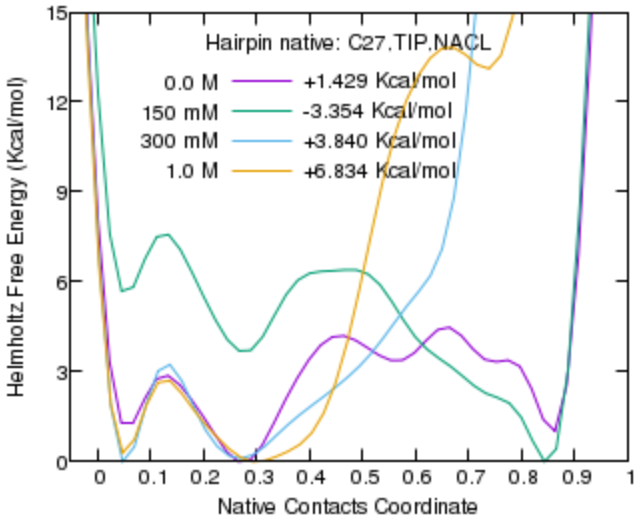

Hairpin native: C36.TIP.KCL
0.0 M +3.177 Kcal/mol
150 mM +3.482 Kcal/mol
300 mM +2.468 Kcal/mol
1.0 M -3.059 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

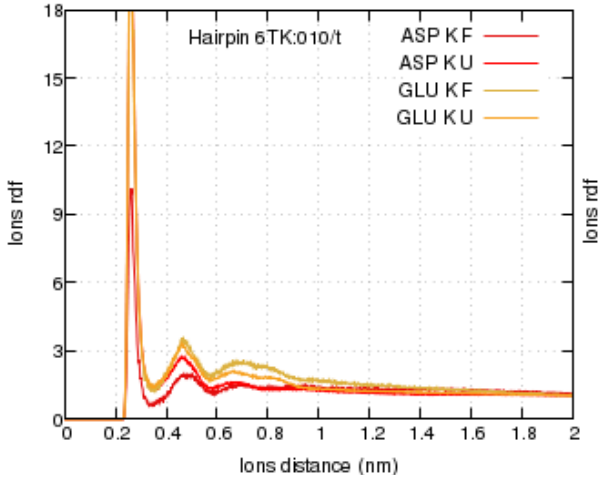

Hairpin ASK:010/r
ASP K F
ASP K U
GLU K F
GLU K U
Ions rdf
Ions distance (nm)

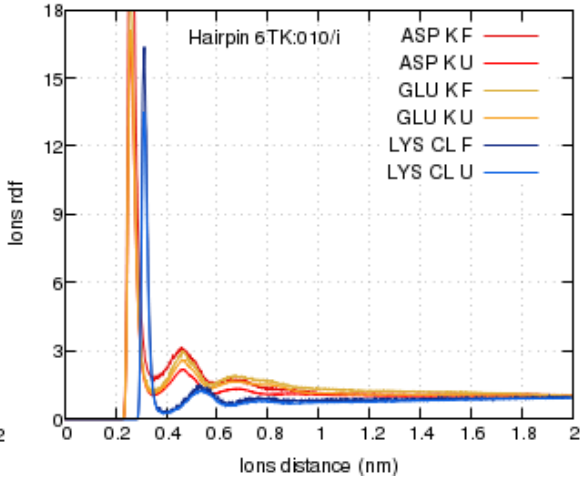

Hairpin ASK:010/M
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

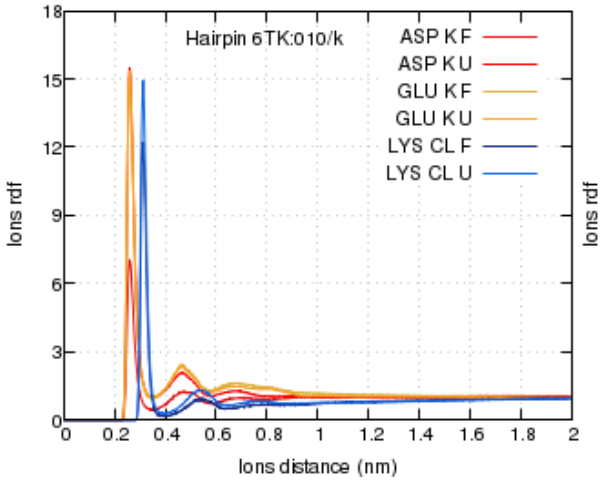

Hairpin ASK:010/O
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

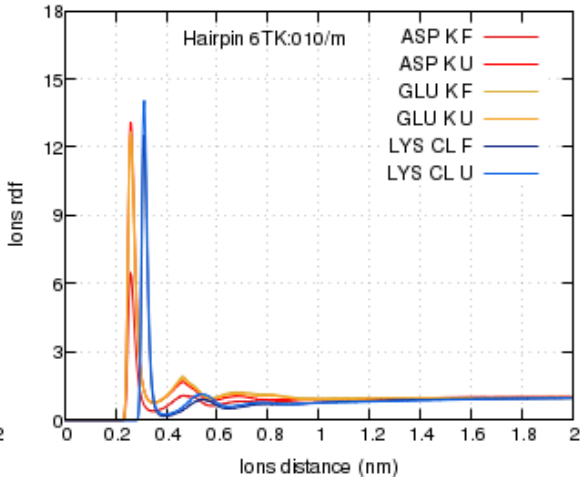

Hairpin ASK:010/Q
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

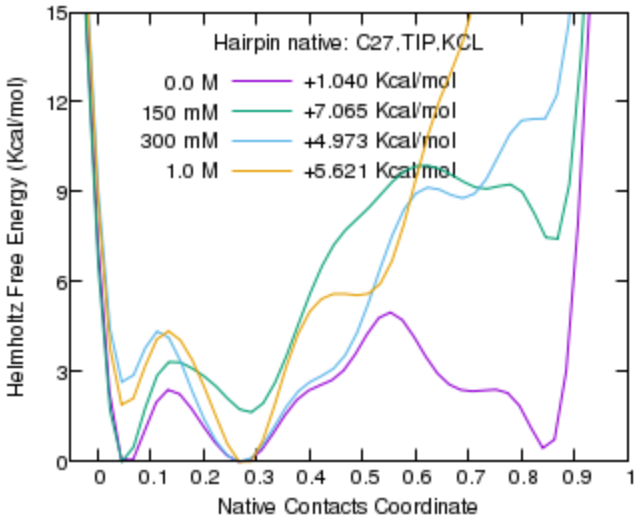

Hairpin native: C27.TIP.NACL
0.0 M +1.429 Kcal/mol
150 mM -3.354 Kcal/mol
300 mM +3.840 Kcal/mol
1.0 M +6.834 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

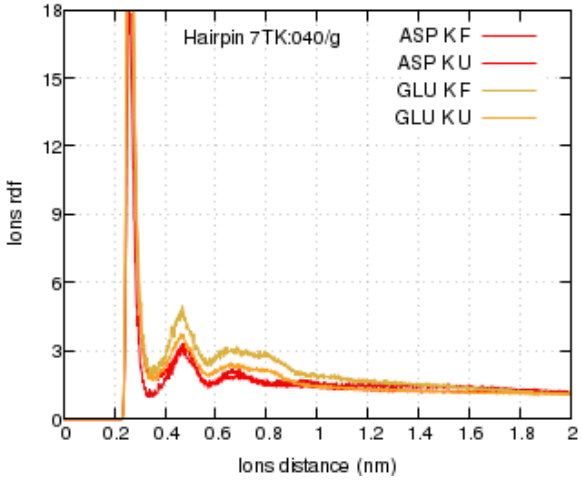

Hairpin 6TK:010/t
ASP K F
ASP K U
GLU K F
GLU K U
Ions rdf
Ions distance (nm)

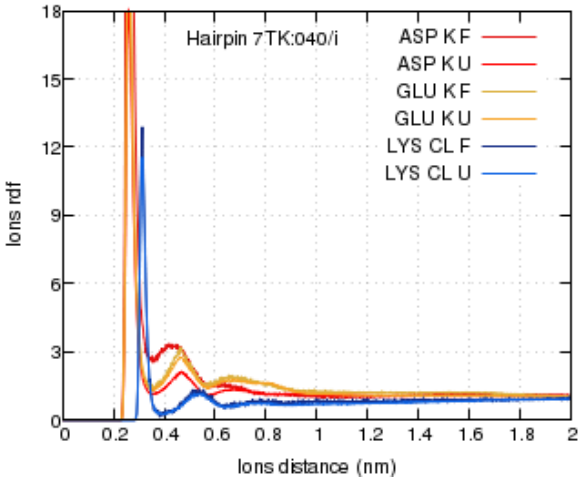

Hairpin 6TK:010/i
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

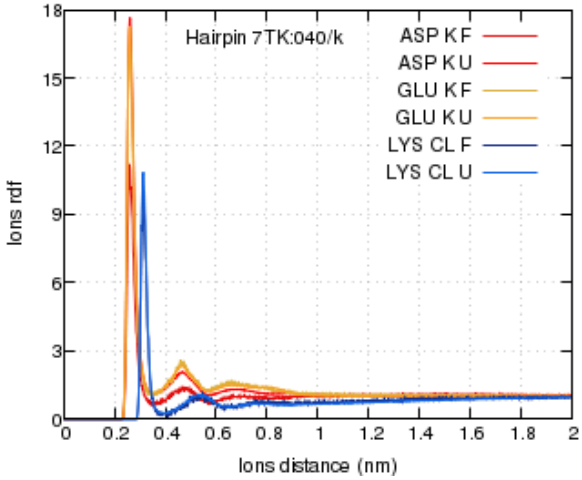

Hairpin 6TK:010/k
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

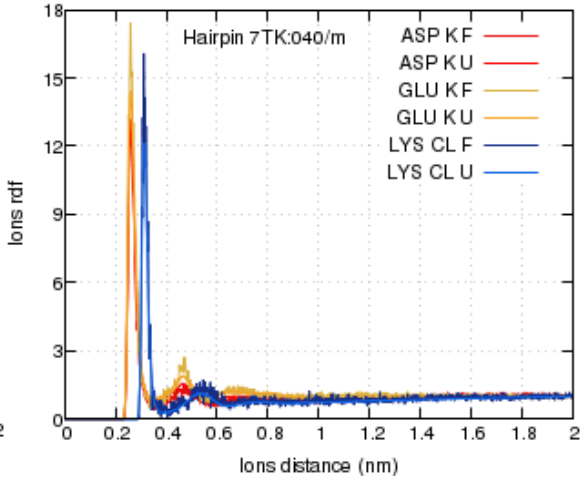

Hairpin 6TK:010/m
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)

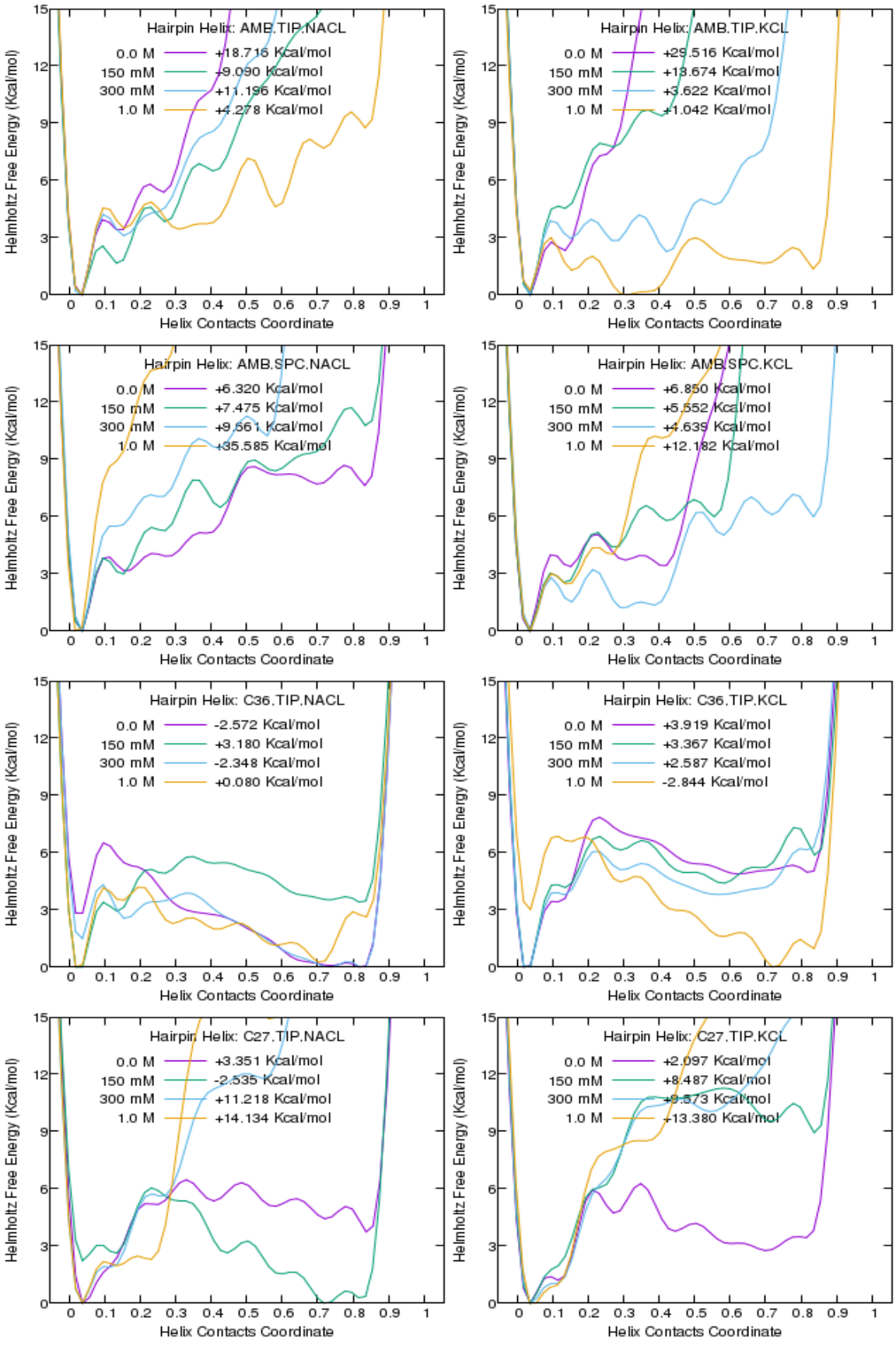

Hairpin native: C27.TIP.KCL
0.0 M +1.040 Kcal/mol
150 mM +7.065 Kcal/mol
300 mM +4.973 Kcal/mol
1.0 M +5.621 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Native Contacts Coordinate

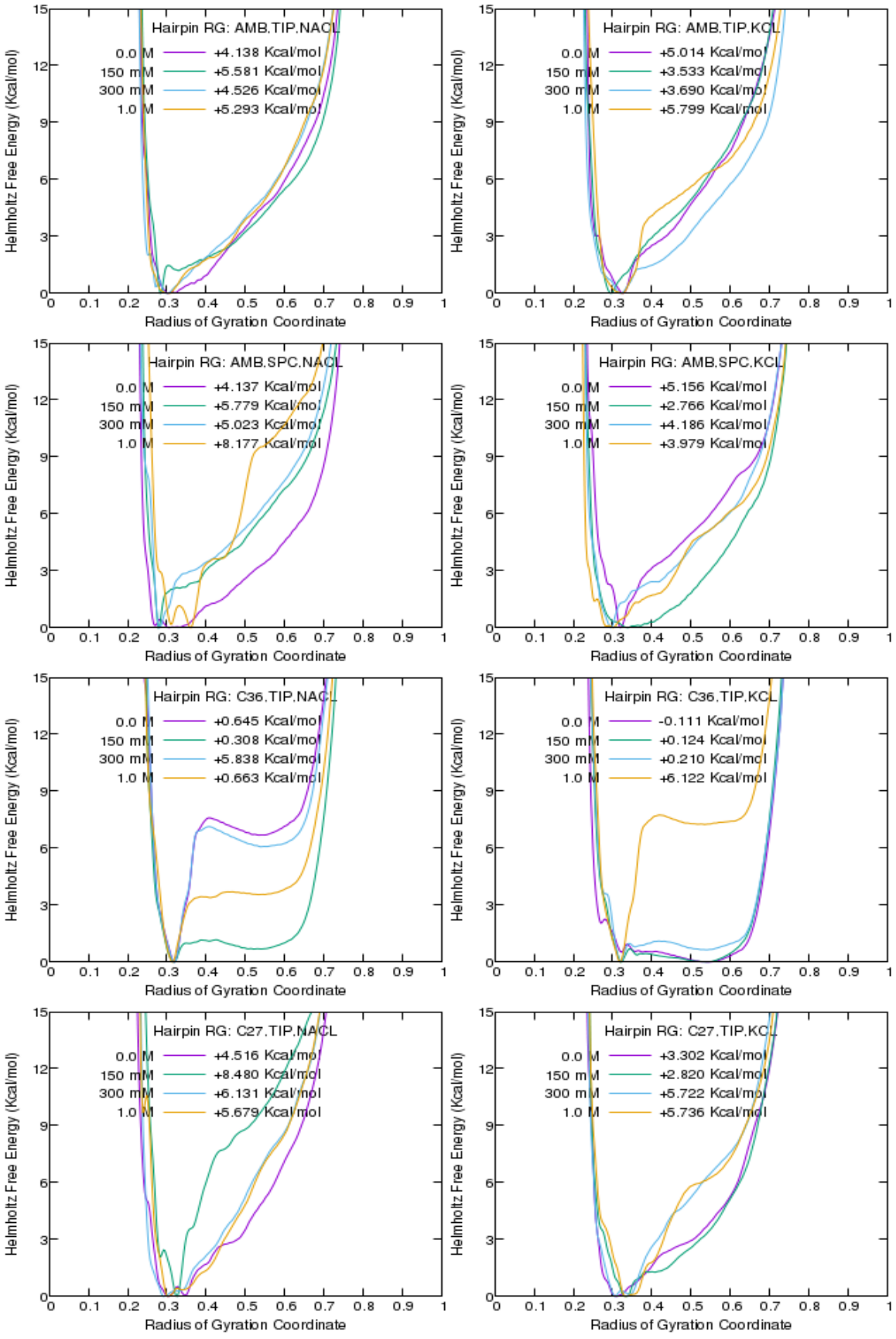

Hairpin 7TK:040/g
ASP K F
ASP K U
GLU K F
GLU K U
Ions rdf
Ions distance (nm)

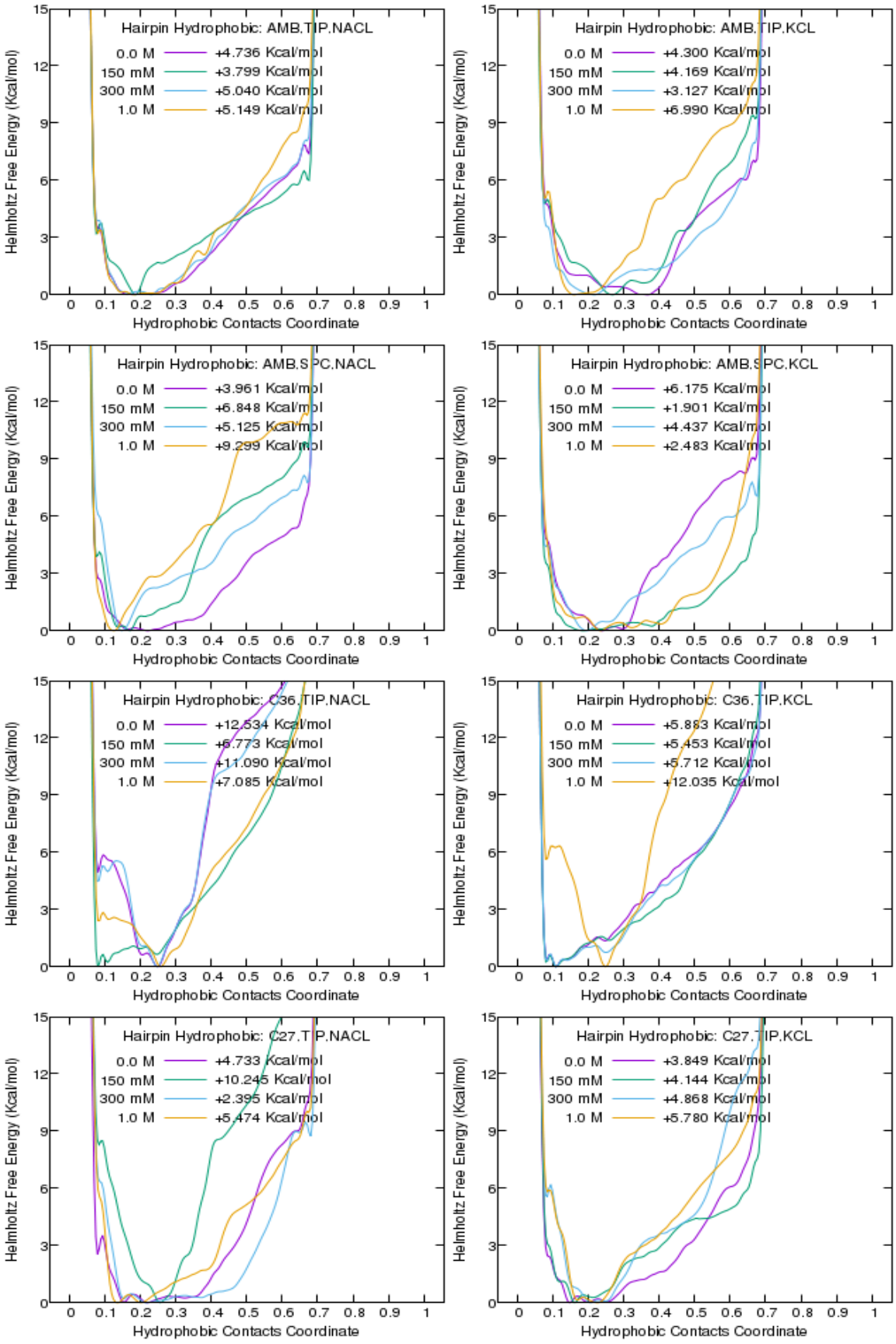

Hairpin 7TK:040/i
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)


Hairpin 7TK:040/k
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)


Hairpin 7TK:040/m
ASP K F
ASP K U
GLU K F
GLU K U
LYS CL F
LYS CL U
Ions rdf
Ions distance (nm)


Hairpin Helix: AMB.TIP.NACL
0.0 M +18.716 Kcal/mol
150 mM +9.090 Kcal/mol
300 mM +11.196 Kcal/mol
1.0 M +4.278 Kcal/mol
Hairpin Helix: AMB.TIP.KCL
0.0 M +29.516 Kcal/mol
150 mM +13.674 Kcal/mol
300 mM +3.622 Kcal/mol
1.0 M +1.042 Kcal/mol
Hairpin Helix: AMB.SPC.NACL
0.0 M +6.320 Kcal/mol
150 mM +7.475 Kcal/mol
300 mM +9.561 Kcal/mol
1.0 M +35.585 Kcal/mol
Hairpin Helix: AMB.SPC.KCL
0.0 M +6.850 Kcal/mol
150 mM +5.552 Kcal/mol
300 mM +4.639 Kcal/mol
1.0 M +12.182 Kcal/mol
Hairpin Helix: C36.TIP.NACL
0.0 M -2.572 Kcal/mol
150 mM +3.180 Kcal/mol
300 mM -2.348 Kcal/mol
1.0 M +0.080 Kcal/mol
Hairpin Helix: C36.TIP.KCL
0.0 M +3.919 Kcal/mol
150 mM +3.367 Kcal/mol
300 mM +2.587 Kcal/mol
1.0 M -2.844 Kcal/mol
Hairpin Helix: C27.TIP.NACL
0.0 M +3.351 Kcal/mol
150 mM -2.535 Kcal/mol
300 mM +11.218 Kcal/mol
1.0 M +14.134 Kcal/mol
Hairpin Helix: C27.TIP.KCL
0.0 M +2.097 Kcal/mol
150 mM +8.487 Kcal/mol
300 mM +9.573 Kcal/mol
1.0 M +13.380 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Helix Contacts Coordinate


Hairpin RG: AMB.TIP.NACL
0.0 M +4.138 Kcal/mol
150 mM +5.581 Kcal/mol
300 mM +4.526 Kcal/mol
1.0 M +5.293 Kcal/mol
Hairpin RG: AMB.TIP.KCL
0.0 M +5.014 Kcal/mol
150 mM +3.533 Kcal/mol
300 mM +3.690 Kcal/mol
1.0 M +5.799 Kcal/mol
Hairpin RG: AMB.SPC.NACL
0.0 M +4.137 Kcal/mol
150 mM +5.779 Kcal/mol
300 mM +5.023 Kcal/mol
1.0 M +8.177 Kcal/mol
Hairpin RG: AMB.SPC.KCL
0.0 M +5.156 Kcal/mol
150 mM +2.766 Kcal/mol
300 mM +4.186 Kcal/mol
1.0 M +3.979 Kcal/mol
Hairpin RG: C36.TIP.NACL
0.0 M +0.645 Kcal/mol
150 mM +0.308 Kcal/mol
300 mM +5.838 Kcal/mol
1.0 M +0.663 Kcal/mol
Hairpin RG: C36.TIP.KCL
0.0 M -0.111 Kcal/mol
150 mM +0.124 Kcal/mol
300 mM +0.210 Kcal/mol
1.0 M +6.122 Kcal/mol
Hairpin RG: C27.TIP.NACL
0.0 M +4.516 Kcal/mol
150 mM +8.480 Kcal/mol
300 mM +6.131 Kcal/mol
1.0 M +5.679 Kcal/mol
Hairpin RG: C27.TIP.KCL
0.0 M +3.302 Kcal/mol
150 mM +2.820 Kcal/mol
300 mM +5.722 Kcal/mol
1.0 M +5.736 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Radius of Gyration Coordinate


Hairpin Hydrophobic: AMB.TIP.NACL
0.0 M +4.736 Kcal/mol
150 mM +3.799 Kcal/mol
300 mM +5.040 Kcal/mol
1.0 M +5.149 Kcal/mol
Hairpin Hydrophobic: AMB.TIP.KCL
0.0 M +4.300 Kcal/mol
150 mM +4.169 Kcal/mol
300 mM +3.127 Kcal/mol
1.0 M +6.990 Kcal/mol
Hairpin Hydrophobic: AMB.SPC.NACL
0.0 M +3.961 Kcal/mol
150 mM +6.848 Kcal/mol
300 mM +5.125 Kcal/mol
1.0 M +9.299 Kcal/mol
Hairpin Hydrophobic: AMB.SPC.KCL
0.0 M +6.175 Kcal/mol
150 mM +1.901 Kcal/mol
300 mM +4.437 Kcal/mol
1.0 M +2.483 Kcal/mol
Hairpin Hydrophobic: C36.TIP.NACL
0.0 M +12.534 Kcal/mol
150 mM +6.773 Kcal/mol
300 mM +11.090 Kcal/mol
1.0 M +7.085 Kcal/mol
Hairpin Hydrophobic: C36.TIP.KCL
0.0 M +5.883 Kcal/mol
150 mM +5.453 Kcal/mol
300 mM +5.712 Kcal/mol
1.0 M +12.035 Kcal/mol
Hairpin Hydrophobic: C27.TIP.NACL
0.0 M +4.733 Kcal/mol
150 mM +10.245 Kcal/mol
300 mM +2.395 Kcal/mol
1.0 M +5.474 Kcal/mol
Hairpin Hydrophobic: C27.TIP.KCL
0.0 M +3.849 Kcal/mol
150 mM +4.144 Kcal/mol
300 mM +4.868 Kcal/mol
1.0 M +5.780 Kcal/mol
Helmholtz Free Energy (Kcal/mol)
Hydrophobic Contacts Coordinate

CURRICULUM VITA

NAME: J. Patrick Brian, P.E.

ADDRESS: Department of Chemical Engineering
216 Eastern Parkway
University of Louisville
Ernst Hall, Rm 207
Louisville, KY 40202

DOB: Louisville, Kentucky – February 1, 1957

EDUCATION, TRAINING & EXPERIENCE:

B.S., Chemical Engineering
University of Kentucky, 1976-80

BFGoodrich/Westlake Chemical Corporation
Calvert City, Kentucky 1980-2012

Professional Engineering License, KY #14222
Chemical Engineering, 1985

Professional Engineering License, KY #14222
Electrical Engineering, 1988

Ph.D., Chemical Engineering
University of Louisville, 2014-2020

PROFESSIONAL SOCIETIES:

Omega Chi Epsilon
Kentucky Society of Professional Engineers
National Society of Professional Engineers

PUBLICATIONS:

Controlling the Product Syngas H-2:CO Ratio through Pulsed-Bias Electrochemical Reduction of CO2 on Copper
ACS CATALYSIS - July 1, 2016

New trends in the development of heterogeneous catalysts for electrochemical CO2 reduction
CATALYSIS TODAY - July 15, 2016

Reduced SnO2 Porous Nanowires with a High Density of Grain Boundaries as Catalysts for Efficient Electrochemical CO2-into-HCOOH Conversion

ANGEWANDTE CHEMIE-INTERNATIONAL EDITION
February 23, 2017

Electrophysiological measurements reveal that a succinyl linker enhances performance of the synthetic chloride channel SCMTR.
Chemical communications (Cambridge, England) - April 20, 2018

Heterogeneously catalyzed two-step cascade electrochemical reduction of CO2 to ethanol
Electrochimica Acta - April 30, 2018